

User's Manual

NEC

μ PD78078, 78078Y Subseries

8-bit Single-chip Microcontrollers

μ PD78076
 μ PD78078
 μ PD78P078
 μ PD78076Y
 μ PD78078Y
 μ PD78P078Y

Document No. U10641EJ4V0UM00 (4th edition)
Date Published December 1997 N

© NEC Corporation 1994
Printed in Japan

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

FIP, EEPROM, IEBus, and QTOP are trademarks of NEC Corporation.

MS-DOS, Windows, and WindowsNT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

IBM DOS, IBM PC/AT, and PC DOS are trademarks of International Business Machines Corporation.

HP9000 Series 700 and HP-UX are trademarks of Hewlett-Packard Company.

SPARCstation is a trademark of SPARC International, Inc.

SunOS is a trademark of Sun Microsystems, Inc.

Ethernet is a trademark of XEROX Corporation.

OSF/Motif is a trademark of Open Software Foundation, Inc.

NEWS and NEWS-OS are trademarks of Sony Corporation.

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: μ PD78P078KL-T, 78P078YKL-T

The customer must judge the need for license:

μ PD78076GC-xxx-7EA, 78076GC-xxx-8EU, 78076GF-xxx-3BA,

μ PD78076YGF-xxx-3BA

μ PD78078GC-xxx-7EA, 78078GC-xxx-8EU, 78078GF-xxx-3BA,

μ PD78078YGF-xxx-3BA

μ PD78P078GC-7EA, 78P078GC-8EU, 78P078GF-3BA, 78P078YGF-3BA

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

Purchase of NEC I²C components conveys a license under the Philips I²C Patent Rights to use these components in an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips.

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

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

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

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

NEC devices are classified into the following three quality grades:

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

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

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

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

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

Anti-radioactive design is not implemented in this product.

Regional Information

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

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

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

NEC Electronics Inc. (U.S.)

Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC Electronics (Germany) GmbH

Duesseldorf, Germany
Tel: 0211-65 03 02
Fax: 0211-65 03 490

NEC Electronics (UK) Ltd.

Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics Italiana s.r.l.

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
Taebly, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

NEC Electronics Hong Kong Ltd.

Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.

Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Singapore Pte. Ltd.

United Square, Singapore 1130
Tel: 253-8311
Fax: 250-3583

NEC Electronics Taiwan Ltd.

Taipei, Taiwan
Tel: 02-719-2377
Fax: 02-719-5951

NEC do Brasil S.A.

Cumbica-Guarulhos-SP, Brasil
Tel: 011-6465-6810
Fax: 011-6465-6829

Major Revisions in This Edition

Page	Description
Throughout	The following products have been changed from “under development” to “already developed”. μ PD78078Y Subseries: μ PD78076Y, 78078Y, 78P078Y
	The following packages have been added to the μ PD78078Y Subseries. 100-pin plastic LQFP (Fine pitch) (14 × 14 mm, resin thickness 1.4 mm)
p. 139 to 143, 149, 153	Block diagrams of ports have been changed. Figure 6-5. Block Diagram of P20, P21, P23 to P26, Figure 6-6. Block Diagram of P22 and P27, Figure 6-7. Block Diagram of P20, P21, P23 to P26, Figure 6-8. Block Diagram of P22 and P27, Figure 6-9. Block Diagram of P30 to P37, Figure 6-16. Block Diagram of P71 and P72, Figure 6-20. Block Diagram of P100 and P101
p. 169	Table 7-2. Relationship between CPU Clock and Minimum Instruction Execution Time has been added.
p. 181	8.1 Outline of Timers Incorporated into μPD78078, 78078Y Subseries has been added.
p. 241	Figure 9-10. Square Wave Output Operation Timing has been added.
p. 262	Figure 10-13. Square Wave Output Operation Timing has been added.
p. 277	Figure 12-1. Block Diagram of Watchdog Timer has been corrected.
p. 316, 366	Precautions have been added to 17.1, 18.1 Serial Interface Channel 0 Functions .
p. 323, 374	Precautions have been added to 17.3 (2), 18.3 (2) Serial Operating Mode Register 0 (CSIM0) .
p. 372	Note about the BSYE flag in Figure 17-5. Serial Bus Interface Control Register Format has been changed.
p. 336	Precautions have been added to 17.4.3 (2) (a) Bus release signal (REL), (b) Command signal (CMD) .
p. 449	19.4.3 (3) (d) Busy control option, (e) Busy & strobe control option, and (f) Bit slippage detection function have been changed to (4) Synchronization control , and the explanation has been improved.
p. 481	20.4.2 (2) (d) Reception Conditions of INTSR generation when receive error occurs has been corrected.
p. 482	Figure 20-10. Receive Error Timing has been corrected, and note has been added.
p. 490	20.4.3 (3) MSB/LSB switching as the start bit has been added.
p. 491	20.4.4 Restrictions on using UART mode has been added.
p. 569	Precautions have been added to Table 27-1. Differences between μPD78P078, 78P078Y and Mask ROM Versions .
p. 597	APPENDIX A DIFFERENCES BETWEEN μPD78078, 78075B SUBSERIES AND μPD78070A has been added.
p. 599 to 612	APPENDIX B DEVELOPMENT TOOL Entirely revised: Supports in-circuit emulator IE-78K0-NS
p. 613, 614	APPENDIX C EMBEDDED SOFTWARE Entirely revised: Fuzzy inference developing support system has been deleted.

The mark ★ shows major revised points.

INTRODUCTION

Readers

This manual has been prepared for user engineers who understand the functions of the μ PD78078 and 78078Y Subseries and design and develop its application systems and programs.

The μ PD78078 and 78078Y Subseries consist of the following members.

- μ PD78078 Subseries: μ PD78076, 78078, 78P078
- μ PD78078Y Subseries: μ PD78076Y, 78078Y, 78P078Y

★

Caution Of the above members, the following devices with the suffix **KL-T** should be used only for experiment or function evaluation, because they are not intended for use in equipment that will be mass-produced and do not have enough reliability.

- μ PD78P078KL-T and 78P078YKL-T

Purpose

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

Organization

The μ PD78078 and 78078Y Subseries manual is separated into two parts: this manual and the instructions edition (common to the 78K/0 Series).

μPD78078, 78078Y Subseries User's Manual (This manual)	78K/0 Series User's Manual — Instructions —
<ul style="list-style-type: none">• Pin functions• Internal block functions• Interrupt• Other on-chip peripheral functions	<ul style="list-style-type: none">• CPU functions• Instruction set• Explanation of each instruction

How to Read This Manual

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

- When you want to understand the functions in general:
 - Read this manual in the order of the contents.
- How to interpret the register format:
 - For the circled bit number, the bit name is defined as a reserved word in RA78K/0, and in CC78K/0, already defined in the header file named sfrbit.h.
- When you know a register name and want to confirm its details:
 - Read "**APPENDIX D REGISTER INDEX**"
- To know the differences between the μ PD78054 and 78054Y Subseries:
 - See sections **1.10** and **2.10**, titled "**Differences with μ PD78054 Subseries**" and "**Differences with μ PD78054Y Subseries**", respectively.
- To know the μ PD78078 and 78078Y Subseries instruction function in detail:
 - Refer to "**78K/0 Series User's Manual—Instructions (U12326E)**"
- To know the application example of each function of the μ PD78078 and 78078Y Subseries:
 - Refer to separately available Application Note.

Chapter Organization: This manual divides the descriptions for the μ PD78078 and 78078Y Subseries into different chapters as shown below. Read only the chapters related to the device you use.

Chapter	μ PD78078 Subseries	μ PD78078Y Subseries
Chapter 1 Outline (μ PD78078 Subseries)	√	—
Chapter 2 Outline (μ PD78078Y Subseries)	—	√
Chapter 3 Pin Function (μ PD78078 Subseries)	√	—
Chapter 4 Pin Function (μ PD78078Y Subseries)	—	√
Chapter 5 CPU Architecture	√	√
Chapter 6 Port Functions	√	√
Chapter 7 Clock Generator	√	√
Chapter 8 16-Bit Timer/Event Counter	√	√
Chapter 9 8-Bit Timer/Event Counters 1 and 2	√	√
Chapter 10 8-Bit Timer/Event Counters 5 and 6	√	√
Chapter 11 Watch Timer	√	√
Chapter 12 Watchdog Timer	√	√
Chapter 13 Clock Output Control Circuit	√	√
Chapter 14 Buzzer Output Control Circuit	√	√
Chapter 15 A/D Converter	√	√
Chapter 16 D/A Converter	√	√
Chapter 17 Serial Interface Channel 0 (μ PD78078 Subseries)	√	—
Chapter 18 Serial Interface Channel 0 (μ PD78078Y Subseries)	—	√
Chapter 19 Serial Interface Channel 1	√	√
Chapter 20 Serial Interface Channel 2	√	√
Chapter 21 Real-Time Output Port	√	√
Chapter 22 Interrupt and Test Functions	√	√
Chapter 23 External Device Expansion Function	√	√
Chapter 24 Standby Function	√	√
Chapter 25 Reset Function	√	√
Chapter 26 ROM Correction	√	√
Chapter 27 μ PD78P078, μ PD78P078Y	√	√
Chapter 28 Instruction Set	√	√

Differences between μ PD78078 and μ PD78078Y Subseries

The μ PD78078 and μ PD78078Y Subseries are different in the following functions of the serial interface channel 0.

Mode of serial interface channel 0	μ PD78078 Subseries	μ PD78078Y Subseries
3-wire serial I/O mode	√	√
2-wire serial I/O mode	√	√
SBI (serial bus interface) mode	√	—
I ² C (Inter IC) bus mode	—	√

√ : Supported

— : Not supported

Legend	Data significance	: High digits on the left and low digits on the right
	Active low representations	: $\bar{x}x\bar{x}$ (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 ... xxxx or xxxxB Decimal ... xxxx Hexadecimal ... xxxxH

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

• Related documents for this subseries

Document name	Document No.	
	English	Japanese
μ PD78078, 78078Y Subseries User's Manual	This manual	U10641J
μ PD78076, 78078 Data Sheet	U10167E	U10167J
μ PD78P078 Data Sheet	U10168E	U10168J
μ PD78076Y, 78078Y Data Sheet	U10605E	U10605J
μ PD78P078Y Data Sheet	U10606E	U10606J
78K/0 Series User's Manual—Instructions	U12326E	U12326J
78K/0 Series Instruction Table	—	U10903J
78K/0 Series Instruction Set	—	U10904J
μ PD78078 Subseries Special Function Register Table	—	IEM-5607
μ PD78078Y Subseries Special Function Register Table	—	IEM-5601
78K/0 Series Application Note Basics (III)	U10182E	U10182J

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

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

Document Name		Document No.	
		English	Japanese
RA78K Series Assembler Package	Operation	EEU-1399	EEU-809
	Language	EEU-1404	EEU-815
RA78K Series Structured Assembler Preprocessor		EEU-1402	U12323J
RA78K0 Assembler Package	Operation	U11802E	U11802J
	Language	U11801E	U11801J
	Structured Assembly Language	U11789E	U11789J
CC78K Series C Compiler	Operation	EEU-1280	EEU-656
	Language	EEU-1284	EEU-655
CC78K0 C Compiler	Operation	U11517E	U11517J
	Language	U11518E	U11518J
CC78K0 C Compiler Application Note	Programming Know-how	EEA-1208	U13034J
CC78K Series Library Source File		—	U12322J
PG-1500 PROM Programmer		EEU-1335	U11940J
PG-1500 Controller PC-9800 Series (MS-DOS™)-Based		EEU-1291	EEU-704
PG-1500 Controller IBM PC Series (PC DOS™)-Based		U10540E	EEU-5008
IE-78K0-NS		To be prepared	To be prepared
IE-78001-R-A		To be prepared	To be prepared
IE-78K0-R-EX1		To be prepared	To be prepared
IE-78078-NS-EM1		To be prepared	To be prepared
IE-78078-R-EM		U10775E	U10775J
EP-78064		EEU-1469	EEU-934
SM78K0 System Simulator Windows™-Based	Reference	U10181E	U10181J
SM78K Series System Simulator External Part User Open Interface Specifications		U10092E	U10092J
ID78K0-NS Integrated Debugger	Reference	To be prepared	U12900J
ID78K0 Integrated Debugger EWS Based	Reference	—	U11151J
ID78K0 Integrated Debugger PC Based	Reference	U11539E	U11539J
ID78K0 Integrated Debugger Windows Based	Guides	U11649E	U11649J

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

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

Document Name		Document No.	
		English	Japanese
78K/0 Series Real-time OS	Basics	U11537E	U11537J
	Installation	U11536E	U11536J
78K/0 Series OS MX78K0	Basics	U12257E	U12257J

- **Other Documents**

Document Name		Document No.	
		English	Japanese
IC Package Manual		C10943X	
Semiconductor Device Mounting Technology Manual		C10535E	C10535J
Quality Grades on NEC Semiconductor Devices		C11531E	C11531J
NEC Semiconductor Device Reliability/Quality Control System		U10983E	U10983J
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)		C11892E	C11892J
Guide to Quality Assurance for Semiconductor Devices		MEI-1202	—
Microcomputer Product Series Guide		—	U11416J

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

[MEMO]

TABLE OF CONTENTS

CHAPTER 1 OUTLINE (μPD78078 SUBSERIES)	33
1.1 Features	33
1.2 Application Fields	34
1.3 Ordering Information	34
1.4 Quality Grade	35
1.5 Pin Configuration (Top View)	36
1.6 78K/0 Series Expansion	42
1.7 Block Diagram	44
1.8 Outline of Function	45
1.9 Mask Options	47
1.10 Differences with μPD78054 Subseries	47
CHAPTER 2 OUTLINE (μPD78078Y SUBSERIES)	49
2.1 Features	49
2.2 Application Fields	50
2.3 Ordering Information	50
2.4 Quality Grade	51
2.5 Pin Configuration (Top View)	52
2.6 78K/0 Series Expansion	58
2.7 Block Diagram	60
2.8 Outline of Function	61
2.9 Mask Options	63
2.10 Differences with μPD78054Y Subseries	63
CHAPTER 3 PIN FUNCTION (μPD78078 SUBSERIES)	65
3.1 Pin Function List	65
3.1.1 Normal operating mode pins	65
3.1.2 PROM programming mode pins (μ PD78P078 only)	69
3.2 Description of Pin Functions	70
3.2.1 P00 to P07 (Port 0)	70
3.2.2 P10 to P17 (Port 1)	70
3.2.3 P20 to P27 (Port 2)	71
3.2.4 P30 to P37 (Port 3)	72
3.2.5 P40 to P47 (Port 4)	72
3.2.6 P50 to P57 (Port 5)	73
3.2.7 P60 to P67 (Port 6)	73
3.2.8 P70 to P72 (Port 7)	74
3.2.9 P80 to P87 (Port 8)	74
3.2.10 P90 to P96 (Port 9)	75
3.2.11 P100 to P103 (Port 10)	75
3.2.12 P120 to P127 (Port 12)	75
3.2.13 P130 and P131 (Port 13)	76
3.2.14 AVREF0	76
3.2.15 AVREF1	76
3.2.16 AVDD	76

3.2.17	AV _{SS}	76
3.2.18	RESET	76
3.2.19	X1 and X2	76
3.2.20	XT1 and XT2	76
3.2.21	V _{DD}	77
3.2.22	V _{SS}	77
3.2.23	V _{PP} (μ PD78P078 only)	77
3.2.24	IC (Mask ROM version only)	77
3.3	Input/output Circuits and Recommended Connection of Unused Pins	78
CHAPTER 4	PIN FUNCTION (μPD78078Y SUBSERIES)	83
4.1	Pin Function List	83
4.1.1	Normal operating mode pins	83
4.1.2	PROM programming mode pins (μ PD78P078Y only)	87
4.2	Description of Pin Functions	88
4.2.1	P00 to P07 (Port 0)	88
4.2.2	P10 to P17 (Port 1)	88
4.2.3	P20 to P27 (Port 2)	89
4.2.4	P30 to P37 (Port 3)	90
4.2.5	P40 to P47 (Port 4)	90
4.2.6	P50 to P57 (Port 5)	91
4.2.7	P60 to P67 (Port 6)	91
4.2.8	P70 to P72 (Port 7)	92
4.2.9	P80 to P87 (Port 8)	92
4.2.10	P90 to P96 (Port 9)	93
4.2.11	P100 to P103 (Port 10)	93
4.2.12	P120 to P127 (Port 12)	93
4.2.13	P130 and P131 (Port 13)	94
4.2.14	AV _{REF0}	94
4.2.15	AV _{REF1}	94
4.2.16	AV _{DD}	94
4.2.17	AV _{SS}	94
4.2.18	RESET	94
4.2.19	X1 and X2	94
4.2.20	XT1 and XT2	94
4.2.21	V _{DD}	95
4.2.22	V _{SS}	95
4.2.23	V _{PP} (μ PD78P078Y only)	95
4.2.24	IC (Mask ROM version only)	95
4.3	Input/output Circuits and Recommended Connection of Unused Pins	96
CHAPTER 5	CPU ARCHITECTURE	101
5.1	Memory Spaces	101
5.1.1	Internal program memory space	104
5.1.2	Internal data memory space	106
5.1.3	Special function register (SFR) area	106
5.1.4	External memory space	106
5.1.5	Data memory addressing	107

5.2	Processor Registers	110
5.2.1	Control registers	110
5.2.2	General registers	113
5.2.3	Special function register (SFR)	114
5.3	Instruction Address Addressing	118
5.3.1	Relative addressing	118
5.3.2	Immediate addressing	119
5.3.3	Table indirect addressing	120
5.3.4	Register addressing	121
5.4	Operand Address Addressing	122
5.4.1	Implied addressing	122
5.4.2	Register addressing	123
5.4.3	Direct addressing	124
5.4.4	Short direct addressing	125
5.4.5	Special function register (SFR) addressing	126
5.4.6	Register indirect addressing	127
5.4.7	Based addressing	128
5.4.8	Based indexed addressing	129
5.4.9	Stack addressing	129
CHAPTER 6	PORT FUNCTIONS	131
6.1	Port Functions	131
6.2	Port Configuration	136
6.2.1	Port 0	136
6.2.2	Port 1	138
6.2.3	Port 2 (μ PD78078 Subseries)	139
6.2.4	Port 2 (μ PD78078Y Subseries)	141
6.2.5	Port 3	143
6.2.6	Port 4	144
6.2.7	Port 5	145
6.2.8	Port 6	146
6.2.9	Port 7	148
6.2.10	Port 8	150
6.2.11	Port 9	151
6.2.12	Port 10	153
6.2.13	Port 12	155
6.2.14	Port 13	156
6.3	Port Function Control Registers	157
6.4	Port Function Operations	163
6.4.1	Writing to input/output port	163
6.4.2	Reading from input/output port	163
6.4.3	Operations on input/output port	163
6.5	Selection of Mask Option	164

CHAPTER 7	CLOCK GENERATOR	165
7.1	Clock Generator Functions	165
7.2	Clock Generator Configuration	166
7.3	Clock Generator Control Register	167
7.4	System Clock Oscillator	171
7.4.1	Main system clock oscillator	171
7.4.2	Subsystem clock oscillator	172
7.4.3	Divider	174
7.4.4	When no subsystem clocks are used	174
7.5	Clock Generator Operations	175
7.5.1	Main system clock operations	176
7.5.2	Subsystem clock operations	177
7.6	Changing System Clock and CPU Clock Settings	178
7.6.1	Time required for switchover between system clock and CPU clock	178
7.6.2	System clock and CPU clock switching procedure	179
CHAPTER 8	16-BIT TIMER/EVENT COUNTER	181
★ 8.1	Outline of Timers Incorporated into μPD78078, 78078Y Subseries	181
8.2	16-Bit Timer/Event Counter Functions	183
8.3	16-Bit Timer/Event Counter Configuration	185
8.4	16-Bit Timer/Event Counter Control Registers	190
8.5	16-Bit Timer/Event Counter Operations	198
8.5.1	Interval timer operations	198
8.5.2	PWM output operations	200
8.5.3	PPG output operations	203
8.5.4	Pulse width measurement operations	204
8.5.5	External event counter operation	211
8.5.6	Square-wave output operation	213
8.5.7	One-shot pulse output operation	215
8.6	16-Bit Timer/Event Counter Operating Precautions	219
CHAPTER 9	8-BIT TIMER/EVENT COUNTERS 1 AND 2	223
9.1	8-Bit Timer/Event Counters 1 and 2 Functions	223
9.1.1	8-bit timer/event counter mode	223
9.1.2	16-bit timer/event counter mode	226
9.2	8-Bit Timer/Event Counters 1 and 2 Configurations	228
9.3	8-Bit Timer/Event Counters 1 and 2 Control Registers	231
9.4	8-Bit Timer/Event Counters 1 and 2 Operations	236
9.4.1	8-bit timer/event counter mode	236
9.4.2	16-bit timer/event counter mode	242
9.5	8-Bit Timer/Event Counters 1 and 2 Precautions	246

CHAPTER 10 8-BIT TIMER/EVENT COUNTERS 5 AND 6	249
10.1 8-Bit Timer/Event Counters 5 and 6 Functions	249
10.2 8-Bit Timer/Event Counters 5 and 6 Configurations	252
10.3 8-Bit Timer/Event Counters 5 and 6 Control Registers	254
10.4 8-Bit Timer/Event Counters 5 and 6 Operations	259
10.4.1 Interval timer operations	259
10.4.2 External event counter operation	261
10.4.3 Square-wave output	262
10.4.4 PWM output operations	264
10.5 8-Bit Timer/Event Counters 5 and 6 Precautions	267
CHAPTER 11 WATCH TIMER	269
11.1 Watch Timer Functions	269
11.2 Watch Timer Configuration	270
11.3 Watch Timer Control Registers	271
11.4 Watch Timer Operations	274
11.4.1 Watch timer operation	274
11.4.2 Interval timer operation	274
CHAPTER 12 WATCHDOG TIMER	275
12.1 Watchdog Timer Functions	275
12.2 Watchdog Timer Configuration	277
12.3 Watchdog Timer Control Registers	278
12.4 Watchdog Timer Operations	281
12.4.1 Watchdog timer operation	281
12.4.2 Interval timer operation	282
CHAPTER 13 CLOCK OUTPUT CONTROL CIRCUIT	283
13.1 Clock Output Control Circuit Functions	283
13.2 Clock Output Control Circuit Configuration	284
13.3 Clock Output Function Control Registers	285
CHAPTER 14 BUZZER OUTPUT CONTROL CIRCUIT	289
14.1 Buzzer Output Control Circuit Functions	289
14.2 Buzzer Output Control Circuit Configuration	289
14.3 Buzzer Output Function Control Registers	290
CHAPTER 15 A/D CONVERTER	293
15.1 A/D Converter Functions	293
15.2 A/D Converter Configuration	293
15.3 A/D Converter Control Registers	296
15.4 A/D Converter Operations	300
15.4.1 Basic operations of A/D converter	300
15.4.2 Input voltage and conversion results	302
15.4.3 A/D converter operating mode	303
15.5 A/D Converter Cautions	305

CHAPTER 16 D/A CONVERTER	309
16.1 D/A Converter Functions	309
16.2 D/A Converter Configuration	310
16.3 D/A Converter Control Registers	312
16.4 D/A Converter Operations	313
16.5 D/A Converter Cautions	314
CHAPTER 17 SERIAL INTERFACE CHANNEL 0 (μPD78078 SUBSERIES)	315
17.1 Serial Interface Channel 0 Functions	316
17.2 Serial Interface Channel 0 Configuration	318
17.3 Serial Interface Channel 0 Control Registers	321
17.4 Serial Interface Channel 0 Operations	328
17.4.1 Operation stop mode	328
17.4.2 3-wire serial I/O mode operation	329
17.4.3 SBI mode operation	333
17.4.4 2-wire serial I/O mode operation	357
17.4.5 $\overline{SCK0}$ /P27 pin output manipulation	363
CHAPTER 18 SERIAL INTERFACE CHANNEL 0 (μPD78078Y Subseries)	365
18.1 Serial Interface Channel 0 Functions	366
18.2 Serial Interface Channel 0 Configuration	368
18.3 Serial Interface Channel 0 Control Registers	372
18.4 Serial Interface Channel 0 Operations	380
18.4.1 Operation stop mode	380
18.4.2 3-wire serial I/O mode operation	381
18.4.3 2-wire serial I/O mode operation	385
18.4.4 I ² C bus mode operation	390
18.4.5 Cautions on use of I ² C bus mode	408
18.4.6 Restrictions in I ² C bus mode	411
18.4.7 $\overline{SCK0}$ /SCL/P27 pin output manipulation	413
CHAPTER 19 SERIAL INTERFACE CHANNEL 1	415
19.1 Serial Interface Channel 1 Functions	415
19.2 Serial Interface Channel 1 Configuration	416
19.3 Serial Interface Channel 1 Control Registers	418
19.4 Serial Interface Channel 1 Operations	425
19.4.1 Operation stop mode	425
19.4.2 3-wire serial I/O mode operation	426
19.4.3 3-wire serial I/O mode operation with automatic transmit/receive function	429
CHAPTER 20 SERIAL INTERFACE CHANNEL 2	457
20.1 Serial Interface Channel 2 Functions	457
20.2 Serial Interface Channel 2 Configuration	458
20.3 Serial Interface Channel 2 Control Registers	461
20.4 Serial Interface Channel 2 Operation	469
20.4.1 Operation stop mode	469
20.4.2 Asynchronous serial interface (UART) mode	471
20.4.3 3-wire serial I/O mode	484
★ 20.4.4 Restrictions on using UART mode	491

CHAPTER 21 REAL-TIME OUTPUT PORT	495
21.1 Real-Time Output Port Functions	495
21.2 Real-Time Output Port Configuration	495
21.3 Real-Time Output Port Control Registers	497
CHAPTER 22 INTERRUPT FUNCTIONS	499
22.1 Interrupt Function Types	499
22.2 Interrupt Sources and Configuration	500
22.3 Interrupt Function Control Registers	504
22.4 Interrupt Servicing Operations	513
22.4.1 Non-maskable interrupt request acknowledge operation	513
22.4.2 Maskable interrupt request acknowledge operation	516
22.4.3 Software interrupt request acknowledge operation	518
22.4.4 Multiple interrupt servicing	519
22.4.5 Interrupt request reserve	522
22.5 Test Functions	523
22.5.1 Registers controlling the test function	523
22.5.2 Test input signal acknowledge operation	525
CHAPTER 23 EXTERNAL DEVICE EXPANSION FUNCTION	527
23.1 External Device Expansion Functions	527
23.2 External Device Expansion Function Control Register	531
23.3 External Device Expansion Function Timing	534
23.3.1 Timings in multiplexed bus mode	534
23.3.2 Timings in separate bus mode	539
CHAPTER 24 STANDBY FUNCTION	545
24.1 Standby Function and Configuration	545
24.1.1 Standby function	545
24.1.2 Standby function control register	546
24.2 Standby Function Operations	547
24.2.1 HALT mode	547
24.2.2 STOP mode	550
CHAPTER 25 RESET FUNCTION	553
25.1 Reset Function	553
CHAPTER 26 ROM CORRECTION	559
26.1 ROM Correction Functions	559
26.2 ROM Correction Configuration	559
26.3 ROM Correction Control Registers	561
26.4 ROM Correction Application	562
26.5 ROM Correction Example	565
26.6 Program Execution Flow	566
26.7 Cautions on ROM Correction	568

CHAPTER 27	μPD78P078, 78P078Y	569
27.1	Internal Memory Size Switching Register	570
27.2	Internal Extension RAM Size Switching Register	571
27.3	PROM Programming	572
27.3.1	Operating modes	572
27.3.2	PROM write procedure	574
27.3.3	PROM reading procedure	578
27.4	Erasure Procedure (μ PD78P078KL-T and 78P078YKL-T Only)	579
27.5	Opaque Film Masking Window (μ PD78P078KL-T and 78P078YKL-T Only)	579
27.6	Screening of One-Time PROM Versions	579
CHAPTER 28	INSTRUCTION SET	581
28.1	Legends Used in Operation List	582
28.1.1	Operand identifiers and description methods	582
28.1.2	Description of "operation" column	583
28.1.3	Description of "flag operation" column	583
28.2	Operation List	584
28.3	Instructions Listed by Addressing Type	592
★	APPENDIX A DIFFERENCES BETWEEN μPD78078, 78075B SUBSERIES, AND μPD78070A	597
APPENDIX B	DEVELOPMENT TOOLS	599
B.1	Language Processing Software	602
B.2	PROM Writing Tools	604
B.2.1	Hardware	604
B.2.2	Software	604
B.3	Debugging Tools	605
B.3.1	Hardware	605
B.3.2	Software	607
B.4	OS for IBM PC	609
B.5	System Upgrading from Former-type In-circuit Emulator for 78K/0 Series to IE-78001-R-A	609
APPENDIX C	EMBEDDED SOFTWARE	613
APPENDIX D	REGISTER INDEX	615
D.1	Register Name Index	615
D.2	Register Symbol Index	619
APPENDIX E	REVISION HISTORY	623

LIST OF FIGURES (1/9)

Figure No.	Title	Page
3-1	List of Pin Input/Output Circuits	80
4-1	List of Pin Input/Output Circuits	98
5-1	Memory Map (μ PD78076, 78076Y)	101
5-2	Memory Map (μ PD78078, 78078Y)	102
5-3	Memory Map (μ PD78P078, μ PD78P078Y)	103
5-4	Data Memory Addressing (μ PD78076, 78076Y)	107
5-5	Data Memory Addressing (μ PD78078, 78078Y)	108
5-6	Data Memory Addressing (μ PD78P078, 78P078Y)	109
5-7	Program Counter Configuration	110
5-8	Program Status Word Configuration	110
5-9	Stack Pointer Configuration	112
5-10	Data to be Saved to Stack Memory	112
5-11	Data to be Reset from Stack Memory	112
5-12	General Register Configuration	113
6-1	Port Types	131
6-2	Block Diagram of P00 and P07	137
6-3	Block Diagram of P01 to P06	137
6-4	Block Diagram of P10 to P17	138
6-5	Block Diagram of P20, P21, P23 to P26	139
6-6	Block Diagram of P22 and P27	140
6-7	Block Diagram of P20, P21, P23 to P26	141
6-8	Block Diagram of P22 and P27	142
6-9	Block Diagram of P30 to P37	143
6-10	Block Diagram of P40 to P47	144
6-11	Block Diagram of Falling Edge Detection Circuit	144
6-12	Block Diagram of P50 to P57	145
6-13	Block Diagram of P60 to P63	147
6-14	Block Diagram of P64 to P67	147
6-15	Block Diagram of P70	148
6-16	Block Diagram of P71 and P72	149
6-17	Block Diagram of P80 to P87	150
6-18	Block Diagram of P90 to P93	152
6-19	Block Diagram of P94 to P96	152
6-20	Block Diagram of P100 and P101	153
6-21	Block Diagram of P102 and P103	154
6-22	Block Diagram of P120 to P127	155
6-23	Block Diagram of P130 and P131	156
6-24	Port Mode Register Format	159
6-25	Pull-Up Resistor Option Register Format	160
6-26	Memory Expansion Mode Register Format	161
6-27	Key Return Mode Register Format	162

LIST OF FIGURES (2/9)

Figure No.	Title	Page
7-1	Block Diagram of Clock Generator	166
7-2	Subsystem Clock Feedback Resistor	167
7-3	Processor Clock Control Register Format	168
7-4	Oscillation Mode Selection Register Format	170
7-5	Main System Clock Waveform due to Writing to OSMS	170
7-6	External Circuit of Main System Clock Oscillator	171
7-7	External Circuit of Subsystem Clock Oscillator	172
7-8	Examples of Oscillator with Bad Connection	172
7-9	Main System Clock Stop Function	176
7-10	System Clock and CPU Clock Switching	179
8-1	16-Bit Timer/Event Counter Block Diagram	186
8-2	16-Bit Timer/Event Counter Output Control Circuit Block Diagram	187
8-3	Timer Clock Selection Register 0 Format	191
8-4	16-Bit Timer Mode Control Register Format	192
8-5	Capture/Compare Control Register 0 Format	193
8-6	16-Bit Timer Output Control Register Format	194
8-7	Port Mode Register 3 Format	195
8-8	External Interrupt Mode Register 0 Format	196
8-9	Sampling Clock Select Register Format	197
8-10	Control Register Settings for Interval Timer Operation	198
8-11	Interval Timer Configuration Diagram	199
8-12	Interval Timer Operation Timings	199
8-13	Control Register Settings for PWM Output Operation	201
8-14	Example of D/A Converter Configuration with PWM Output	202
8-15	TV Tuner Application Circuit Example	202
8-16	Control Register Settings for PPG Output Operation	203
8-17	Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register	204
8-18	Configuration Diagram for Pulse Width Measurement by Free-Running Counter	205
8-19	Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)	205
8-20	Control Register Settings for Two Pulse Width Measurements with Free-Running Counter	206
8-21	Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified)	207
8-22	Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers	208
8-23	Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified)	209
8-24	Control Register Settings for Pulse Width Measurement by Means of Restart	210
8-25	Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)	210

LIST OF FIGURES (3/9)

Figure No.	Title	Page
8-26	Control Register Settings in External Event Counter Mode	211
8-27	External Event Counter Configuration Diagram	212
8-28	External Event Counter Operation Timings (with Rising Edge Specified)	212
8-29	Control Register Settings in Square-Wave Output Mode	213
8-30	Square-Wave Output Operation Timing	214
8-31	Control Register Settings for One-Shot Pulse Output Operation Using Software Trigger	215
8-32	Timing of One-Shot Pulse Output Operation Using Software Trigger	216
8-33	Control Register Settings for One-Shot Pulse Output Operation Using External Trigger	217
8-34	Timing of One-Shot Pulse Output Operation Using External Trigger (With Rising Edge Specified)	218
8-35	16-Bit Timer Register Start Timing	219
8-36	Timings After Change of Compare Register During Timer Count Operation	219
8-37	Capture Register Data Retention Timing	220
8-38	Operation Timing of OVF0 Flag	221
9-1	8-Bit Timer/Event Counters 1 and 2 Block Diagram	228
9-2	Block Diagram of 8-Bit Timer/Event Counter Output Control Circuit 1	229
9-3	Block Diagram of 8-Bit Timer/Event Counter Output Control Circuit 2	229
9-4	Timer Clock Select Register 1 Format	232
9-5	8-Bit Timer Mode Control Register 1 Format	233
9-6	8-Bit Timer Output Control Register Format	234
9-7	Port Mode Register 3 Format	235
9-8	Interval Timer Operation Timing	236
9-9	External Event Counter Operation Timings (with Rising Edge Specified)	239
9-10	Timing of Square Wave Output Operation	241
9-11	Interval Timer Operation Timing	242
9-12	External Event Counter Operation Timings (with Rising Edge Specified)	244
9-13	8-Bit Timer Registers 1 and 2 Start Timing	246
9-14	External Event Counter Operation Timing	246
9-15	Timing after Compare Register Change during Timer Count Operation	247
10-1	8-Bit Timer/Event Counters 5 and 6 Block Diagram	252
10-2	Block Diagram of 8-Bit Timer/Event Counters 5 and 6 Output Control Circuit	253
10-3	Timer Clock Select Register 5 Format	254
10-4	Timer Clock Select Register 6 Format	255
10-5	8-Bit Timer Output Control Register Format	256
10-6	8-Bit Timer Output Control Register 6 Format	257
10-7	Port Mode Register 10 Format	258
10-8	8-Bit Timer Mode Control Register Settings for Interval Timer Operation	259
10-9	Interval Timer Operation Timings	259
10-10	8-Bit Timer Mode Control Register Setting for External Event Counter Operation	261
10-11	External Event Counter Operation Timings (with Rising Edge Specified)	261
10-12	8-Bit Timer Mode Control Register Settings for Square-Wave Output Operation	262
10-13	Timing of Square Wave Output Operation	262

LIST OF FIGURES (4/9)

Figure No.	Title	Page
10-14	8-Bit Timer Control Register Settings for PWM Output Operation	264
10-15	PWM Output Operation Timing (Active High Setting).....	265
10-16	PWM Output Operation Timings (CRn0 = 00H, Active High Setting)	265
10-17	PWM Output Operation Timings (CRn0 = FFH, Active High Setting)	266
10-18	PWM Output Operation Timings (CRn0 Changing, Active High Setting)	266
10-19	8-bit Timer Registers 5 and 6 Start Timings	267
10-20	External Event Counter Operation Timings	267
10-21	Timings after Compare Register Change during Timer Count Operation	268
11-1	Watch Timer Block Diagram	271
11-2	Timer Clock Select Register 2 Format	272
11-3	Watch Timer Mode Control Register Format	273
12-1	Watchdog Timer Block Diagram	277
12-2	Timer Clock Select Register 2 Format	279
12-3	Watchdog Timer Mode Register Format	280
13-1	Remote Controlled Output Application Example	283
13-2	Clock Output Control Circuit Block Diagram	284
13-3	Timer Clock Select Register 0 Format	286
13-4	Port Mode Register 3 Format	287
14-1	Buzzer Output Control Circuit Block Diagram	289
14-2	Timer Clock Select Register 2 Format	291
14-3	Port Mode Register 3 Format	292
15-1	A/D Converter Block Diagram	294
15-2	A/D Converter Mode Register Format	297
15-3	A/D Converter Input Select Register Format	298
15-4	External Interrupt Mode Register 1 Format	299
15-5	A/D Converter Basic Operation	301
15-6	Relationships between Analog Input Voltage and A/D Conversion Result	302
15-7	A/D Conversion by Hardware Start	303
15-8	A/D Conversion by Software Start	304
15-9	Example of Method of Reducing Current Consumption in Standby Mode	305
15-10	Analog Input Pin Disposition	306
15-11	A/D Conversion End Interrupt Request Generation Timing	307
15-12	Handling of AV _{DD} Pin	307
16-1	D/A Converter Block Diagram	310
16-2	D/A Converter Mode Register Format	312
16-3	Use Example of Buffer Amplifier	314

LIST OF FIGURES (5/9)

Figure No.	Title	Page
17-1	Serial Bus Interface (SBI) System Configuration Example	317
17-2	Serial Interface Channel 0 Block Diagram	318
17-3	Timer Clock Select Register 3 Format	322
17-4	Serial Operating Mode Register 0 Format	324
17-5	Serial Bus Interface Control Register Format	325
17-6	Interrupt Timing Specify Register Format	327
17-7	3-Wire Serial I/O Mode Timings	331
17-8	RELT and CMDT Operations	331
17-9	Circuit of Switching in Transfer Bit Order	332
17-10	Example of Serial Bus Configuration with SBI	333
17-11	SBI Transfer Timings	335
17-12	Bus Release Signal	336
17-13	Command Signal	336
17-14	Addresses	337
17-15	Slave Selection with Address	337
17-16	Commands	338
17-17	Data	338
17-18	Acknowledge Signal	339
17-19	$\overline{\text{BUSY}}$ and READY Signals	339
17-20	RELT, CMDT, RELD, and CMDD Operations (Master)	344
17-21	RELD and CMDD Operations (Slave)	344
17-22	ACKT Operation	345
17-23	ACKE Operations	346
17-24	ACKD Operations	347
17-25	BSYE Operation	347
17-26	Pin Configuration	350
17-27	Address Transmission from Master Device to Slave Device (WUP = 1)	352
17-28	Command Transmission from Master Device to Slave Device	353
17-29	Data Transmission from Master Device to Slave Device	354
17-30	Data Transmission from Slave Device to Master Device	355
17-31	Serial Bus Configuration Example Using 2-Wire Serial I/O Mode	357
17-32	2-Wire Serial I/O Mode Timings	361
17-33	RELT and CMDT Operations	361
17-34	$\overline{\text{SCK0/P27}}$ Pin Configuration	363
18-1	Serial Bus Configuration Example Using I ² C Bus	367
18-2	Serial Interface Channel 0 Block Diagram	369
18-3	Timer Clock Select Register 3 Format	373
18-4	Serial Operating Mode Register 0 Format	375
18-5	Serial Bus Interface Control Register Format	376
18-6	Interrupt Timing Specify Register Format	378
18-7	3-Wire Serial I/O Mode Timings	383
18-8	RELT and CMDT Operations	383
18-9	Circuit of Switching in Transfer Bit Order	384

LIST OF FIGURES (6/9)

Figure No.	Title	Page
18-10	Serial Bus Configuration Example Using 2-Wire Serial I/O Mode	385
18-11	2-Wire Serial I/O Mode Timings	388
18-12	RELT and CMDT Operations	389
18-13	Example of Serial Bus Configuration Using I ² C Bus	390
18-14	I ² C Bus Serial Data Transfer Timing	391
18-15	Start Condition	392
18-16	Address	392
18-17	Transfer Direction Specification	392
18-18	Acknowledge Signal	393
18-19	Stop Condition	393
18-20	Wait Signal	394
18-21	Pin Configuration	399
18-22	Example of Communication from Master to Slave (with 9-Clock Wait Selected for Both Master and Slave)	401
18-23	Example of Communication from Slave to Master (with 9-Clock Wait Selected for Both Master and Slave)	404
18-24	Start Condition Output	408
18-25	Slave Wait Release (Transmission)	409
18-26	Slave Wait Release (Reception)	410
18-27	$\overline{\text{SCK0}}/\text{SCL}/\text{P27}$ Pin Configuration	413
18-28	$\overline{\text{SCK0}}/\text{SCL}/\text{P27}$ Pin Configuration	414
18-29	Logic Circuit of SCL Signal	414
19-1	Serial Interface Channel 1 Block Diagram	416
19-2	Timer Clock Select Register 3 Format	418
19-3	Serial Operation Mode Register 1 Format	419
19-4	Automatic Data Transmit/Receive Control Register Format	420
19-5	Automatic Data Transmit/Receive Interval Specify Register Format	421
19-6	3-Wire Serial I/O Mode Timings	427
19-7	Circuit of Switching in Transfer Bit Order	428
19-8	Basic Transmission/Reception Mode Operation Timings	436
19-9	Basic Transmission/Reception Mode Flowchart	437
19-10	Buffer RAM Operation in 6-byte Transmission/Reception (in Basic Transmit/Receive Mode)	438
19-11	Basic Transmission Mode Operation Timings	440
19-12	Basic Transmission Mode Flowchart	441
19-13	Buffer RAM Operation in 6-byte Transmission (in Basic Transmit Mode)	442
19-14	Repeat Transmission Mode Operation Timing	444
19-15	Repeat Transmission Mode Flowchart	445
19-16	Buffer RAM Operation in 6-byte Transmission (in Repeat Transmit mode)	446
19-17	Automatic Transmission/Reception Suspension and Restart	448
19-18	System Configuration when the Busy Control Option is Used	449
19-19	Operation Timings when Using Busy Control Option (BUSY0 = 0)	450
19-20	Busy Signal and Wait Cancel (BUSY0 = 0)	451

LIST OF FIGURES (7/9)

Figure No.	Title	Page
19-21	Operation Timings when Using Busy & Strobe Control Option (BUSY0 = 0)	452
19-22	Operation Timing of the Bit Slippage Detection Function through the Busy Signal (BUSY0 = 1)	453
19-23	Automatic Data Transmit/Receive Interval	454
19-24	Operation Timing with Automatic Data Transmit/Receive Function Performed by Internal Clock	455
20-1	Serial Interface Channel 2 Block Diagram	458
20-2	Baud Rate Generator Block Diagram	459
20-3	Serial Operating Mode Register 2 Format	461
20-4	Asynchronous Serial Interface Mode Register Format	462
20-5	Asynchronous Serial Interface Status Register Format	464
20-6	Baud Rate Generator Control Register Format	465
20-7	Asynchronous Serial Interface Transmit/Receive Data Format	478
20-8	Asynchronous Serial Interface Transmission Completion Interrupt Request Generation Timing	480
20-9	Asynchronous Serial Interface Reception Completion Interrupt Request Generation Timing	481
20-10	Receive Error Timing	482
20-11	State of Receive Buffer Register (RXB) When Receive Operation is Stopped and Whether Interrupt Request (INTSR) is Generated or Not	483
20-12	3-Wire Serial I/O Mode Timing	489
20-13	Circuit of Switching in Transfer Bit Order	490
20-14	Receive Completion Interrupt Request Generation Timing (ISRM = 1)	491
20-15	Period that Reading Receive Buffer Register is Prohibited	492
21-1	Real-time Output Port Block Diagram	495
21-2	Real-time Output Buffer Register Configuration	496
21-3	Port Mode Register 12 Format	497
21-4	Real-time Output Port Mode Register Format	497
21-5	Real-time Output Port Control Register Format	498
22-1	Basic Configuration of Interrupt Function	502
22-2	Interrupt Request Flag Register Format	505
22-3	Interrupt Mask Flag Register Format	506
22-4	Priority Specify Flag Register Format	507
22-5	External Interrupt Mode Register 0 Format	508
22-6	External Interrupt Mode Register 1 Format	509
22-7	Sampling Clock Select Register Format	510
22-8	Noise Eliminator Input/Output Timing (during Rising Edge Detection)	511
22-9	Program Status Word Format	512
22-10	Flowchart from Non-Maskable Interrupt Generation to Acknowledge	514
22-11	Non-Maskable Interrupt Request Acknowledge Timing	514
22-12	Non-Maskable Interrupt Request Acknowledge Operation	515

LIST OF FIGURES (8/9)

Figure No.	Title	Page
22-13	Interrupt Request Acknowledge Processing Algorithm	517
22-14	Interrupt Request Acknowledge Timing (Minimum Time)	518
22-15	Interrupt Request Acknowledge Timing (Maximum Time)	518
22-16	Multiple Interrupt Example	520
22-17	Interrupt Request Hold	522
22-18	Basic Configuration of Test Function	523
22-19	Format of Interrupt Request Flag Register 1L	524
22-20	Format of Interrupt Mask Flag Register 1L	524
22-21	Key Return Mode Register Format	525
23-1	Memory Map when Using External Device Expansion Function	529
23-2	Memory Expansion Mode Register Format	531
23-3	Internal Memory Size Switching Register Format	532
23-4	External Bus Type Select Register Format	533
23-5	Instruction Fetch from External Memory in Multiplexed Bus Mode	535
23-6	External Memory Read Timing in Multiplexed Bus Mode	536
23-7	External Memory Write Timing in Multiplexed Bus Mode	537
23-8	External Memory Read Modify Write Timing in Multiplexed Bus Mode	538
23-9	Instruction Fetch from External Memory in Separate Bus Mode	540
23-10	External Memory Read Timing in Separate Bus Mode	541
23-11	External Memory Write Timing in Separate Bus Mode	542
23-12	External Memory Read Modify Write Timing in Separate Bus Mode	543
24-1	Oscillation Stabilization Time Select Register Format	546
24-2	HALT Mode Released by Interrupt Request Generation	548
24-3	HALT Mode Released by $\overline{\text{RESET}}$ Input	549
24-4	STOP Mode Released by Interrupt Request Generation	551
24-5	STOP Mode Released by $\overline{\text{RESET}}$ Input	552
25-1	Block Diagram of Reset Function	553
25-2	Timing of Reset by $\overline{\text{RESET}}$ Input	554
25-3	Timing of Reset due to Watchdog Timer Overflow	554
25-4	Timing of Reset by $\overline{\text{RESET}}$ Input in STOP Mode	554
26-1	Block Diagram of ROM Correction	559
26-2	Correction Address Registers 0 and 1 Format	560
26-3	Correction Control Register Format	561
26-4	Storing Example to EEPROM (when One Place is Corrected)	562
26-5	Connecting Example with EEPROM (Using 2-Wire Serial I/O Mode)	562
26-6	Initialization Routine	563
26-7	ROM Correction Operation	564
26-8	ROM Correction Example	565
26-9	Program Transition Diagram (when One Place is Corrected)	566
26-10	Program Transition Diagram (when Two Places are Corrected)	567

LIST OF FIGURES (9/9)

Figure No.	Title	Page
27-1	Internal Memory Size Switching Register Format	570
27-2	Internal Extension RAM Size Switching Register Format	571
27-3	Page Program Mode Flowchart	574
27-4	Page Program Mode Timing	575
27-5	Byte Program Mode Flowchart.....	576
27-6	Byte Program Mode Timing	577
27-7	PROM Read Timing	578
B-1	Development Tool Configuration	600
B-2	TGC-100SDW Drawing (For Reference Only)	610
B-3	EV-9200GF-100 Drawing (For Reference Only)	611
B-4	EV-9200GF-100 Recommended Footprints (For Reference Only)	612

LIST OF TABLES (1/3)

Table No.	Title	Page
1-1	Mask Options of Mask ROM Versions	47
1-2	Differences between μ PD78078 Subseries and μ PD78054 Subseries	47
2-1	Mask Options of Mask ROM Versions	63
2-2	Differences between μ PD78078Y Subseries and μ PD78054Y Subseries	63
3-1	Pin Input/Output Circuit Types	78
4-1	Pin Input/Output Circuit Types	96
5-1	Internal ROM Capacities	104
5-2	Vector Table	105
5-3	Special Function Register List	115
6-1	Port Functions (μ PD78078 Subseries)	132
6-2	Port Functions (μ PD78078Y Subseries)	134
6-3	Port Configuration	136
6-4	Pull-up Resistor Options for Port 6	146
6-5	Pull-up Resistor Options for Port 9	151
6-6	Port Mode Register and Output Latch Settings when Using Alternate Function	158
6-7	Comparison between Mask ROM Version and the μ PD78P078 and 78P078Y	164
7-1	Clock Generator Configuration	166
7-2	Relationship between CPU Clock and Minimum Instruction Execution Time	169
7-3	Maximum Time Required for CPU Clock Switchover	178
8-1	Timer/Event Counter Operations	182
8-2	16-Bit Timer/Event Counter Interval Times	183
8-3	16-Bit Timer/Event Counter Square-Wave Output Ranges	184
8-4	16-Bit Timer/Event Counter Configuration	185
8-5	INTP0/TI00 Pin Valid Edge and CR00 Capture Trigger Valid Edge	188
8-6	16-Bit Timer/Event Counter Interval Times	200
8-7	16-Bit Timer/Event Count Square-Wave Output Ranges	214
9-1	8-Bit Timer/Event Counters 1 and 2 Interval Times	224
9-2	8-Bit Timer/Event Counters 1 and 2 Square-Wave Output Ranges	225
9-3	Interval Times when 8-Bit Timer/Event Counters 1 and 2 are Used as 16-Bit Timer/Event Counters	226
9-4	Square-Wave Output Ranges when 8-Bit Timer/Event Counters 1 and 2 are Used as 16-Bit Timer/Event Counters	227
9-5	8-Bit Timer/Event Counters 1 and 2 Configurations	228
9-6	8-Bit Timer/Event Counter 1 Interval Time	237
9-7	8-Bit Timer/Event Counter 2 Interval Time	238
9-8	8-Bit Timer/Event Counters 1 and 2 Square-Wave Output Ranges	240

LIST OF TABLES (2/3)

Table No.	Title	Page
9-9	Interval Times when 2-Channel 8-Bit Timer/Event Counters (TM1 and TM2) are Used as 16-Bit Timer/Event Counter	243
9-10	Square-Wave Output Ranges when 2-Channel 8-Bit Timer/Event Counters (TM1 and TM2) are Used as 16-Bit Timer/Event Counter	245
10-1	8-Bit Timer/Event Counters 5 and 6 Interval Times	250
10-2	8-Bit Timer/Event Counters 5 and 6 Square-Wave Output Ranges	251
10-3	8-Bit Timer/Event Counters 5 and 6 Configurations	252
10-4	8-Bit Timer/Event Counters 5 and 6 Interval Times	260
10-5	8-Bit Timer/Event Counters 5 and 6 Square-Wave Output Ranges	263
11-1	Interval Timer Interval Time	269
11-2	Watch Timer Configuration	270
11-3	Interval Timer Interval Time	274
12-1	Watchdog Timer Runaway Detection Times	275
12-2	Interval Times	276
12-3	Watchdog Timer Configuration	277
12-4	Watchdog Timer Runaway Detection Time	281
12-5	Interval Timer Interval Time	282
13-1	Clock Output Control Circuit Configuration	284
14-1	Buzzer Output Control Circuit Configuration	289
15-1	A/D Converter Configuration	293
16-1	D/A Converter Configuration	310
17-1	Differences between Channels 0, 1, and 2	315
17-2	Serial Interface Channel 0 Configuration	318
17-3	Various Signals in SBI Mode	348
18-1	Differences between Channels 0, 1, and 2	365
18-2	Serial Interface Channel 0 Configuration	368
18-3	Serial Interface Channel 0 Interrupt Request Signal Generation	371
18-4	Signals in I ² C Bus Mode	398
19-1	Serial Interface Channel 1 Configuration	416
19-2	Interval Timing through CPU Processing (when the Internal Clock is Operating)	455
19-3	Interval Timing through CPU Processing (when the External Clock is Operating)	456
20-1	Serial Interface Channel 2 Configuration	458
20-2	Serial Interface Channel 2 Operating Mode Settings	463

LIST OF TABLES (3/3)

Table No.	Title	Page
20-3	Relationship between Main System Clock and Baud Rate	467
20-4	Relationship between ASCK Pin Input Frequency and Baud Rate (When BRGC is set to 00H)	468
20-5	Relationship between Main System Clock and Baud Rate	476
20-6	Relationship between ASCK Pin Input Frequency and Baud Rate (When BRGC is set to 00H)	477
20-7	Receive Error Causes	482
21-1	Real-time Output Port Configuration	495
21-2	Operation in Real-time Output Buffer Register Manipulation	496
21-3	Real-time Output Port Operating Mode and Output Trigger	498
22-1	Interrupt Source List	500
22-2	Various Flags Corresponding to Interrupt Request Sources	504
22-3	Times from Maskable Interrupt Request Generation to Interrupt Service	516
22-4	Interrupt Request Enabled for Multiple Interrupt during Interrupt Servicing	519
22-5	Test Input Factors	523
22-6	Flags Corresponding to Test Input Signals	523
23-1	Pin Functions in External Memory Expansion Mode	527
23-2	State of Port 4 to Port 6 Pins in External Memory Expansion Mode	527
23-3	Pin Functions in Separate Bus Mode	528
23-4	State of Port 4 to Port 6 and Port 8 Pins in Separate Bus Mode	528
23-5	Values when the Internal Memory Size Switching Register is Reset	532
24-1	HALT Mode Operating Status	547
24-2	Operation after HALT Mode Release	549
24-3	STOP Mode Operating Status	550
24-4	Operation after STOP Mode Release	552
25-1	Hardware Status after Reset	555
26-1	ROM Correction Configuration	559
27-1	Differences between PROM and Mask ROM Versions	569
27-2	Examples of Internal Memory Size Switching Register Settings	570
27-3	Examples of Internal Extension RAM Size Switching Register Settings	571
27-4	PROM Programming Operating Modes	572
A-1	Major Differences between μ PD78078, 78075B Subseries, and μ PD78070A	597
B-1	OS for IBM PC	609
B-2	System Upgrading from Former-type In-circuit Emulator for 78K/0 Series to IE-78001-R-A	609

CHAPTER 1 OUTLINE (μ PD78078 SUBSERIES)

1.1 Features

- Internal high-capacity ROM and RAM

Part Number \ Type	Program Memory (ROM)	Data Memory		
		Internal High-Speed RAM	Internal Buffer RAM	Internal Expansion RAM
μ PD78076	48 Kbytes	1024 bytes	32 bytes	1024 bytes
μ PD78078	60 Kbytes			1024 bytes ^{Note 2}
μ PD78P078	60 Kbytes ^{Note 1}			

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

2. The capacity of internal high-speed RAM can be changed by means of the internal expansion RAM size switching register (IXS).

- External Memory Expansion Space: 64 Kbytes
- Minimum instruction execution time changeable from high speed (0.4 μ s: @ 5.0-MHz operation with main system clock) to ultra-low speed (122 μ s: @ 32.768-kHz operation with subsystem clock)
- Instruction set suited to system control
 - Bit manipulation possible in all address spaces
 - Multiply and divide instructions incorporated
- 88 I/O port pins: (including eight N-ch open-drain port pins)
- 8-bit resolution A/D converter: 8 channels
- 8-bit resolution D/A converter: 2 channels
- Serial interface: Three channels
 - 3-wire serial I/O/SBI/2-wire serial I/O mode: 1 channel
 - 3-wire serial I/O mode (automatic transmit/receive function): 1 channel
 - 3-wire serial I/O/UART mode: 1 channel
- Timer: Seven channels
 - 16-bit timer/event counter : 1 channel
 - 8-bit timer/event counter : 4 channels
 - Watch timer : 1 channel
 - Watchdog timer : 1 channel
- 24 vectored interrupt sources
- Two test inputs
- Two types of on-chip clock oscillator circuits (main system clock and subsystem clock)
- Power supply voltage: 1.8 to 5.5 V

1.2 Application Fields

Cellular phones, cordless telephones, printers, AV equipment, air conditioners, cameras, PPCs, fuzzy home appliances, vending machines, etc.

1.3 Ordering Information

Part number	Package	Internal ROM
μ PD78076GC-xxx-7EA	100-pin plastic QFP (Fine pitch) (14 x 14 mm, resin thickness 1.45 mm)	Mask ROM
μ PD78076GC-xxx-8EU ^{Note}	100-pin plastic LQFP (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm)	Mask ROM
μ PD78076GF-xxx-3BA	100-pin plastic QFP (14 x 20 mm, resin thickness 2.7 mm)	Mask ROM
μ PD78078GC-xxx-7EA	100-pin plastic QFP (Fine pitch) (14 x 14 mm, resin thickness 1.45 mm)	Mask ROM
μ PD78078GC-xxx-8EU ^{Note}	100-pin plastic LQFP (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm)	Mask ROM
μ PD78078GF-xxx-3BA	100-pin plastic QFP (14 x 20 mm, resin thickness 2.7 mm)	Mask ROM
μ PD78P078GC-7EA	100-pin plastic QFP (Fine pitch) (14 x 14 mm, resin thickness 1.45 mm)	One-time PROM
μ PD78P078GC-8EU ^{Note}	100-pin plastic LQFP (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm)	One-time PROM
μ PD78P078GF-3BA	100-pin plastic QFP (14 x 20 mm, resin thickness 2.7 mm)	One-time PROM
μ PD78P078KL-T	100-pin ceramic WQFN (14 x 20 mm)	EPROM

Note Under development

Caution Two types of packages are available for the μ PD78076GC, 78078GC, and 78P078GC. For the suppliable package, contact an NEC sales representative.

Remark xxx indicates ROM code suffix.

1.4 Quality Grade

Part number	Package	Quality grades
μ PD78076GC-xxx-7EA	100-pin plastic QFP (Fine pitch) (14 x 14 mm, resin thickness 1.45 mm)	Standard
μ PD78076GC-xxx-8EU ^{Note}	100-pin plastic LQFP (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm)	Standard
μ PD78076GF-xxx-3BA	100-pin plastic QFP (14 x 20 mm, resin thickness 2.7 mm)	Standard
μ PD78078GC-xxx-7EA	100-pin plastic QFP (Fine pitch) (14 x 14 mm, resin thickness 1.45 mm)	Standard
μ PD78078GC-xxx-8EU ^{Note}	100-pin plastic LQFP (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm)	Standard
μ PD78078GF-xxx-3BA	100-pin plastic QFP (14 x 20 mm, resin thickness 2.7 mm)	Standard
μ PD78P078GC-7EA	100-pin plastic QFP (Fine pitch) (14 x 14 mm, resin thickness 1.45 mm)	Standard
μ PD78P078GC-8EU ^{Note}	100-pin plastic LQFP (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm)	Standard
μ PD78P078GF-3BA	100-pin plastic QFP (14 x 20 mm, resin thickness 2.7 mm)	Standard
μ PD78P078KL-T	100-pin ceramic WQFN (14 x 20 mm)	Not assured

(for function evaluation only)

Note Under development

Caution Of the above members, the following device with the suffix KL-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.

- μ PD78P078KL-T

Remark xxx indicates ROM code suffix.

Please refer to "Quality Grades on NEC Semiconductor Devices"(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

100-pin plastic QFP (Fine pitch) (14 x 14 mm, resin thickness 1.45 mm)

μ PD78076GC-xxx-7EA, 78078GC-xxx-7EA

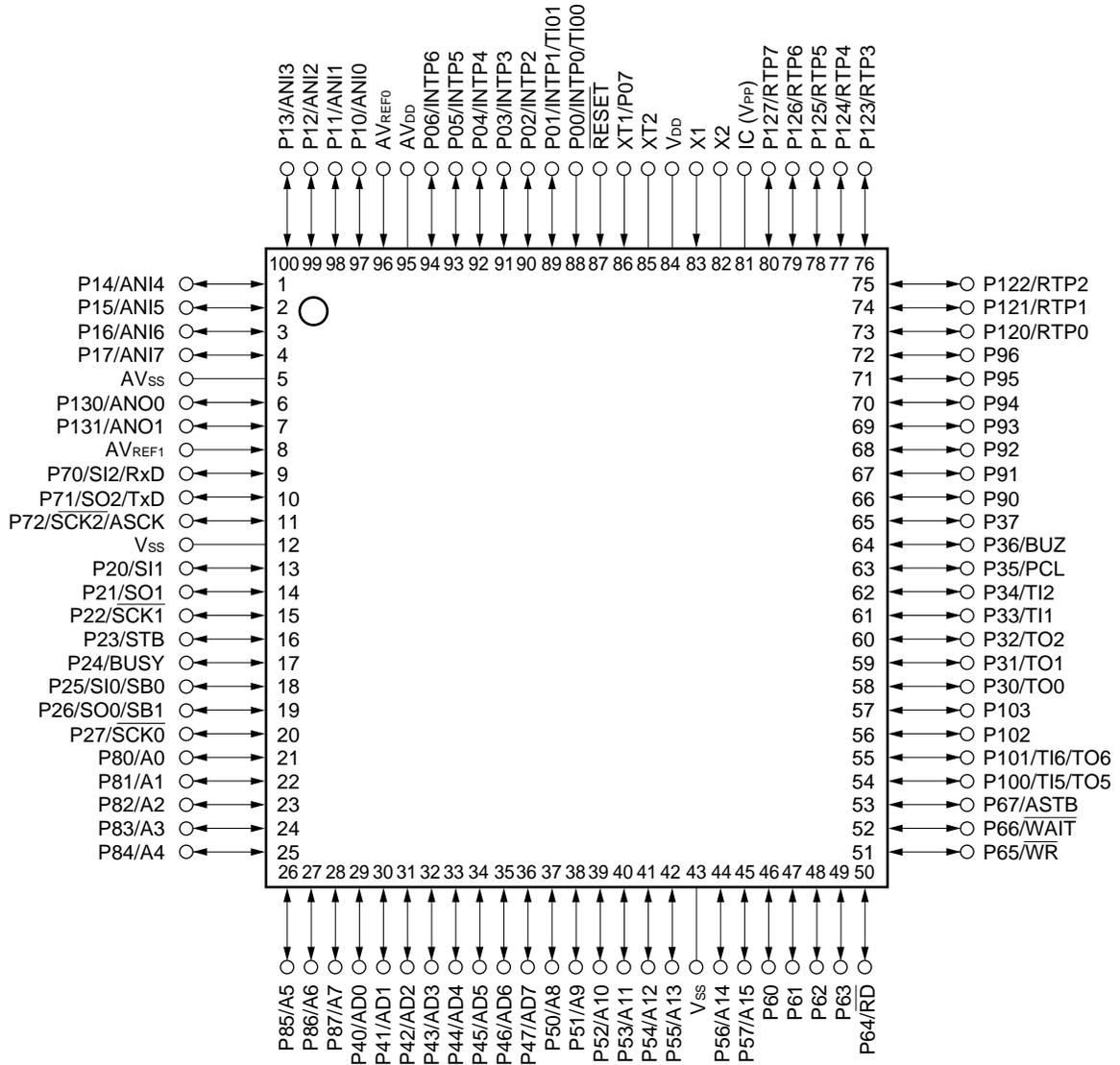
μ PD78P078GC-7EA

100-pin plastic LQFP (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm)

μ PD78076GC-xxx-8EU^{Note}, 78078GC-xxx-8EU^{Note}

μ PD78P078GC-8EU^{Note}

Note Under development



- Cautions**
1. Connect IC (Internally Connected) pin to Vss directly.
 2. Connect AVDD pin to VDD.
 3. Connect AVSS pin to Vss.

Remark Pin connection in parentheses is for the μ PD78P078.

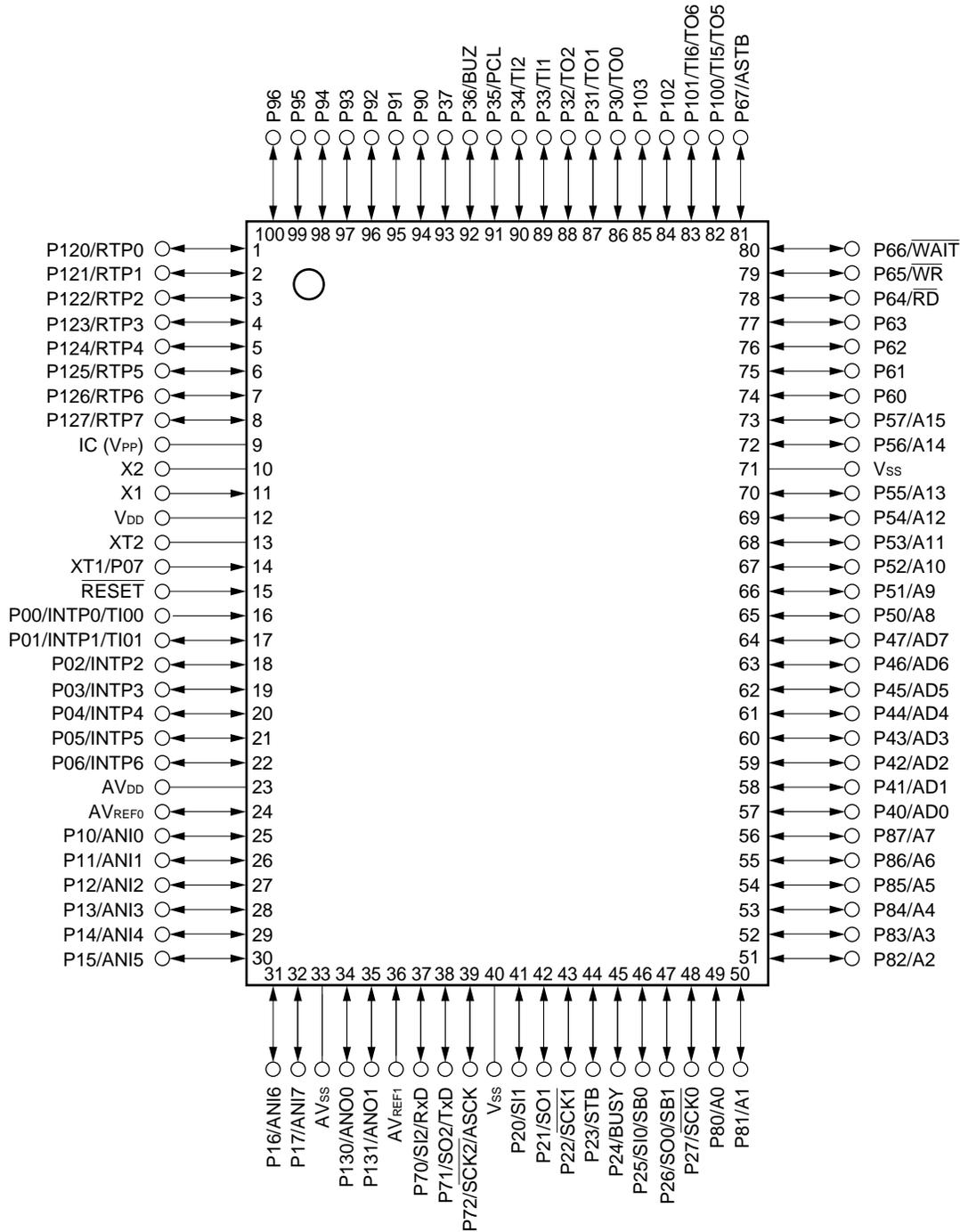
100-pin plastic QFP (14 x 20 mm)

μ PD78076GF-xxx-3BA, 78078GF-xxx-3BA

μ PD78P078GF-3BA

100-pin ceramic WQFN (14 x 20 mm)

μ PD78P078KL-T



- Cautions**
1. Connect IC (Internally Connected) pin to Vss directly.
 2. Connect AVDD pin to VDD.
 3. Connect AVSS pin to Vss.

Remark Pin connection in parentheses is for the μ PD78P078.

Pin Identifications

A0 to A15	: Address Bus	P120 to P127	: Port 12
AD0 to AD7	: Address/Data Bus	P130, P131	: Port 13
ANI0 to ANI7	: Analog Input	PCL	: Programmable Clock
ANO0, ANO1	: Analog Output	\overline{RD}	: Read Strobe
ASCK	: Asynchronous Serial Clock	\overline{RESET}	: Reset
ASTB	: Address Strobe	RTP0 to RTP7	: Real-time Output Port
AV _{DD}	: Analog Power Supply	RxD	: Receive Data
AV _{REF0} , AV _{REF1}	: Analog Reference Voltage	SB0, SB1	: Serial Bus
AV _{SS}	: Analog Ground	$\overline{SCK0}$ to $\overline{SCK2}$: Serial Clock
BUSY	: Busy	SI0 to SI2	: Serial Input
BUZ	: Buzzer Clock	SO0 to SO2	: Serial Output
IC	: Internally Connected	STB	: Strobe
INTP0 to INTP6	: Interrupt from Peripherals	TI00, TI01	: Timer Input
P00 to P07	: Port 0	TI1, TI2, TI5, TI6	: Timer Input
P10 to P17	: Port 1	TO0 to TO2, TO5, TO6	: Timer Output
P20 to P27	: Port 2	TxD	: Transmit Data
P30 to P37	: Port 3	V _{DD}	: Power Supply
P40 to P47	: Port 4	V _{PP}	: Programming Power Supply
P50 to P57	: Port 5	V _{SS}	: Ground
P60 to P67	: Port 6	\overline{WAIT}	: Wait
P70 to P72	: Port 7	\overline{WR}	: Write Strobe
P80 to P87	: Port 8	X1, X2	: Crystal (Main System Clock)
P90 to P96	: Port 9	XT1, XT2	: Crystal (Subsystem Clock)
P100 to P103	: Port 10		

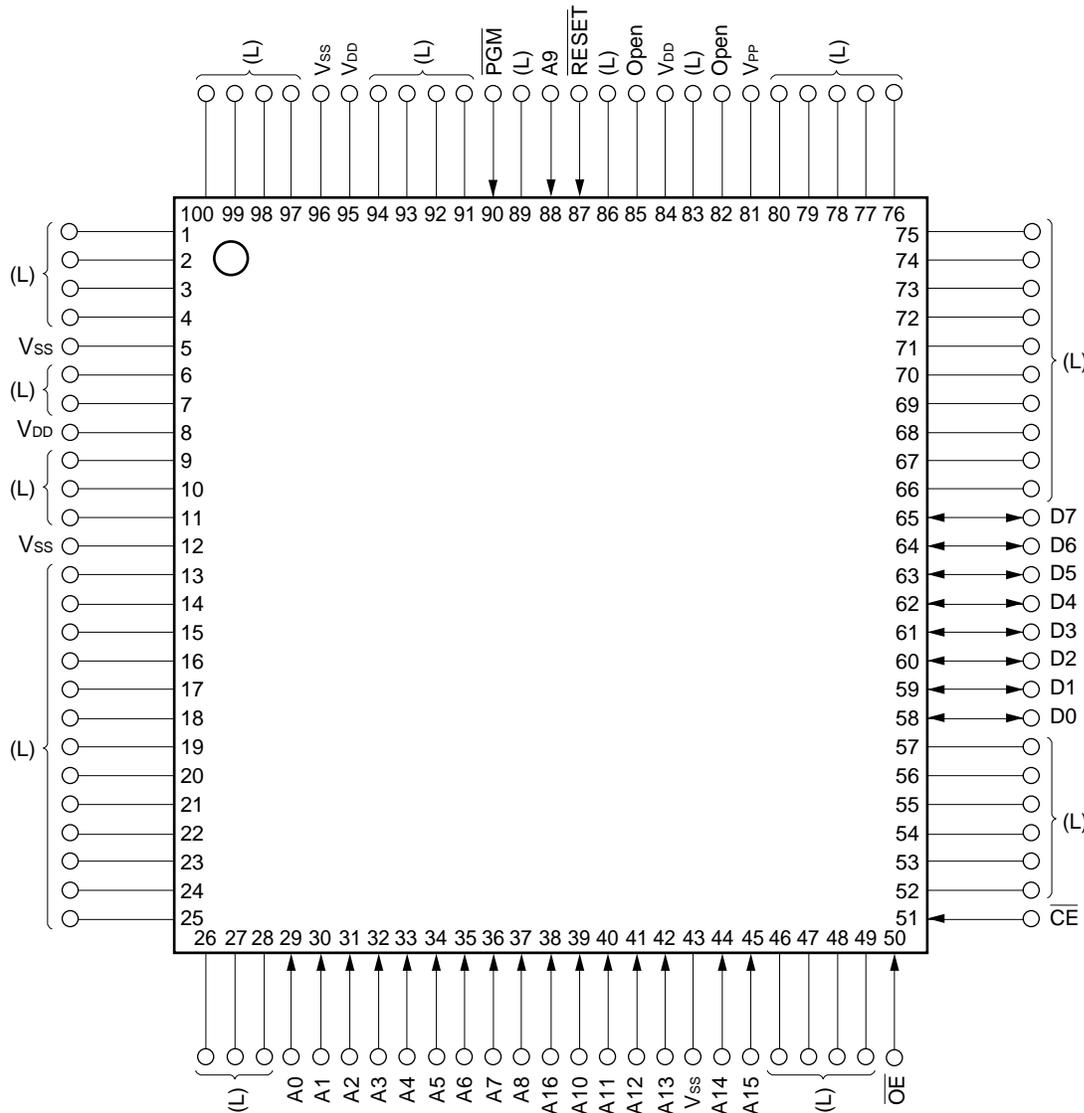
(2) PROM programming mode

100-pin plastic QFP (Fine pitch) (14 x 14 mm, resin thickness 1.45 mm)

μ PD78P078GC-7EA

100-pin plastic LQFP (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm)

μ PD78P078GC-8EU^{Note}



Note Under development

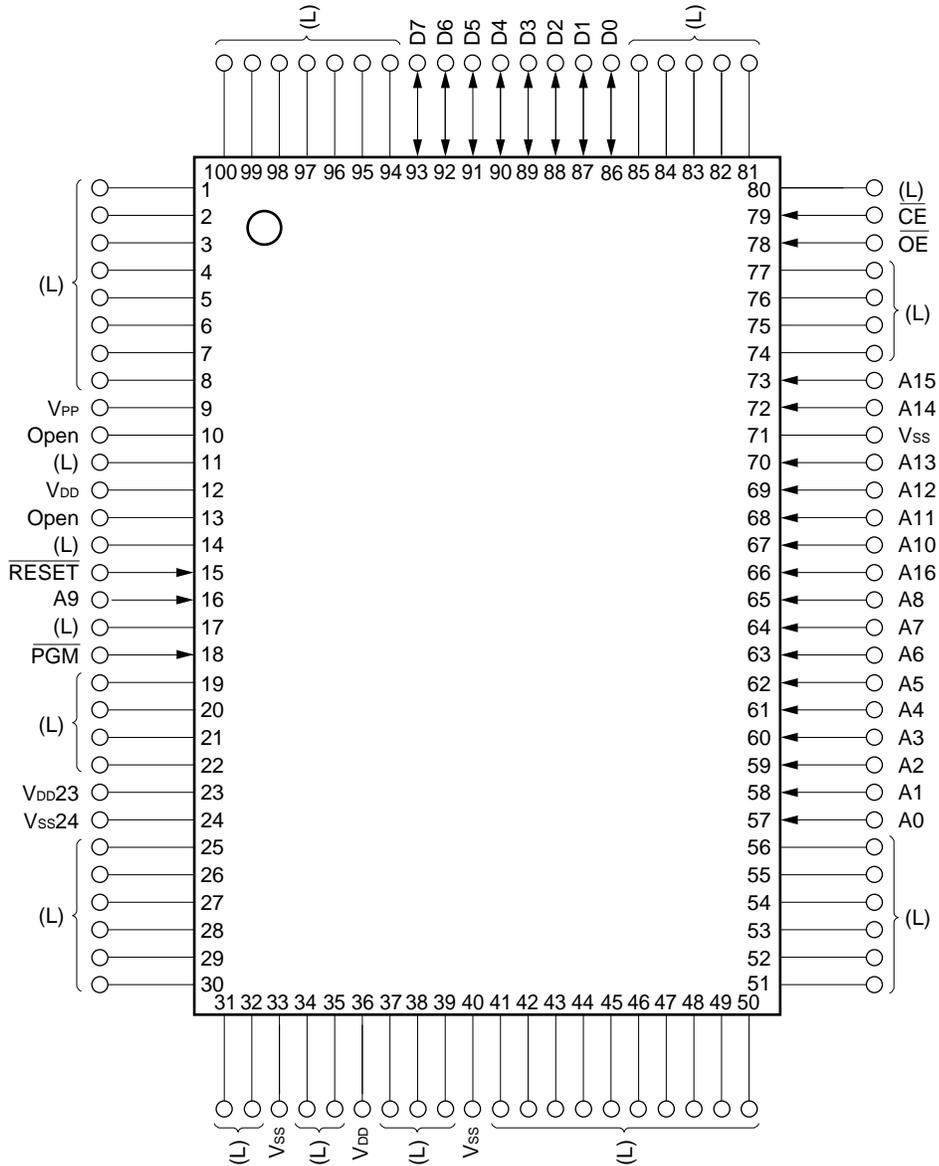
- Cautions**
1. (L) : Connect independently to Vss via a pull-down resistor.
 2. Vss : Connect to the ground.
 3. $\overline{\text{RESET}}$: Set to the low level.
 4. Open : Leave open.

100-pin plastic QFP (14 x 20 mm)

μ PD78P078GF-3BA

100-pin ceramic WQFN

μ PD78P078KL-T

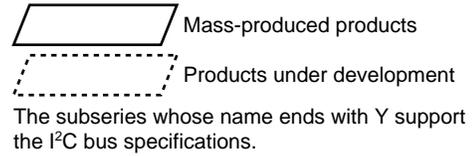


- Cautions**
1. (L) : Connect independently to V_{SS} via a pull-down resistor.
 2. V_{SS} : Connect to the ground.
 3. \overline{RESET} : Set to the low level.
 4. Open : Leave open.

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

★ 1.6 78K/0 Series Expansion

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



Control		
100-pin	μ PD78075B	Reduced EMI noise version of μ PD78078
100-pin	μ PD78078	Added timers to μ PD78054 and enhanced external interface
100-pin	μ PD78070A	ROM-less version of μ PD78078
100-pin	μ PD78070AY	Enhanced serial I/O of μ PD78078Y and functions are defined.
80-pin	μ PD780058	Enhanced serial I/O of μ PD78054, reduced EMI noise version
80-pin	μ PD780058Y	Enhanced serial I/O of μ PD78054, reduced EMI noise version
80-pin	μ PD78058F	Reduced EMI noise version of μ PD78054
80-pin	μ PD78058FY	Reduced EMI noise version of μ PD78054
80-pin	μ PD78054	Added UART and D/A to μ PD78014 and enhanced I/Os
80-pin	μ PD78054Y	Added UART and D/A to μ PD78014 and enhanced I/Os
64-pin	μ PD780034	Enhanced A/D of μ PD780024
64-pin	μ PD780034Y	Enhanced A/D of μ PD780024
64-pin	μ PD780024	Enhanced serial I/O of μ PD78018F
64-pin	μ PD780024Y	Enhanced serial I/O of μ PD78018F
64-pin	μ PD78014H	Reduced EMI noise version of μ PD78018F
64-pin	μ PD78018F	Low-voltage (1.8 V) version of μ PD78014 and enhanced ROM/RAM size options
64-pin	μ PD78018FY	Low-voltage (1.8 V) version of μ PD78014 and enhanced ROM/RAM size options
64-pin	μ PD78014	Added A/D and 16-bit timer to μ PD78002
64-pin	μ PD78014Y	Added A/D and 16-bit timer to μ PD78002
64-pin	μ PD780001	Added A/D to μ PD78002
64-pin	μ PD78002	Basic subseries for control applications
64-pin	μ PD78002Y	Basic subseries for control applications
42-/44-pin	μ PD78083	Equipped with UART and operates at low-voltage (1.8 V)
Inverter control		
64-pin	μ PD780988	Enhanced inverter control, timer, and SIO of μ PD780964, expanded ROM and RAM
64-pin	μ PD780964	Enhanced A/D of μ PD780924
64-pin	μ PD780924	Equipped with inverter control circuit and UART, reduced EMI noise version
FIP™ driving		
100-pin	μ PD780208	Enhanced I/O and FIP C/D of μ PD78044F, 53 display outputs
100-pin	μ PD780228	Enhanced I/O and FIP C/D of μ PD78044H, 48 display outputs
80-pin	μ PD78044H	Added N-ch open-drain I/O to μ PD78044F, 34 display outputs
80-pin	μ PD78044F	Basic subseries for driving FIPs, 34 display outputs
LCD driving		
100-pin	μ PD780308	Enhanced SIO of μ PD78064, expanded ROM and RAM
100-pin	μ PD780308Y	Enhanced SIO of μ PD78064, expanded ROM and RAM
100-pin	μ PD78064B	Reduced EMI noise version of μ PD78064
100-pin	μ PD78064	Basic subseries for driving LCDs, equipped with UART
100-pin	μ PD78064Y	Basic subseries for driving LCDs, equipped with UART
IEBus™ supported		
80-pin	μ PD78098B	Reduced EMI noise version of μ PD78098
80-pin	μ PD78098	Added IEBus controller to μ PD78054
Meter control		
80-pin	μ PD780973	Equipped with controller/driver for driving automobile meters

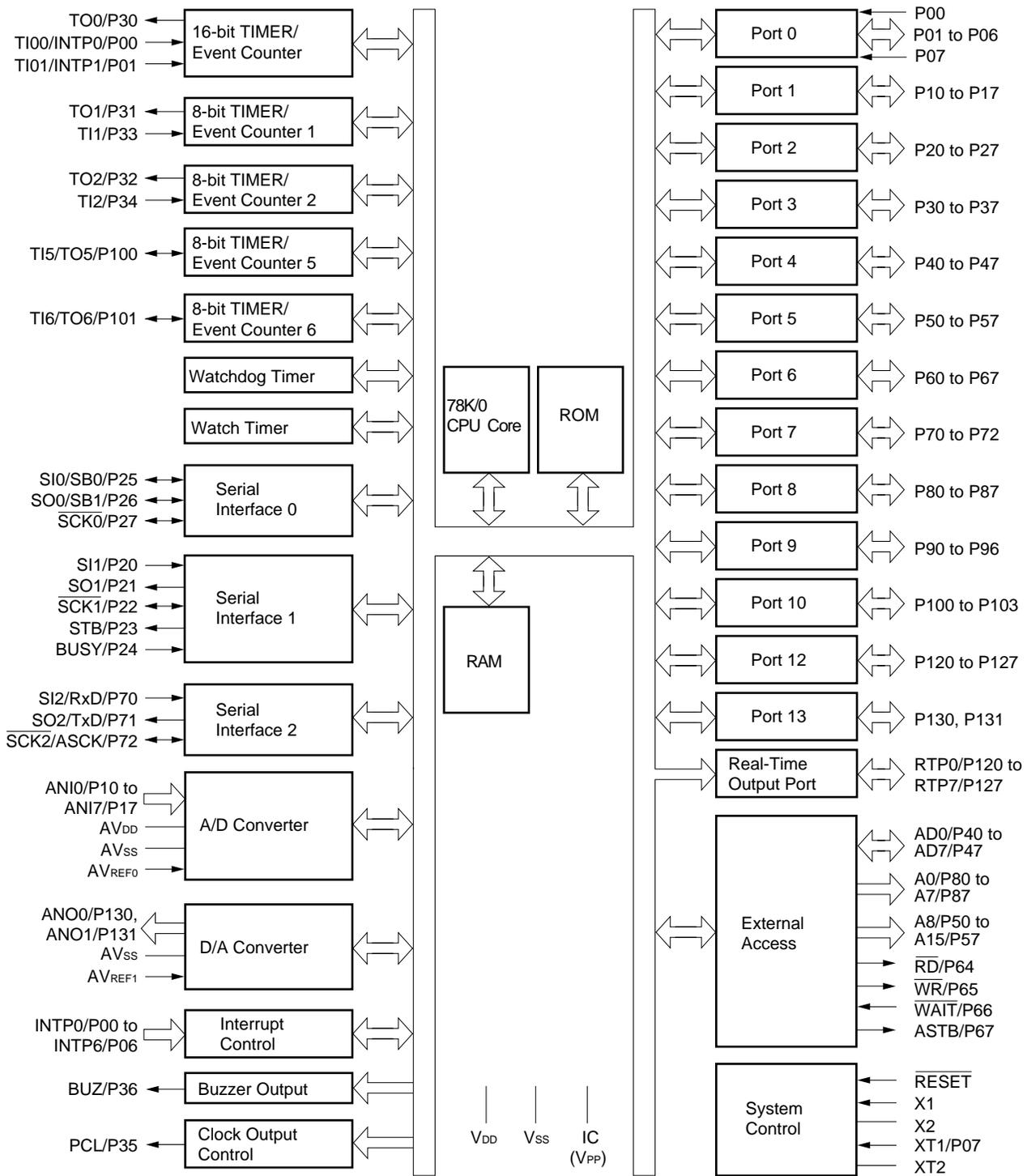
Note Planned

The following shows the major differences between subseries products.

Function Subseries Name		ROM Capacity	Timer				8-bit	10-bit	8-bit	Serial Interface	I/O	V _{DD} MIN. Value	External Expansion							
			8-bit	16-bit	Watch	WDT	A/D	A/D	D/A											
Control	μ PD78075B	32 K to 40 K	4 ch	1 ch	1 ch	1 ch	8 ch	-	2 ch	3 ch (UART: 1 ch)	88	1.8 V	√							
	μ PD78078	48 K to 60 K									61	2.7 V								
	μ PD78070A	-																		
	μ PD780058	24 K to 60 K	2 ch								3 ch (Time division UART: 1 ch)	68	1.8 V							
	μ PD78058F	48 K to 60 K									3 ch (UART: 1 ch)	69	2.7 V							
	μ PD78054	16 K to 60 K									2.0 V									
	μ PD780034	8 K to 32 K									-	8 ch	-		3 ch (UART: 1 ch, Time division 3-wire: 1 ch)	51	1.8 V			
	μ PD780024										8 ch	-	2 ch		53					
	μ PD78014H																			
	μ PD78018F	8 K to 60 K									1 ch									2.7 V
	μ PD78014	8 K to 32 K																		
	μ PD780001	8 K																		39
	μ PD78002	8 K to 16 K									53									
μ PD78083	8 K																			
	μ PD78002	8 K to 16 K																		
	μ PD78083					8 ch			1 ch (UART: 1 ch)	33	1.8 V	-								
Inverter control	μ PD780988	32 K to 60 K	3 ch	Note 1	-	1 ch	-	8 ch	-	3 ch (UART: 2 ch)	47	4.0 V	√							
	μ PD780964	8 K to 32 K									2 ch (UART: 2 ch)	2.7 V								
	μ PD780924																			
	μ PD780924					8 ch	-													
FIP driving	μ PD780208	32 K to 60 K	2 ch	1 ch	1 ch	1 ch	8 ch	-	-	2 ch	74	2.7 V	-							
	μ PD780228	48 K to 60 K									3 ch	-		-	1 ch	72	4.5 V			
	μ PD78044H	32 K to 48 K	2 ch	1 ch	1 ch	68										2.7 V				
	μ PD78044F	16 K to 40 K									2 ch									
LCD driving	μ PD780308	48 K to 60 K	2 ch	1 ch	1 ch	1 ch	8 ch	-	-	3 ch (Time division UART: 1 ch)	57	2.0 V	-							
	μ PD78064B	32 K												2 ch (UART: 1 ch)						
	μ PD78064	16 K to 32 K																		
IEBus supported	μ PD78098B	40 K to 60 K	2 ch	1 ch	1 ch	1 ch	8 ch	-	2 ch	3 ch (UART: 1 ch)	69	2.7 V	√							
	μ PD78098	32 K to 60 K																		
Meter control	μ PD780973	24 K to 32 K	3 ch	1 ch	1 ch	1 ch	5 ch	-	-	2 ch (UART: 1 ch)	56	4.5 V	-							

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

1.7 Block Diagram



- Remarks**
1. The internal ROM and RAM capacities depend on the product.
 2. Pin connection in parentheses is for the μ PD78P078.

1.8 Outline of Function

Item		Part Number	μ PD78076	μ PD78078	μ PD78P078
Internal memory	ROM		Mask ROM		PROM
			48 Kbytes	60 Kbytes	60 Kbytes ^{Note 1}
	High-speed RAM		1024 bytes		
	Buffer RAM		32 bytes		
	Expansion RAM		1024 bytes		1024 bytes ^{Note 2}
Memory space			64 Kbytes		
General register			8 bits x 8 x 4 banks		
Minimum instruction execution time	With main system clock selected		0.4 μ s/0.8 μ s/1.6 μ s/3.2 μ s/6.4 μ s/12.8 μ s (@ 5.0-MHz operation)		
	With subsystem clock selected		122 μ s (@ 32.768-kHz operation)		
Instruction set			<ul style="list-style-type: none"> • 16-bit operation • Multiply/divide (8-bit x 8-bit, 16-bit/8-bit) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc. 		
I/O port			Total : 88 <ul style="list-style-type: none"> • CMOS input : 2 • CMOS input/output : 78 • N-ch open drain I/O : 8 		
A/D converter			8-bit resolution x 8 channels		
D/A converter			8-bit resolution x 2 channels		
Serial interface			<ul style="list-style-type: none"> • 3-wire serial I/O/SBI/2-wire serial I/O mode selection possible : 1 channel • 3-wire serial I/O mode (Maximum 32-byte on-chip automatic transmit/receive function) : 1 channel • 3-wire serial I/O/UART mode selection possible : 1 channel 		
Timer			<ul style="list-style-type: none"> • 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 4 channels • Watch timer : 1 channel • Watchdog timer : 1 channel 		
Timer output			5 outputs: (14-bit PWM output enable: 1, 8-bit PWM output enable: 2)		
Clock output			19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (@ 5.0-MHz operation with main system clock) 32.768 kHz (@ 32.768-kHz operation with subsystem clock)		
Buzzer output			1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (@ 5.0-MHz operation with main system clock)		

- Notes**
1. The capacity of the internal PROM can be changed using the internal memory size switching register (IMS).
 2. The capacity of the internal expansion RAM can be changed using the internal expansion RAM size switching register (IXS).

Item		Part Number	μ PD78076	μ PD78078	μ PD78P078
Vectored interrupt source	Maskable		Internal: 15 External: 7		
	Non-maskable		Internal: 1		
	Software		Internal: 1		
Test input			Internal: 1 External: 1		
Power supply voltage			$V_{DD} = 1.8$ to 5.5 V		
Operating ambient temperature			$T_A = -40$ to $+85^\circ\text{C}$		
Package			<ul style="list-style-type: none"> • 100-pin plastic QFP (Fine pitch) (14 x 14 mm, resin thickness 1.45 mm) • 100-pin plastic LQFP^{Note} (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm) • 100-pin plastic QFP (14 x 20 mm, resin thickness 2.7 mm) • 100-pin ceramic WQFN (14 x 20 mm) (μPD78P078 only) 		

Note Under development

1.9 Mask Options

The mask ROM versions (μ PD78076, 78078) provide pull-up register mask options which allow users to specify whether to connect a pull-up register to a specific port pin when the user places an order for the device production. Using this mask option when pull-up resistors are required reduces the number of components to add to the device, resulting in board space saving.

The mask options provided in the μ PD78078 Subseries are shown in Table 1-1.

Table 1-1. Mask Options of Mask ROM Versions

Pin Names	Mask Options
P60 to P63, P90 to P93	Pull-up resistor connection can be specified in 1-bit units.

1.10 Differences with μ PD78054 Subseries

The μ PD78078 Subseries is upward-compatible with the μ PD78054 Subseries. The differences between the two subseries are shown in the table below. The functions and specifications other than those shown in this table are common to these two series.

Table 1-2. Differences between μ PD78078 Subseries and μ PD78054 Subseries

Item	μ PD78054 Subseries	μ PD78078 Subseries
No. of I/O ports	69	88
8-bit timer/event counter	2 channels	4 channels
External device expansion function	Address bus separate function is not provided.	Address bus separate function is provided. (P80/A0 to P87/A7)
Power supply voltage	V _{DD} = 2.0 to 6.0 V	V _{DD} = 1.8 to 5.5 V
Package	80-pin plastic TQFP (12 x 12 mm) 80-pin plastic QFP (14 x 14 mm) 80-pin plastic WQFP (14 x 14 mm)* * : Only for PROM version	100-pin plastic TQFP (14 x 14 mm) 100-pin plastic LQFP (14 x 14 mm) 100-pin plastic QFP (14 x 20 mm) 100-pin plastic WQFP (14 x 20 mm)* * : Only for PROM version

[MEMO]

CHAPTER 2 OUTLINE (μ PD78078Y SUBSERIES)

2.1 Features

- Internal high-capacity ROM and RAM

Part Number \ Type	Program Memory (ROM)	Data Memory		
		Internal High-Speed RAM	Internal Buffer RAM	Internal Expansion RAM
μ PD78076Y	48 Kbytes	1024 bytes	32 bytes	1024 bytes
μ PD78078Y	60 Kbytes			1024 bytes
μ PD78P078Y	60 Kbytes ^{Note 1}			1024 bytes ^{Note 2}

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

2. The capacity of internal high-speed RAM can be changed using the internal expansion RAM size switching register (IXS).

- External memory expansion space: 64 Kbytes
- Minimum instruction execution time changeable from high speed (0.4 μ s: @ 5.0-MHz operation with main system clock) to ultra-low speed (122 μ s: @ 32.768-kHz operation with subsystem clock)
- Instruction set suited to system control
 - Bit manipulation possible in all address spaces
 - Multiply and divide instructions incorporated
- 88 I/O port pins: (including eight N-ch open-drain port pins)
- 8-bit resolution A/D converter: 8 channels
- 8-bit resolution D/A converter: 2 channels
- Serial interface: Three channels
 - 3-wire serial I/O/2-wire serial I/O/I²C bus mode: 1 channel
 - 3-wire serial I/O mode (Automatic transmit/receive function): 1 channel
 - 3-wire serial I/O/UART mode: 1 channel
- Timer: Seven channels
 - 16-bit timer/event counter : 1 channel
 - 8-bit timer/event counter : 4 channels
 - Watch timer : 1 channel
 - Watchdog timer : 1 channel
- 24 vectored interrupt sources
- Two test inputs
- Two types of on-chip clock oscillation circuits (main system clock and subsystem clock)
- Power supply voltage: 1.8 to 5.5 V

2.2 Application Fields

Cellular phones, cordless telephones, printers, AV equipment, air conditioners, cameras, PPCs, fuzzy home appliances, vending machines, etc.

2.3 Ordering Information

	Part number	Package	Internal ROM
★	μ PD78076YGC-xxx-8EU ^{Note}	100-pin plastic LQFP (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm)	Mask ROM
	μ PD78076YGF-xxx-3BA	100-pin plastic QFP (14 x 20 mm, resin thickness 2.7 mm)	Mask ROM
★	μ PD78078YGC-xxx-8EU ^{Note}	100-pin plastic LQFP (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm)	Mask ROM
	μ PD78078YGF-xxx-3BA	100-pin plastic QFP (14 x 20 mm, resin thickness 2.7 mm)	Mask ROM
★	μ PD78P078YGC-8EU ^{Note}	100-pin plastic LQFP (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm)	One-time PROM
	μ PD78P078YGF-3BA	100-pin plastic QFP (14 x 20 mm, resin thickness 2.7 mm)	One-time PROM
	μ PD78P078YKL-T	100-pin ceramic WQFN (14 x 20 mm)	EPROM

Note Under development

Remark xxx indicates ROM code suffix.

2.4 Quality Grade

Part number	Package	Quality grades
μ PD78076YGC-xxx-8EU ^{Note}	100-pin plastic LQFP (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm)	Standard
μ PD78076YGF-xxx-3BA	100-pin plastic QFP (14 x 20 mm, resin thickness 2.7 mm)	Standard
μ PD78078YGC-xxx-8EU ^{Note}	100-pin plastic LQFP (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm)	Standard
μ PD78078YGF-xxx-3BA	100-pin plastic QFP (14 x 20 mm, resin thickness 2.7 mm)	Standard
μ PD78P078YGC-8EU ^{Note}	100-pin plastic LQFP (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm)	Standard
μ PD78P078YGF-3BA	100-pin plastic QFP (14 x 20 mm, resin thickness 2.7 mm)	Standard
μ PD78P078YKL-T	100-pin ceramic WQFN (14 x 20 mm)	Not assured (for function evaluation only)

Note Under development

Caution Of the above members, the following device with the suffix KL-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.

- μ PD78P078YKL-T

Remark xxx indicates ROM code suffix.

Please refer to "Quality Grades on NEC Semiconductor Devices"(C11531E) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

2.5 Pin Configuration (Top View)

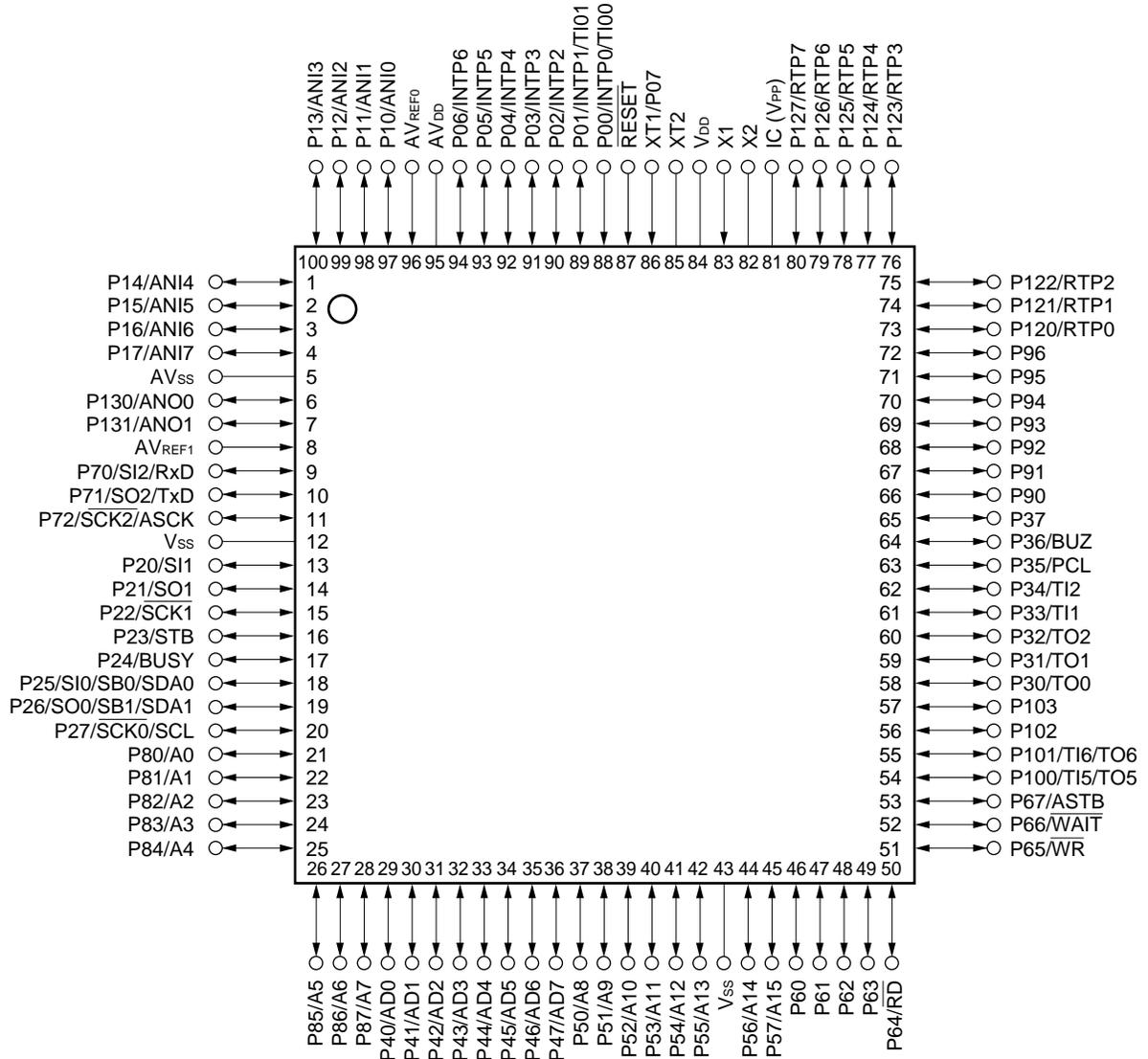
(1) Normal operating mode

100-pin plastic LQFP (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm)

μ PD78076YGC-xxx-8EU^{Note}, 78078YGC-xxx-8EU^{Note}

μ PD78P078YGC-8EU^{Note}

Note Under development



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

Remark Pin connection in parentheses is for the μPD78P078Y.

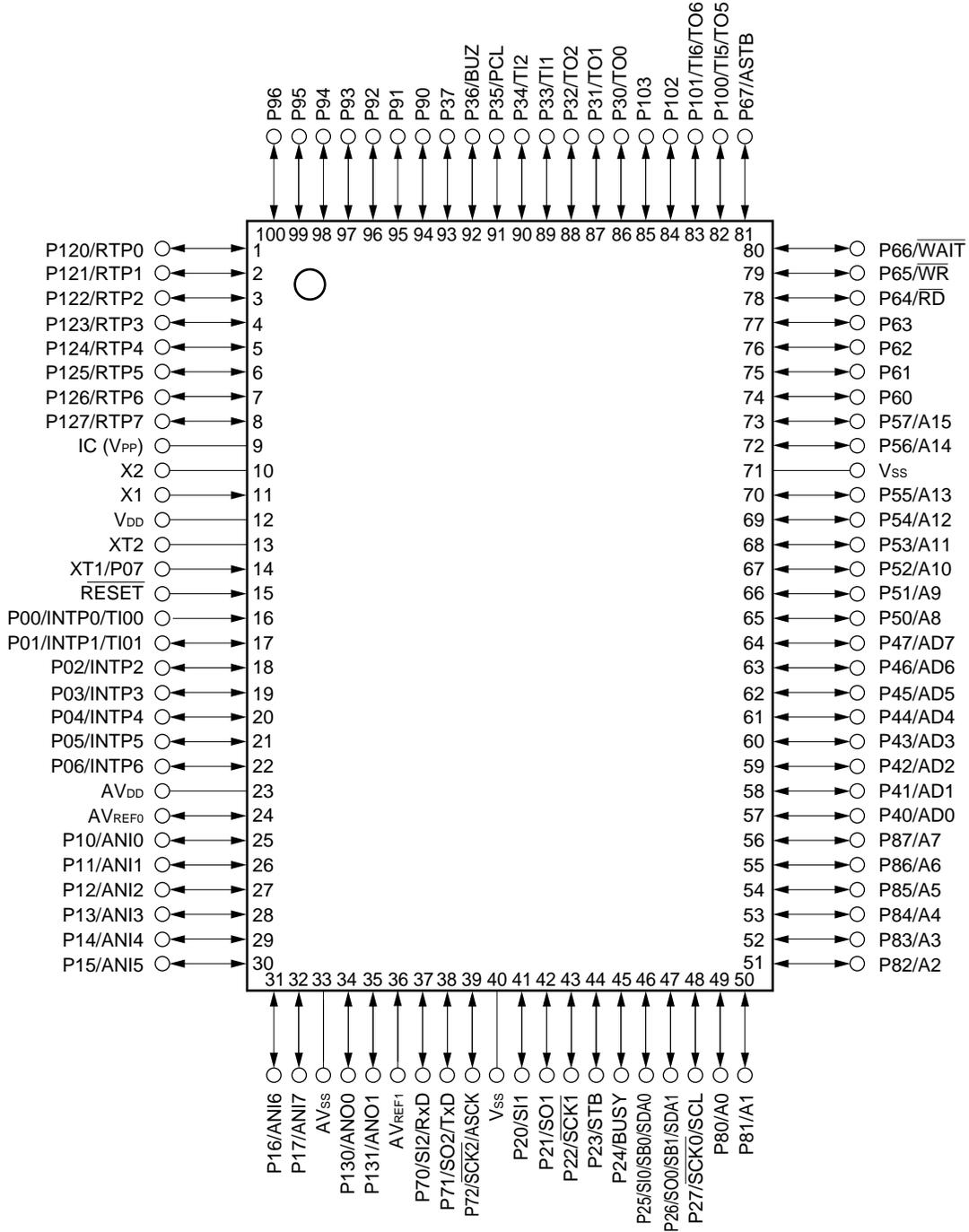
100-pin plastic QFP (14 x 20 mm)

μ PD78076YGF-xxx-3BA

μ PD78078YGF-xxx-3BA, 78P078YGF-3BA

100-pin ceramic WQFN (14 x 20 mm)

μ PD78P078YKL-T



Cautions 1. Connect IC (Internally Connected) pin to Vss directly.

2. Connect AVDD pin to VDD.

3. Connect AVss pin to Vss.

Remark Pin connection in parentheses is for the μ PD78P078Y.

Pin Identifications

A0 to A15	: Address Bus	P130, P131	: Port 13
AD0 to AD7	: Address/Data Bus	PCL	: Programmable Clock
ANI0 to ANI7	: Analog Input	\overline{RD}	: Read Strobe
ANO0, ANO1	: Analog Output	\overline{RESET}	: Reset
ASCK	: Asynchronous Serial Clock	RTP0 to RTP7	: Real-time Output Port
ASTB	: Address Strobe	RxD	: Receive Data
AV _{DD}	: Analog Power Supply	SB0, SB1	: Serial Bus
AV _{REF0} , AV _{REF1}	: Analog Reference Voltage	$\overline{SCK0}$ to $\overline{SCK2}$: Serial Clock
AV _{SS}	: Analog Ground	SCL	: Serial Clock
BUSY	: Busy	SDA0, SDA1	: Serial Data
BUZ	: Buzzer Clock	SI0 to SI2	: Serial Input
IC	: Internally Connected	SO0 to SO2	: Serial Output
INTP0 to INTP6	: Interrupt from Peripherals	STB	: Strobe
P00 to P07	: Port 0	TI00, TI01	: Timer Input
P10 to P17	: Port 1	TI1, TI2, TI5, TI6	: Timer Input
P20 to P27	: Port 2	TO0 to TO2, TO5, TO6	: Timer Output
P30 to P37	: Port 3	TxD	: Transmit Data
P40 to P47	: Port 4	V _{DD}	: Power Supply
P50 to P57	: Port 5	V _{PP}	: Programming Power Supply
P60 to P67	: Port 6	V _{SS}	: Ground
P70 to P72	: Port 7	\overline{WAIT}	: Wait
P80 to P87	: Port 8	\overline{WR}	: Write Strobe
P90 to P96	: Port 9	X1, X2	: Crystal (Main System Clock)
P100 to P103	: Port 10	XT1, XT2	: Crystal (Subsystem Clock)
P120 to P127	: Port 12		

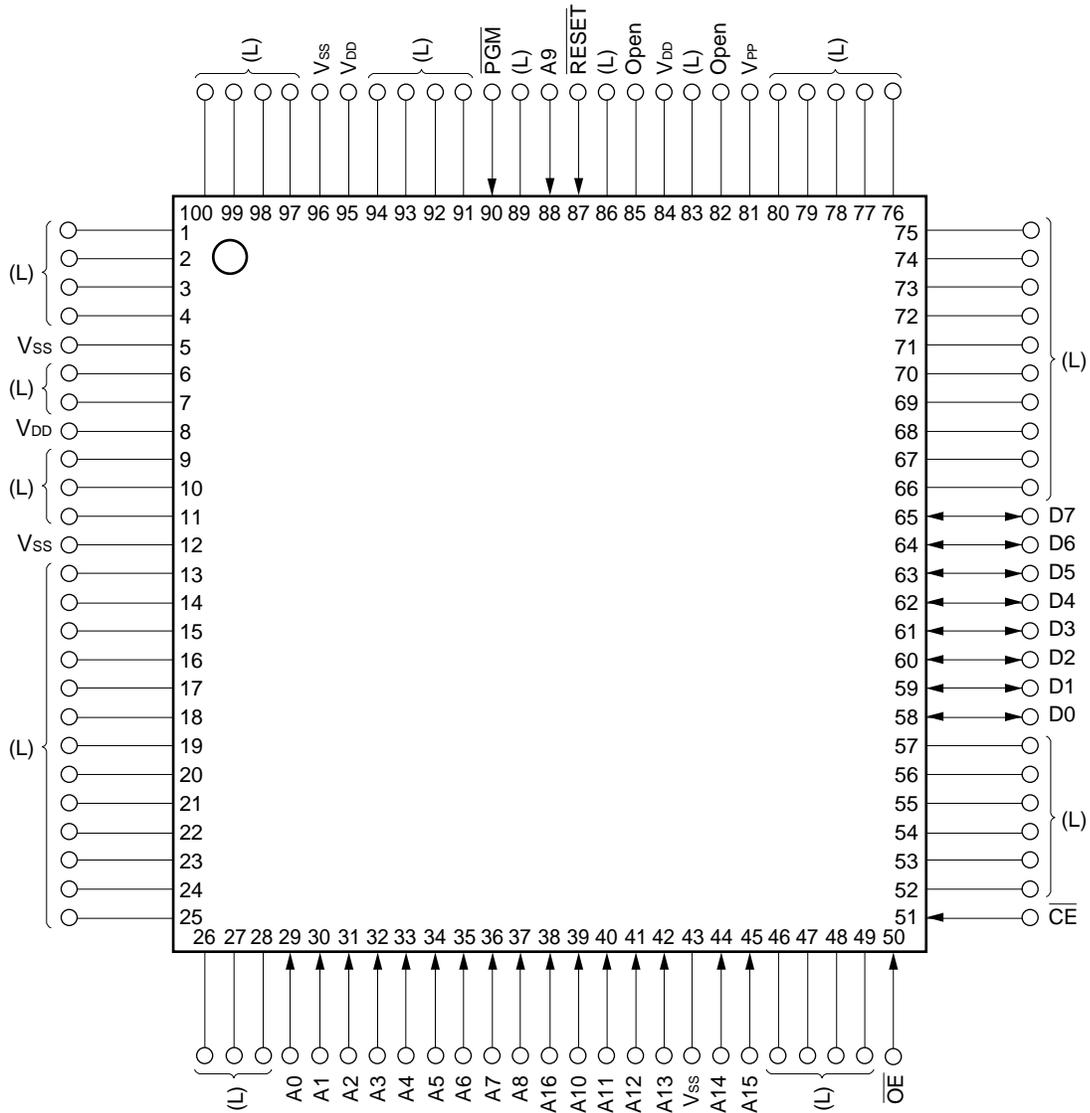
(2) PROM programming mode

100-pin plastic QFP (Fine pitch) (14 x 14 mm, resin thickness 1.45 mm)

μ PD78P078YGC-7EA

100-pin plastic LQFP (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm)

μ PD78P078YGC-8EU^{Note}



Note Under development

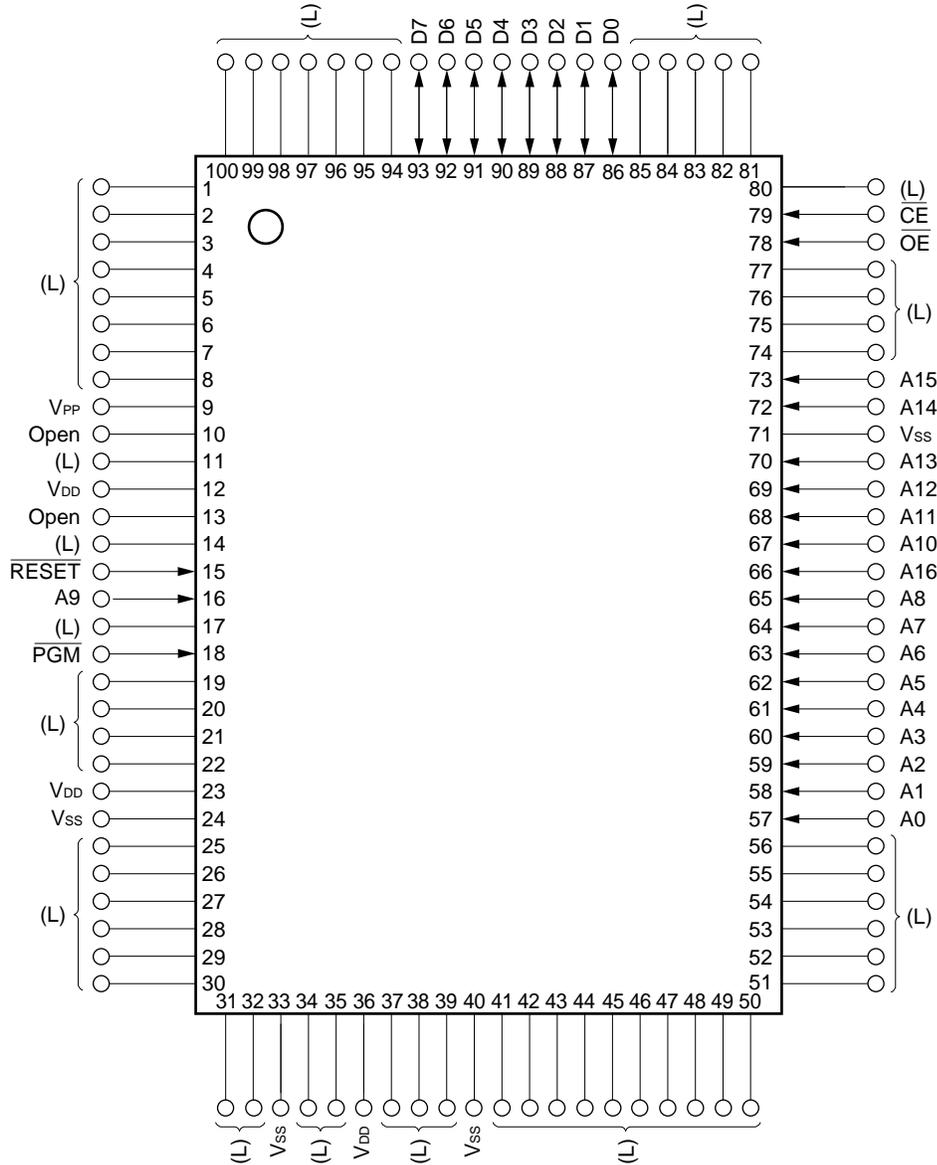
- Cautions**
1. (L) : Connect independently to Vss via a pull-down resistor.
 2. Vss : Connect to the ground.
 3. $\overline{\text{RESET}}$: Set to the low level.
 4. Open : Leave open.

100-pin plastic QFP (14 x 20 mm)

μ PD78P078YGF-3BA

100-pin ceramic WQFN

μ PD78P078YKL-T

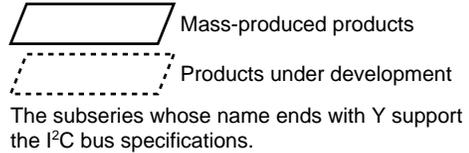


- Cautions**
1. (L) : Connect independently to V_{SS} via a pull-down resistor.
 2. V_{SS} : Connect to the ground.
 3. RESET : Set to the low level.
 4. Open : Leave open.

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

★ 2.6 78K/0 Series Expansion

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



78K/0 Series	Control		
	100-pin	μPD78075B	Reduced EMI noise version of μPD78078
	100-pin	μPD78078	Added timers to μPD78054 and enhanced external interface
	100-pin	μPD78070A	ROM-less version of μPD78078
	100-pin	μPD780018AY	Enhanced serial I/O of μPD78078Y and functions are defined.
	80-pin	μPD780058	Enhanced serial I/O of μPD78054, reduced EMI noise version
	80-pin	μPD78058F	Reduced EMI noise version of μPD78054
	80-pin	μPD78054	Added UART and D/A to μPD78014 and enhanced I/Os
	64-pin	μPD780034	Enhanced A/D of μPD780024
	64-pin	μPD780024	Enhanced serial I/O of μPD78018F
	64-pin	μPD78014H	Reduced EMI noise version of μPD78018F
	64-pin	μPD78018F	Low-voltage (1.8 V) version of μPD78014 and enhanced ROM/RAM size options
	64-pin	μPD78014	Added A/D and 16-bit timer to μPD78002
	64-pin	μPD780001	Added A/D to μPD78002
	64-pin	μPD78002	Basic subseries for control applications
	42-/44-pin	μPD78083	Equipped with UART and operates at low-voltage (1.8 V)
	Inverter control		
	64-pin	μPD780988	Enhanced inverter control, timer, and SIO of μPD780964, expanded ROM and RAM
	64-pin	μPD780964	Enhanced A/D of μPD780924
	64-pin	μPD780924	Equipped with inverter control circuit and UART, reduced EMI noise version
FIP driving			
100-pin	μPD780208	Enhanced I/O and FIP C/D of μPD78044F, 53 display outputs	
100-pin	μPD780228	Enhanced I/O and FIP C/D of μPD78044H, 48 display outputs	
80-pin	μPD78044H	Added N-ch open-drain I/O to μPD78044F, 34 display outputs	
80-pin	μPD78044F	Basic subseries for driving FIPs, 34 display outputs	
LCD driving			
100-pin	μPD780308	Enhanced SIO of μPD78064, expanded ROM and RAM	
100-pin	μPD78064B	Reduced EMI noise version of μPD78064	
100-pin	μPD78064	Basic subseries for driving LCDs, equipped with UART	
IEBus supported			
80-pin	μPD78098B	Reduced EMI noise version of μPD78098	
80-pin	μPD78098	Added IEBus controller to μPD78054	
Meter control			
80-pin	μPD780973	Equipped with controller/driver for driving automobile meters	

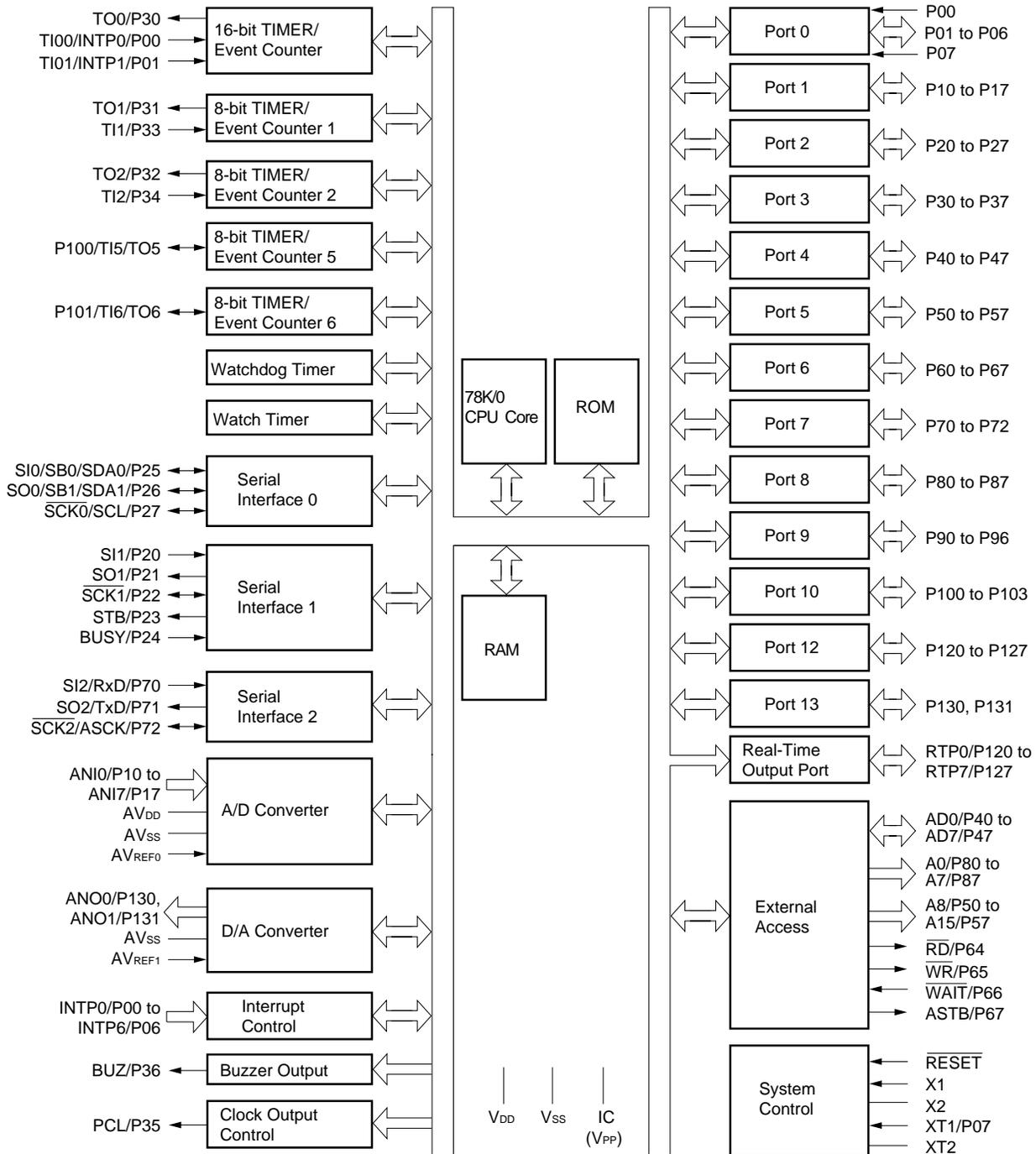
Note Planned

Major differences among Y subseries are tabulated below.

Function		ROM Capacity	Configuration of Serial Interface		I/O	V _{DD} MIN.
Subseries						
Control	μ PD78078Y	48K to 60K	3-wire/2-wire/I ² C	: 1 ch	88	1.8 V
	μ PD78070AY	—	3-wire with automatic transmit/receive function	: 1 ch		
	μ PD780018AY	48K to 60K	3-wire/UART	: 1 ch	61	2.7 V
			3-wire with automatic transmit/receive function	: 1 ch	88	
			Time division 3-wire	: 1 ch		
			I ² C bus (supports multi-master)	: 1 ch		
	μ PD780058Y	24K to 60K	3-wire/2-wire/I ² C	: 1 ch	68	1.8 V
			3-wire with automatic transmit/receive function	: 1 ch		
			3-wire/time division UART	: 1 ch		
	μ PD78058FY	48K to 60K	3-wire/2-wire/I ² C	: 1 ch	69	2.7 V
μ PD78054Y	16K to 60K	3-wire with automatic transmit/receive function	: 1 ch	2.0 V		
		3-wire/UART	: 1 ch			
μ PD780034Y	8K to 32K		UART	: 1 ch	51	1.8 V
μ PD780024Y			3-wire	: 1 ch		
			I ² C bus (supports multi-master)	: 1 ch		
μ PD78018FY	8K to 60K		3-wire/2-wire/I ² C	: 1 ch	53	2.7 V
			3-wire with automatic transmit/receive function	: 1 ch		
μ PD78014Y	8K to 32K		3-wire/2-wire/I ² C	: 1 ch		
			3-wire with automatic transmit/receive function	: 1 ch		
μ PD78002Y	8K to 16K		3-wire/2-wire/SBI/I ² C	: 1 ch		
LCD drive	μ PD780308Y	48K to 60K	3-wire/2-wire/I ² C	: 1 ch	57	2.0 V
			3-wire/time division UART	: 1 ch		
			3-wire	: 1 ch		
μ PD78064Y	16K to 32K		3-wire/2-wire/I ² C	: 1 ch		
			3-wire/UART	: 1 ch		

Remark The functions except serial interface are common with subseries without Y.

2.7 Block Diagram



- Remarks**
1. The internal ROM and RAM capacities depend on the product.
 2. Pin connection in parentheses is for the μ PD78P078Y.

2.8 Outline of Function

Item		Part Number	μPD78076Y	μPD78078Y	μPD78P078Y
Internal memory	ROM	Mask ROM			PROM
		48 Kbytes	60 Kbytes	60 Kbytes ^{Note 1}	
	High-speed RAM	1024 bytes			
	Buffer RAM	32 bytes			
	Expansion RAM	1024 bytes			1024 bytes
Memory space		64 Kbytes			
General register		8 bits x 8 x 4 banks			
Minimum instruction execution time	With main system clock selected	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (@ 5.0-MHz operation)			
	With subsystem clock selected	122 μs (@ 32.768-kHz operation)			
Instruction set		<ul style="list-style-type: none"> • 16-bit operation • Multiply/divide (8-bit x 8-bit, 16-bit/8-bit) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc. 			
I/O port		Total : 88 <ul style="list-style-type: none"> • CMOS input : 2 • CMOS input/output : 78 • N-ch open drain I/O : 8 			
A/D converter		8-bit resolution x 8 channels			
D/A converter		8-bit resolution x 2 channels			
Serial interface		<ul style="list-style-type: none"> • 3-wire serial I/O/2-wire serial I/O/I²C bus mode selection : 1 channel possible • 3-wire serial I/O mode (maximum 32-byte on-chip automatic transmit/receive function) : 1 channel • 3-wire serial I/O/UART mode selection possible : 1 channel 			
Timer		<ul style="list-style-type: none"> • 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 4 channels • Watch timer : 1 channel • Watchdog timer : 1 channel 			
Timer output		5 outputs: (14-bit PWM output enable: 1, 8-bit PWM output enable: 2)			
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (@ 5.0-MHz operation with main system clock) 32.768 kHz (@ 32.768-kHz operation with subsystem clock)			
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (@ 5.0-MHz operation with main system clock)			

- Notes**
1. The capacity of the internal PROM can be changed using the internal memory size switching register (IMS).
 2. The capacity of the internal expansion RAM can be changed using the internal expansion RAM size switching register (IXS).

Item		Part Number	μ PD78076Y	μ PD78078Y	μ PD78P078Y
Vectored interrupt source	Maskable		Internal: 15 External: 7		
	Non-maskable		Internal: 1		
	Software		Internal: 1		
Test input			Internal: 1 External: 1		
Power supply voltage			$V_{DD} = 1.8$ to 5.5 V		
Operating ambient temperature			$T_A = -40$ to $+85^\circ\text{C}$		
Package			<ul style="list-style-type: none"> • 100-pin plastic LQFP^{Note} (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm) • 100-pin plastic QFP (14 x 20 mm, resin thickness 2.7 mm) • 100-pin ceramic WQFN (14 x 20 mm) (μPD78P078Y only) 		

Note Under development

2.9 Mask Options

The mask ROM versions (μ PD78076Y, 78078Y) provide pull-up register mask options which allow users to specify whether to connect a pull-up register to a specific port pin when the user places an order for the device production. Using this mask option when pull-up resistors are required reduces the number of components to add to the device, resulting in board space saving.

The mask options provided in the μ PD78078Y Subseries are shown in Table 2-1.

Table 2-1. Mask Options of Mask ROM Versions

Pin Names	Mask Options
P60 to P63, P90 to P93	Pull-up resistor connection can be specified in 1-bit units.

2.10 Differences with μ PD78054Y Subseries

The μ PD78078Y Subseries is upward-compatible with the μ PD78054Y Subseries. The differences between the two subseries are shown in the table below. The functions and specifications other than those shown in this table are common to these two series.

Table 2-2. Differences between μ PD78078Y Subseries and μ PD78054Y Subseries

Item	μ PD78054Y Subseries	μ PD78078Y Subseries
No. of I/O ports	69	88
8-bit timer/event counter	2 channels	4 channels
External device expansion function	Address bus separate function is not provided.	Address bus separate function is provided. (P80/A0 to P87/A7)
Power supply voltage	V _{DD} = 2.0 to 6.0 V	V _{DD} = 1.8 to 5.5 V
Package	80-pin plastic QFP (14 x 14 mm) 80-pin plastic WQFP (14 x 14 mm)* * : Only for PROM version	100-pin plastic QFP (14 x 20 mm) 100-pin plastic LQFP (14 x 14 mm) 100-pin plastic WQFP (14 x 20 mm)* * : Only for PROM version

[MEMO]

CHAPTER 3 PIN FUNCTION (μ PD78078 SUBSERIES)

3.1 Pin Function List

3.1.1 Normal operating mode pins

(1) Port pins (1/3)

Pin Name	Input/Output	Function		After Reset	Alternate Function	
P00	Input	Port 0. 8-bit input/output port.	Input only	Input	INTP0/TI00	
P01	Input/ output		If used as an input port, an on-chip pull-up resistor can be connected by software.	Input/output mode can be specified bit-wise.	Input	INTP1/TI01
P02						INTP2
P03						INTP3
P04						INTP4
P05						INTP5
P06						INTP6
P07 ^{Note 1}	Input		Input only	Input	XT1	
P10 to P17	Input/ output	Port 1. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as input port, an on-chip pull-up resistor can be connected by software ^{Note 2} .		Input	ANI0 to ANI7	
P20	Input/ output	Port 2. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		Input	SI1	
P21					SO1	
P22					$\overline{\text{SCK}}1$	
P23					STB	
P24					BUSY	
P25					SI0/SB0	
P26					SO0/SB1	
P27					$\overline{\text{SCK}}0$	

- Notes**
1. When the P07/XT1 pin is used as an input port, set the bit 6 (FRC) of the processor clock control register (PCC) to 1 (do not use the feedback resistor internal to the subsystem clock oscillator).
 2. When pins P10/ANI0 to P17/ANI7 are used as an analog input of the A/D converter, set port 1 in the input mode. The on-chip pull-up resistor becomes automatically disabled.

(1) Port pins (2/3)

Pin Name	Input/Output	Function		After Reset	Alternate Function	
P30	Input/ output	Port 3. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		Input	TO0	
P31					TO1	
P32					TO2	
P33					TI1	
P34					TI2	
P35					PCL	
P36					BUZ	
P37					—	
P40 to P47	Input/ output	Port 4. 8-bit input/output port. Input/output mode can be specified in 8-bit units. If used as an input port, an on-chip pull-up resistor can be connected by software. Test input flag (KRIF) is set to 1 by falling edge detection.		Input	AD0 to AD7	
P50 to P57	Input/ output	Port 5. 8-bit input/output port. LEDs can be driven directly. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		Input	A8 to A15	
P61	Input/ output	Port 6. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		Input	N-ch open drain input/output port. On-chip pull-up resistor can be specified by mask option. (Mask ROM version only). LEDs can be driven directly. If used as an input port, an on-chip pull-up resistor can be connected by software.	
P62						—
P63						\overline{RD}
P64						\overline{WR}
P65						WAIT
P66						ASTB
P67						
P70	Input/ output	Port 7. 3-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		Input	SI2/RxD	
P71					SO2/TxD	
P72					$\overline{SCK2/ASCK}$	
P80 to P87	Input/ output	Port 8. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		Input	A0 to A7	

(1) Port pins (3/3)

Pin Name	Input/Output	Function		After Reset	Alternate Function
P90	Input/ output	Port 9. 7-bit input/output port. Input/output mode can be specified bit-wise.	N-ch open-drain input/output port. On-chip pull-up resistor can be specified by mask option. (Mask ROM version only). LEDs can be driven directly. If used as an input port, an on-chip pull-up resistor can be connected by software.	Input	—
P91					
P92					
P93					
P94					
P95					
P96					
P100	Input/ output	Port 10. 4-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		Input	T15/TO5
P101					T16/TO6
P102, P103					—
P120 to P127	Input/ output	Port 12. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		Input	RTP0 to RTP7
P130, P131	Input/ output	Port 13. 2-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		Input	ANO0, ANO1

(2) Non-port pins (1/2)

Pin Name	Input/Output	Function	After Reset	Alternate Function
INTP0	Input	External interrupt request inputs with specifiable valid edges (rising edge, falling edge, both rising and falling edges).	Input	P00/TI00
INTP1				P01/TI01
INTP2				P02
INTP3				P03
INTP4				P04
INTP5				P05
INTP6				P06
SI0	Input	Serial interface serial data input	Input	P25/SB0
SI1				P20
SI2				P70/RxD
SO0	Output	Serial interface serial data output	Input	P26/SB1
SO1				P21
SO2				P71/TxD
SB0	Input/output	Serial interface serial data input/output	Input	P25/SI0
SB1				P26/SO0
$\overline{SCK0}$	Input/output	Serial interface serial clock input/output	Input	P27
$\overline{SCK1}$				P22
$\overline{SCK2}$				P72/ASCK
STB	Output	Serial interface automatic transmit/receive strobe output	Input	P23
BUSY	Input	Serial interface automatic transmit/receive busy input	Input	P24
RxD	Input	Asynchronous serial interface serial data input	Input	P70/SI2
TxD	Output	Asynchronous serial interface serial data output	Input	P71/SO2
ASCK	Input	Asynchronous serial interface serial clock input	Input	P72/ $\overline{SCK2}$
TI00	Input	External count clock input to 16-bit timer (TM0)	Input	P00/INTP0
TI01		Capture trigger signal input to capture register (CR00)		P01/INTP1
TI1		External count clock input to 8-bit timer (TM1)		P33
TI2		External count clock input to 8-bit timer (TM2)		P34
TI5		External count clock input to 8-bit timer (TM5)		P100/TO5
TI6		External count clock input to 8-bit timer (TM6)		P101/TO6
TO0	Output	16-bit timer (TM0) output (also used for 14-bit PWM output)	Input	P30
TO1		8-bit timer (TM1) output		P31
TO2		8-bit timer (TM2) output		P32
TO5		8-bit timer (TM5) output (also used for 8-bit PWM output)		P100/TO5
TO6		8-bit timer (TM6) output (also used for 8-bit PWM output)		P101/TO6
PCL	Output	Clock output (for main system clock and subsystem clock trimming)	Input	P35
BUZ	Output	Buzzer output	Input	P36
RTP0 to RTP7	Output	Real-time output port outputting data in synchronization with trigger	Input	P120 to P127

(2) Non-port pins (2/2)

Pin Name	Input/Output	Function	After Reset	Alternate Function
AD0 to AD7	Input/Output	Low-order address/data bus when expanding external memory	Input	P40 to P47
A0 to A7	Output	Low-order address bus when expanding external memory	Input	P80 to P87
A8 to A15	Output	High-order address bus when expanding external memory	Input	P50 to P57
\overline{RD}	Output	Strobe signal output for read operation from external memory	Input	P64
\overline{WR}		Strobe signal output for write operation to external memory		P65
\overline{WAIT}	Input	Wait insertion when accessing external memory	Input	P66
ASTB	Output	Strobe output externally latching address information output to ports 4, 5 to access external memory	Input	P67
ANI0 to ANI7	Input	A/D converter analog input	Input	P10 to P17
ANO0, ANO1	Output	D/A converter analog output	Input	P130, P131
AV _{REF0}	Input	A/D converter reference voltage input	—	—
AV _{REF1}	Input	D/A converter reference voltage input	—	—
AV _{DD}	—	A/D converter analog power supply. Connect to V _{DD} .	—	—
AV _{SS}	—	A/D converter, D/A converter ground potential. Connect to V _{SS} .	—	—
\overline{RESET}	Input	System reset input	—	—
X1	Input	Crystal connection for main system clock oscillation	—	—
X2	—		—	—
XT1	Input	Crystal connection for subsystem clock oscillation	Input	P07
XT2	—		—	—
V _{DD}	—	Positive power supply	—	—
V _{PP}	—	High-voltage application for program write/verify. Connect directly to V _{SS} in normal operating mode.	—	—
V _{SS}	—	Ground potential	—	—
IC	—	Internally connected. Connect directly to V _{SS} .	—	—

3.1.2 PROM programming mode pins (μ PD78P078 only)

Pin Name	Input/Output	Function
\overline{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{RESET} pin, the PROM programming mode is set.
V _{PP}	Input	High-voltage application for PROM programming mode setting and program write/verify.
A0 to A16	Input	Address bus
D0 to D7	Input/output	Data bus
\overline{CE}	Input	PROM enable input/program pulse input
\overline{OE}	Input	Read strobe input to PROM
\overline{PGM}	Input	Program/program inhibit input in PROM programming mode
V _{DD}	—	Positive power supply
V _{SS}	—	Ground potential

3.2 Description of Pin Functions

3.2.1 P00 to P07 (Port 0)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an external interrupt request input, an external count clock input to the timer, a capture trigger signal input, and crystal connection for subsystem oscillation.

The following operating modes can be specified bit-wise.

(1) Port mode

P00 and P07 function as input-only ports 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). When they are used as input ports, on-chip pull-up resistors can be connected to them by defining the pull-up resistor option register L (PUOL).

(2) Control mode

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

(a) INTPO to 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). INTP0 or INTP1 becomes a 16-bit timer/event counter capture trigger signal input pin with a valid edge input.

(b) TI00

Pin for external count clock input to 16-bit timer/event counter

(c) TI01

Pin for capture trigger signal to capture register (CR00) of 16-bit timer/event counter

(d) XT1

Crystal connect pin for subsystem clock oscillation

3.2.2 P10 to P17 (Port 1)

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

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports.

They can be specified bit-wise as input or output ports with a port mode register 1 (PM1). If used as input ports, on-chip pull-up resistors can be connected to them by defining the pull-up resistor option register L (PUOL).

(2) Control mode

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

3.2.3 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). When they are used as input ports, on-chip pull-up resistors can be connected to them by defining the pull-up resistor option register L (PUOL).

(2) Control mode

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

(a) SI0, SI1, SO0, SO1

Serial interface serial data input/output pins

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

Serial interface serial clock input/output pins

(c) SB0 and SB1

NEC standard serial bus interface input/output pins

(d) BUSY

Serial interface automatic transmit/receive busy input pins

(e) STB

Serial interface automatic transmit/receive strobe output pins

Caution 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 17-4. Serial Operation Mode Register 0 Format and Figure 19-3. Serial Operation Mode Register 1 Format.

3.2.4 P30 to P37 (Port 3)

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

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 3 (PM3). When they are used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register L (PUOL).

(2) Control mode

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

(a) T11 and T12

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

(b) T00 to T02

Timer output pins.

(c) PCL

Clock output pin.

(d) BUZ

Buzzer output pin.

3.2.5 P40 to P47 (Port 4)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an address/data bus. The test input flag (KRIF) can be set to 1 by detecting a falling edge.

The following operating mode can be specified in 8-bit units.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified in 8-bit units for input or output ports by using the memory expansion mode register (MM). When they are used as input ports, on-chip pull-up resistors can be connected to them by defining the pull-up resistor option register L (PUOL).

(2) Control mode

These ports function as low-order address/data bus pins (AD0 to AD7) in external memory expansion mode. When pins are used as an address/data bus, the on-chip pull-up resistor is automatically disabled.

3.2.6 P50 to P57 (Port 5)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an address bus. Port 5 can drive LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input/output ports with port mode register 5 (PM5). When they are used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register L (PUOL).

(2) Control mode

These ports function as high-order address bus pins (A8 to A15) in external memory expansion mode. When pins are used as an address bus, the on-chip pull-up resistor is automatically disabled.

3.2.7 P60 to P67 (Port 6)

These are 8-bit input/output ports. Besides serving as input/output ports, they are used for control in external memory expansion mode. P60 to P63 can drive LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 6 (PM6).

P60 to P63 are N-ch open-drain outputs. Mask ROM version can contain pull-up resistors with the mask option.

When P64 to P67 are used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register L (PUOL).

(2) Control mode

These ports function as control signal output pins (\overline{RD} , \overline{WR} , \overline{WAIT} , ASTB) in external memory expansion mode. When a pin is used as a control signal output, the on-chip pull-up resistor is automatically disabled.

Caution When external wait is not used in external memory expansion mode, P66 can be used as an input/output port.

3.2.8 P70 to P72 (Port 7)

This is a 3-bit input/output port. In addition to its use as an input/output port, it also has serial interface data input/output and clock input/output functions.

The following operating modes can be specified bit-wise.

(1) Port mode

Port 7 functions as a 3-bit input/output port. Bit-wise specification as an input port or output port is possible by means of port mode register 7 (PM7). When used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register L (PUOL).

(2) Control mode

Port 7 functions as serial interface data input/output and clock input/output.

(a) SI2, SO2

Serial interface serial data input/output pins

(b) $\overline{\text{SCK2}}$

Serial interface serial clock input/output pin.

(c) RxD, TxD

Asynchronous serial interface serial data input/output pins.

(d) ASCK

Asynchronous serial interface serial clock input/output pin.

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 the operation mode setting list in Table 20-2. Serial Interface Channel 2.

3.2.9 P80 to P87 (Port 8)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an address bus.

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 8 (PM8). When they are used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register H (PUOH).

(2) Control mode

These ports function as lower address bus pins (A0 to A7) in external memory expansion mode. When a pin is used as an address bus, the on-chip pull-up resistor is automatically not used.

3.2.10 P90 to P96 (Port 9)

These are 7-bit input/output ports.

P90 to P93 can drive LEDs directly.

They can be specified bit-wise as input or output ports with port mode register 9 (PM9).

P90 to P93 are N-ch open-drain pins. Mask ROM version product can contain pull-up resistors with the mask option. When P94 to P96 are used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register H (PUOH).

3.2.11 P100 to P103 (Port 10)

These are 4-bit input/output ports. Besides serving as input/output ports, they function as a timer input/output. The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 4-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 10 (PM10). When they are used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register H (PUOH).

(2) Control mode

These ports function as timer input/output.

(a) TI5 and TI6

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

(b) TO5 to TO6

Pins for timer output.

3.2.12 P120 to P127 (Port 12)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as a real-time output port. 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 12 (PM12). When they are used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register H (PUOH).

(2) Control mode

These ports function as real-time output ports (RTP0 to RTP7) outputting data in synchronization with a trigger.

3.2.13 P130 and P131 (Port 13)

These are 2-bit input/output ports. Besides serving as input/output ports, they are used for D/A converter analog output.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 2-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 13 (PM13). When they are used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register H (PUOH).

(2) Control mode

These ports allow D/A converter analog output (ANO0 and ANO1).

Caution When only either one of the D/A converter channels is used with $AV_{REF1} < V_{DD}$, the other pins that are not used as analog outputs must be set as follows:

- Set PM13x bit of the port mode register 13 (PM13) to 1 (input mode) and connect the pin to V_{SS} .
- Set PM13x bit of the port mode register 13 (PM13) to 0 (output mode) and the output latch to 0, to output low level from the pin.

3.2.14 AV_{REF0}

A/D converter reference voltage input pin.

When A/D converter is not used, connect this pin to V_{SS} .

3.2.15 AV_{REF1}

D/A converter reference voltage input pin.

When D/A converter is not used, connect this pin to V_{DD} .

3.2.16 AV_{DD}

Analog power supply pin of A/D converter. Always use the same voltage as that of the V_{DD} pin even when A/D converter is not used.

3.2.17 AV_{SS}

This is a ground voltage pin of A/D converter and D/A converter. Always use the same voltage as that of the V_{SS} pin even when A/D converter or D/A converter is not used.

3.2.18 \overline{RESET}

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

3.2.19 X1 and X2

Crystal resonator connect pins for main system clock oscillation. For external clock supply, input it to X1 and its inverted signal to X2.

3.2.20 XT1 and XT2

Crystal resonator connect pins for subsystem clock oscillation.

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

3.2.21 V_{DD}

Positive power supply pin

3.2.22 V_{SS}

Ground potential pin

3.2.23 V_{PP} (μ PD78P078 only)

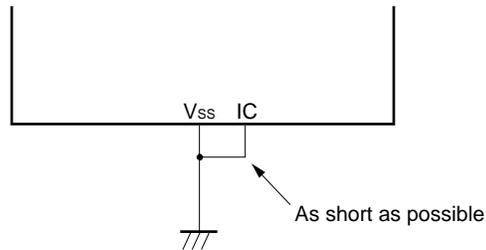
High-voltage apply pin for PROM programming mode setting and program write/verify. Connect directly to V_{SS} in normal operating mode.

3.2.24 IC (Mask ROM version only)

The IC (Internally Connected) pin is provided to set the test mode to check the μ PD78078 at delivery. Connect it directly to the V_{SS} with the shortest possible wire in the normal operating mode.

When a voltage difference is produced between the IC pin and V_{SS} 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 pins to V_{SS} pins directly.**



3.3 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 3-1 for the configuration of the input/output circuit of each type.

Table 3-1. Pin Input/Output Circuit Types (1/2)

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
P00/INTP0/TI00	2	Input	Connect to V_{SS} .
P01/INTP1/TI01	8-A	Input/Output	Connect independently via a resistor to V_{SS} .
P02/INTP2			
P03/INTP3			
P04/INTP4			
P05/INTP5			
P06/INTP6			
P07/XT1	16	Input	Connect to V_{DD} .
P10/ANI0 to P17/ANI7	11	Input/Output	Connect independently via a resistor to V_{DD} or V_{SS} .
P20/SI1	8-A		
P21/SO1	5-A		
P22/ $\overline{SCK1}$	8-A		
P23/STB	5-A		
P24/BUSY	8-A		
P25/SI0/SB0	10-A		
P26/SO0/SB1			
P27/ $\overline{SCK0}$			
P30/TO0	5-A		
P31/TO1			
P32/TO2			
P33/TI1	8-A		
P34/TI2			
P35/PCL	5-A		
P36/BUZ			
P37			
P40/AD0 to P47/AD7	5-E	Input/Output	Connect independently via a resistor to V_{DD} .

Table 3-1. Pin Input/Output Circuit Types (2/2)

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
P50/A8 to P57/A15	5-A	Input/output	Connect independently via a resistor to V_{DD} or V_{SS} .
P60 to P63 (Mask ROM version)	13-B	Input/output	Connect independently via a resistor to V_{DD} .
P60 to P63 (μ PD78P078)	13-D		
P64/ \overline{RD}	5-A	Input/output	Connect independently via a resistor to V_{DD} or V_{SS} .
P65/ \overline{WR}			
P66/ \overline{WAIT}			
P67/ASTB			
P70/SI2/RxD	8-A	Input/output	Connect independently via a resistor to V_{DD} or V_{SS} .
P71/SO2/TxD	5-A		
P72/ $\overline{SCK2}$ /ASCK	8-A		
P80/A0 to P87/A7	5-A		
P90 to P93 (Mask ROM version)	13-B		
P90 to P93 (μ PD78P078)	13-D		
P94 to P96	5-A	Input/output	Connect independently via a resistor to V_{DD} or V_{SS} .
P100/TI5/TO5	8-A		
P101/TI6/TO6	5-A		
P102, P103			
P120/RTP0 to P127/RTP7	5-A		
P130/ANO0, P131/ANO1	12-A	Input/output	Connect independently via a resistor to V_{SS} .
\overline{RESET}	2	Input	—
XT2	16	—	Open
AV_{REF0}	—		Connect to V_{SS} .
AV_{REF1}			Connect to V_{DD} .
AV_{DD}			Connect to V_{SS} .
AV_{SS}			
IC (Mask ROM version)			Connect directly to V_{SS} .
V_{PP} (μ PD78P078)			

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

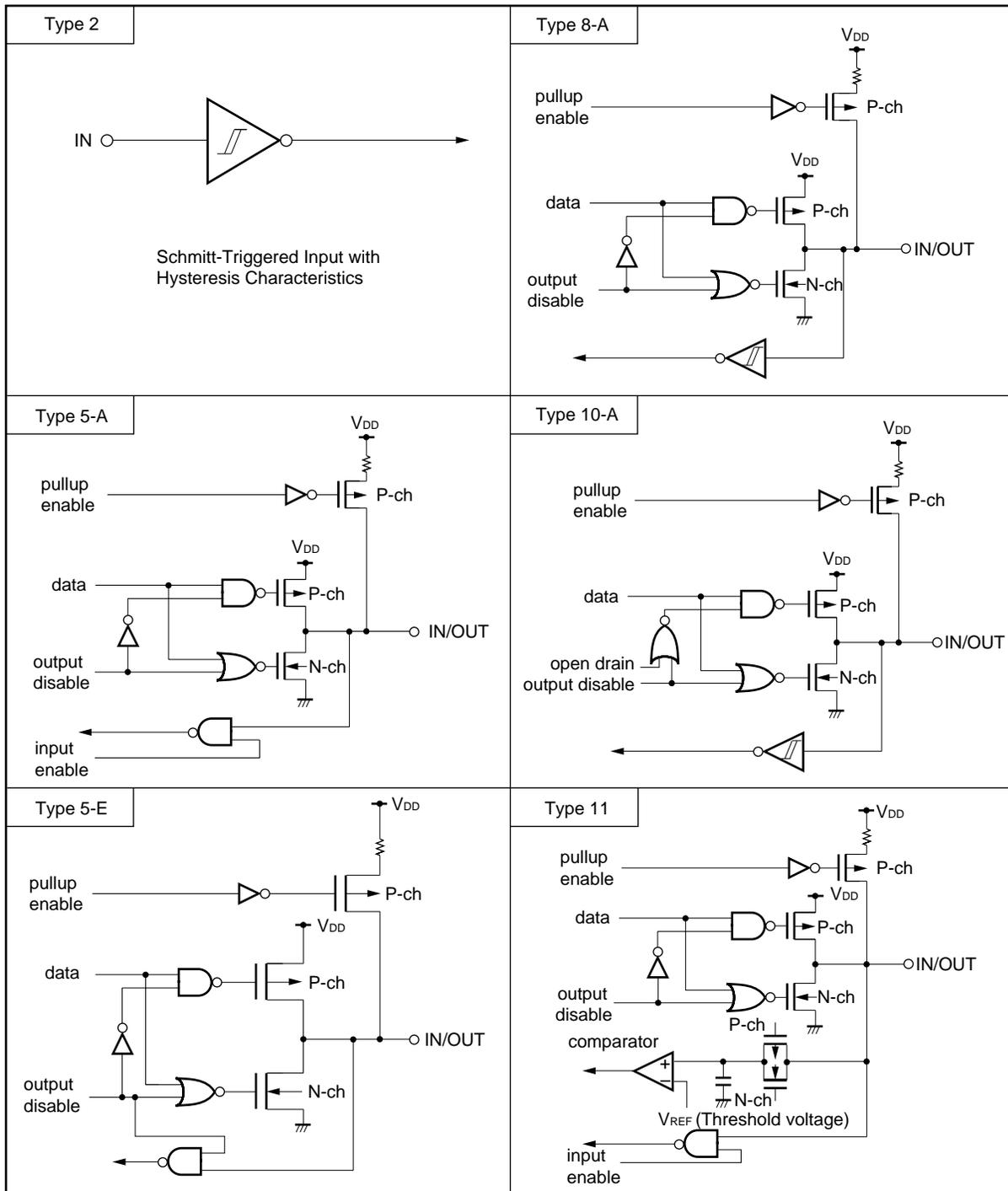
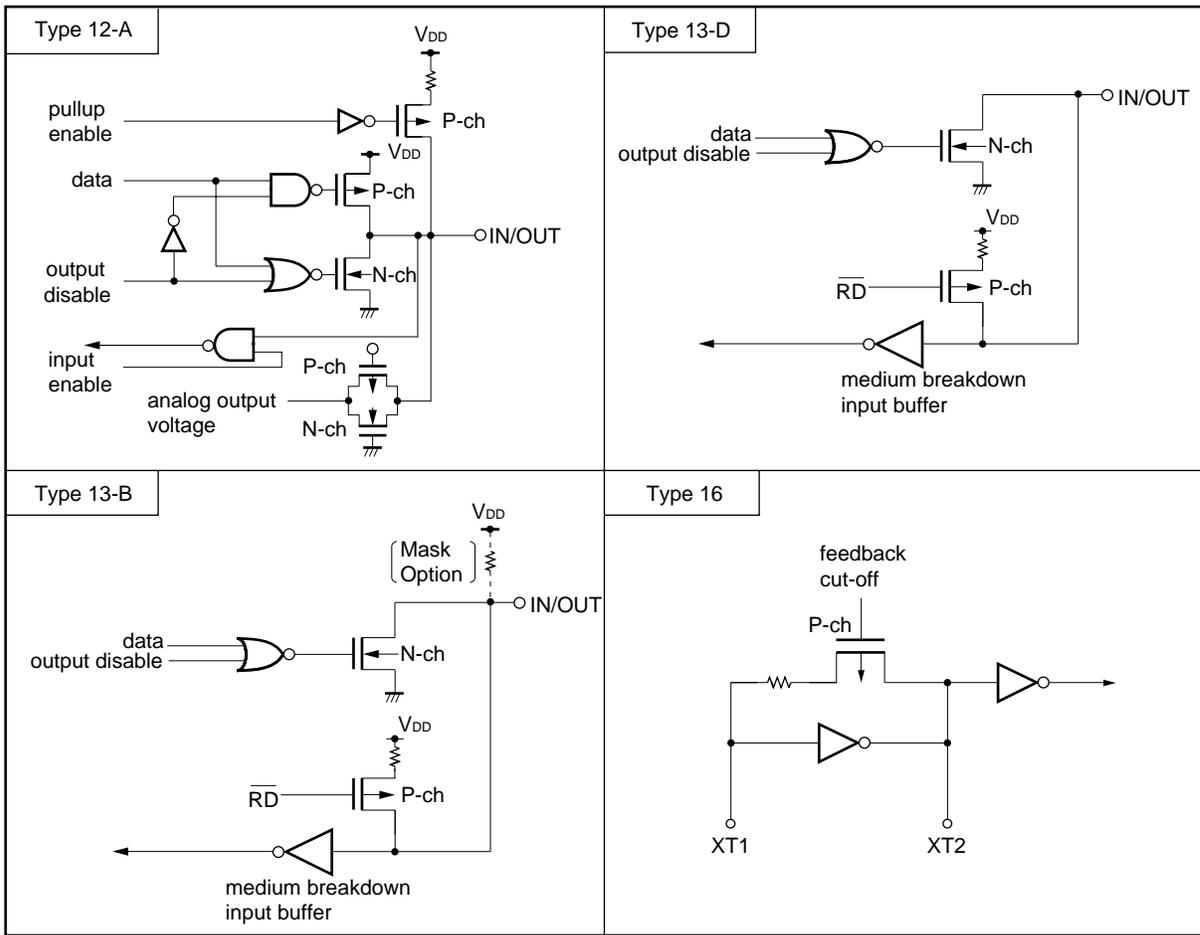


Figure 3-1. List of Pin Input/Output Circuits (2/2)



[MEMO]

CHAPTER 4 PIN FUNCTION (μ PD78078Y SUBSERIES)

4.1 Pin Function List

4.1.1 Normal operating mode pins

(1) Port pins (1/3)

Pin Name	Input/Output	Function		After Reset	Alternate Function
P00	Input	Port 0. 8-bit input/output port.	Input only	Input	INTP0/TI00
P01	Input/ output		Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.	Input	INTP1/TI01
P02					INTP2
P03					INTP3
P04					INTP4
P05					INTP5
P06					INTP6
P07 ^{Note 1}	Input		Input only	Input	XT1
P10 to P17	Input/ output	Port 1. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as input port, an on-chip pull-up resistor can be connected by software ^{Note 2} .		Input	ANI0 to ANI7
P20	Input/ output	Port 2. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.	Input	SI1	
P21				SO1	
P22				SCK1	
P23				STB	
P24				BUSY	
P25				SI0/SB0/SDA0	
P26				SO0/SB1/SDA1	
P27				SCK0/SCL	

- Notes**
1. When the P07/XT1 pin is used as an input port, set the bit 6 (FRC) of the processor clock control register (PCC) to 1 (do not use the feedback resistor internal to the subsystem clock oscillator).
 2. When pins P10/ANI0 to P17/ANI7 are used as an analog input of the A/D converter, set port 1 in the input mode. The on-chip pull-up resistor becomes automatically disabled.

(1) Port pins (2/3)

Pin Name	Input/Output	Function		After Reset	Alternate Function	
P30	Input/ output	Port 3. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		Input	TO0	
P31					TO1	
P32					TO2	
P33					TI1	
P34					TI2	
P35					PCL	
P36					BUZ	
P37					—	
P40 to P47	Input/ output	Port 4. 8-bit input/output port. Input/output mode can be specified in 8-bit units. If used as an input port, an on-chip pull-up resistor can be connected by software. Test input flag (KRIF) is set to 1 by falling edge detection.		Input	AD0 to AD7	
P50 to P57	Input/ output	Port 5. 8-bit input/output port. LEDs can be driven directly. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		Input	A8 to A15	
P61	Input/ output	Port 6. 8-bit input/output port. Input/output mode can be specified bit-wise.		Input	N-ch open drain input/output port. On-chip pull-up resistor can be specified by mask option. (Mask ROM version only). LEDs can be driven directly. If used as an input port, an on-chip pull-up resistor can be connected by software.	
P62						—
P63						RD
P64						WR
P65						WAIT
P66						ASTB
P67						—
P70	Input/ output	Port 7. 3-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		Input	SI2/RxD	
P71					SO2/TxD	
P72					SCK2/ASCK	
P80 to P87	Input/ output	Port 8. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		Input	A0 to A7	

(1) Port pins (3/3)

Pin Name	Input/Output	Function		After Reset	Alternate Function
P90	Input/ output	Port 9. 7-bit input/output port. Input/output mode can be specified bit-wise.	N-ch open-drain input/output port. On-chip pull-up resistor can be specified by mask option. (Mask ROM version only). LEDs can be driven directly. If used as an input port, an on-chip pull-up resistor can be connected by software.	Input	—
P91					
P92					
P93					
P94					
P95					
P96					
P100	Input/ output	Port 10. 4-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		Input	T15/TO5
P101					T16/TO6
P102, P103					—
P120 to P127	Input/ output	Port 12. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		Input	RTP0 to RTP7
P130 to P131	Input/ output	Port 13. 2-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		Input	ANO0 to ANO1

(2) Non-port pins (1/2)

Pin Name	Input/Output	Function	After Reset	Alternate Function
INTP0	Input	External interrupt request inputs with specifiable valid edges (rising edge, falling edge, both rising and falling edges).	Input	P00/TI00
INTP1				P01/TI01
INTP2				P02
INTP3				P03
INTP4				P04
INTP5				P05
INTP6				P06
SI0	Input	Serial interface serial data input	Input	P25/SB0/SDA0
SI1				P20
SI2				P70/RxD
SO0	Output	Serial interface serial data output	Input	P26/SB1/SDA1
SO1				P21
SO2				P71/TxD
SB0	Input/ output	Serial interface serial data input/output	Input	P25/SI0/SDA0
SB1				P26/SO0/SDA1
SDA0				P25/SI0/SB0
SDA1				P26/SO0/SB1
$\overline{SCK0}$	Input/ output	Serial interface serial clock input/output	Input	P27/SCL
$\overline{SCK1}$				P22
$\overline{SCK2}$				P72/ASCK
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
RxD	Input	Asynchronous serial interface serial data input	Input	P70/SI2
TxD	Output	Asynchronous serial interface serial data output	Input	P71/SO2
ASCK	Input	Asynchronous serial interface serial clock input	Input	P72/ $\overline{SCK2}$
TI00	Input	External count clock input to 16-bit timer (TM0)	Input	P00/INTP0
TI01		Capture trigger signal input to capture register (CR00)		P01/INTP1
TI1		External count clock input to 8-bit timer (TM1)		P33
TI2		External count clock input to 8-bit timer (TM2)		P34
TI5		External count clock input to 8-bit timer (TM5)		P100/TO5
TI6		External count clock input to 8-bit timer (TM6)		P101/TO6
TO0	Output	16-bit timer (TM0) output (also used for 14-bit PWM output)	Input	P30
TO1		8-bit timer (TM1) output		P31
TO2		8-bit timer (TM2) output		P32
TO5		8-bit timer (TM5) output (also used for 8-bit PWM output)		P100/TO5
TO6		8-bit timer (TM6) output (also used for 8-bit PWM output)		P101/TO6
PCL	Output	Clock output (for main system clock and subsystem clock trimming)	Input	P35
BUZ	Output	Buzzer output	Input	P36
RTP0 to RTP7	Output	Real-time output port outputting data in synchronization with trigger	Input	P120 to P127

(2) Non-port pins (2/2)

Pin Name	Input/Output	Function	After Reset	Alternate Function
AD0 to AD7	Input/Output	Low-order address/data bus when expanding external memory	Input	P40 to P47
A0 to A7	Output	Low-order address bus when expanding external memory	Input	P80 to P87
A8 to A15	Output	High-order address bus when expanding external memory	Input	P50 to P57
\overline{RD}	Output	Strobe signal output for read operation from external memory	Input	P64
\overline{WR}		Strobe signal output for write operation to external memory		P65
\overline{WAIT}	Input	Wait insertion when accessing external memory	Input	P66
ASTB	Output	Strobe output externally latching address information output to ports 4, 5 to access external memory	Input	P67
ANI0 to ANI7	Input	A/D converter analog input	Input	P10 to P17
ANO0, ANO1	Output	D/A converter analog output	Input	P130, P131
AV _{REF0}	Input	A/D converter reference voltage input	—	—
AV _{REF1}	Input	D/A converter reference voltage input	—	—
AV _{DD}	—	A/D converter analog power supply. Connect to V _{DD} .	—	—
AV _{SS}	—	A/D converter, D/A converter ground potential. Connect to V _{SS} .	—	—
\overline{RESET}	Input	System reset input	—	—
X1	Input	Crystal connection for main system clock oscillation	—	—
X2	—		—	—
XT1	Input	Crystal connection for subsystem clock oscillation	Input	P07
XT2	—		—	—
V _{DD}	—	Positive power supply	—	—
V _{PP}	—	High-voltage application for program write/verify. Connect directly to V _{SS} in normal operating mode.	—	—
V _{SS}	—	Ground potential	—	—
IC	—	Internally connected. Connect directly to V _{SS} .	—	—

4.1.2 PROM programming mode pins (μ PD78P078Y only)

Pin Name	Input/Output	Function
\overline{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{RESET} pin, the PROM programming mode is set.
V _{PP}	Input	High-voltage application for PROM programming mode setting and program write/verify.
A0 to A16	Input	Address bus
D0 to D7	Input/output	Data bus
\overline{CE}	Input	PROM enable input/program pulse input
\overline{OE}	Input	Read strobe input to PROM
\overline{PGM}	Input	Program/program inhibit input in PROM programming mode
V _{DD}	—	Positive power supply
V _{SS}	—	Ground potential

4.2 Description of Pin Functions

4.2.1 P00 to P07 (Port 0)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an external interrupt request input, an external count clock input to the timer, a capture trigger signal input, and crystal connection for subsystem oscillation.

The following operating modes can be specified bit-wise.

(1) Port mode

P00 and P07 function as input-only ports 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). When they are used as input ports, on-chip pull-up resistors can be connected to them by defining the pull-up resistor option register L (PUOL).

(2) Control mode

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

(a) INTPO to INTP6

INTPO to INTP6 are external interrupt request input pins which can specify valid edges (rising edge, falling edge, and both rising and falling edges). INTPO or INTP1 becomes a 16-bit timer/event counter capture trigger signal input pin with a valid edge input.

(b) TI00

Pin for external count clock input to 16-bit timer/event counter

(c) TI01

Pin for capture trigger signal to capture register (CR00) of 16-bit timer/event counter

(d) XT1

Crystal connect pin for subsystem clock oscillation

4.2.2 P10 to P17 (Port 1)

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

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports.

They can be specified bit-wise as input or output ports with a port mode register 1 (PM1). If used as input ports, on-chip pull-up resistors can be connected to these ports by defining the pull-up resistor option register L (PUOL).

(2) Control mode

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

4.2.3 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). When they are used as input ports, on-chip pull-up resistors can be connected to them by defining the pull-up resistor option register L (PUOL).

(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, SB0, SB1, SDA0, SDA1

Serial interface serial data input/output pins

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

Serial interface serial clock input/output pins

(c) BUSY

Serial interface automatic transmit/receive busy input pins

(d) 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 18-4. Serial Operation Mode Register 0 Format and Figure 19-3. Serial Operation Mode Register 1 Format.

4.2.4 P30 to P37 (Port 3)

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

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 3 (PM3). When they are used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register L (PUOL).

(2) Control mode

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

(a) T11 and T12

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

(b) T00 to T02

Timer output pins.

(c) PCL

Clock output pin.

(d) BUZ

Buzzer output pin.

4.2.5 P40 to P47 (Port 4)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an address/data bus. The test input flag (KRIF) can be set to 1 by detecting a falling edge.

The following operating mode can be specified in 8-bit units.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified in 8-bit units for input or output ports by using the memory expansion mode register (MM). When they are used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register L (PUOL).

(2) Control mode

These ports function as low-order address/data bus pins (AD0 to AD7) in external memory expansion mode. When pins are used as an address/data bus, the on-chip pull-up resistor is automatically disabled.

4.2.6 P50 to P57 (Port 5)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an address bus. Port 5 can drive LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input/output ports with port mode register 5 (PM5). When they are used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register L (PUOL).

(2) Control mode

These ports function as high-order address bus pins (A8 to A15) in external memory expansion mode. When pins are used as an address bus, the on-chip pull-up resistor is automatically disabled.

4.2.7 P60 to P67 (Port 6)

These are 8-bit input/output ports. Besides serving as input/output ports, they are used for control in external memory expansion mode. P60 to P63 can drive LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 6 (PM6).

P60 to P63 are N-ch open-drain outputs. Mask ROM version can contain pull-up resistors with the mask option.

When P64 to P67 are used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register L (PUOL).

(2) Control mode

These ports function as control signal output pins (\overline{RD} , \overline{WR} , \overline{WAIT} , ASTB) in external memory expansion mode. When a pin is used as a control signal output, the on-chip pull-up resistor is automatically disabled.

Caution When external wait is not used in external memory expansion mode, P66 can be used as an input/output port.

4.2.8 P70 to P72 (Port 7)

This is a 3-bit input/output port. In addition to its use as an input/output port, it also has serial interface data input/output and clock input/output functions.

The following operating modes can be specified bit-wise.

(1) Port mode

Port 7 functions as a 3-bit input/output port. Bit-wise specification as an input port or output port is possible by means of port mode register 7 (PM7). When used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register L (PUOL).

(2) Control mode

Port 7 functions as serial interface data input/output and clock input/output.

(a) SI2, SO2

Serial interface serial data input/output pins

(b) $\overline{\text{SCK2}}$

Serial interface serial clock input/output pin.

(c) RxD, TxD

Asynchronous serial interface serial data input/output pins.

(d) ASCK

Asynchronous serial interface serial clock input/output pin.

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 the operation mode setting list in Table 20-2. Serial Interface Channel 2.

4.2.9 P80 to P87 (Port 8)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an address bus.

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 8 (PM8). When they are used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register H (PUOH).

(2) Control mode

These ports function as lower address bus pins (A0 to A7) in external memory expansion mode. When a pin is used as an address bus, the on-chip pull-up resistor is automatically not used.

4.2.10 P90 to P96 (Port 9)

These are 7-bit input/output ports.

P90 to P93 can drive LEDs directly.

They can be specified bit-wise as input or output ports with port mode register 9 (PM9).

P90 to P93 are N-ch open-drain pins. Mask ROM version product can contain pull-up resistors with the mask option. When P94 to P96 are used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register H (PUOH).

4.2.11 P100 to P103 (Port 10)

These are 4-bit input/output ports. Besides serving as input/output ports, they function as a timer input/output. The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 4-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 10 (PM10). When they are used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register H (PUOH).

(2) Control mode

These ports function as timer input/output.

(a) TI5 and TI6

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

(b) TO5 to TO6

Pins for timer output.

4.2.12 P120 to P127 (Port 12)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as a real-time output port. 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 12 (PM12). When they are used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register H (PUOH).

(2) Control mode

These ports function as real-time output ports (RTP0 to RTP7) outputting data in synchronization with a trigger.

4.2.13 P130 and P131 (Port 13)

These are 2-bit input/output ports. Besides serving as input/output ports, they are used for D/A converter analog output.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 2-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 13 (PM13). When they are used as input ports, on-chip pull-up resistors can be connected by defining the pull-up resistor option register H (PUOH).

(2) Control mode

These ports allow D/A converter analog output (ANO0 and ANO1).

Caution When only either one of the D/A converter channels is used with $AV_{REF1} < V_{DD}$, the other pins that are not used as analog outputs must be set as follows:

- Set PM13x bit of the port mode register 13 (PM13) to 1 (input mode) and connect the pin to V_{SS} .
- Set PM13x bit of the port mode register 13 (PM13) to 0 (output mode) and the output latch to 0, to output low level from the pin.

4.2.14 AV_{REF0}

A/D converter reference voltage input pin.

When A/D converter is not used, connect this pin to V_{SS} .

4.2.15 AV_{REF1}

D/A converter reference voltage input pin.

When D/A converter is not used, connect this pin to V_{DD} .

4.2.16 AV_{DD}

Analog power supply pin of A/D converter. Always use the same voltage as that of the V_{DD} pin even when A/D converter is not used.

4.2.17 AV_{SS}

This is a ground voltage pin of A/D converter and D/A converter. Always use the same voltage as that of the V_{SS} pin even when A/D converter or D/A converter is not used.

4.2.18 \overline{RESET}

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

4.2.19 X1 and X2

Crystal resonator connect pins for main system clock oscillation. For external clock supply, input it to X1 and its inverted signal to X2.

4.2.20 XT1 and XT2

Crystal resonator connect pins for subsystem clock oscillation.

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

4.2.21 V_{DD}

Positive power supply pin

4.2.22 V_{SS}

Ground potential pin

4.2.23 V_{PP} (μ PD78P078Y only)

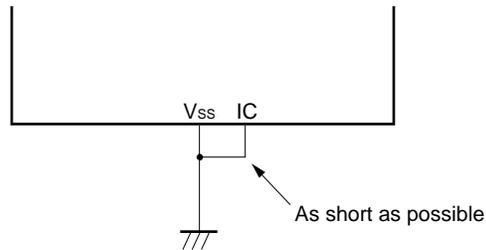
High-voltage apply pin for PROM programming mode setting and program write/verify. Connect directly to V_{SS} in normal operating mode.

4.2.24 IC (Mask ROM version only)

The IC (Internally Connected) pin is provided to set the test mode to check the μ PD78078Y at delivery. Connect it directly to the V_{SS} with the shortest possible wire in the normal operating mode.

When a voltage difference is produced between the IC pin and V_{SS} 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 pins to V_{SS} pins directly.**



4.3 Input/output Circuits and Recommended Connection of Unused Pins

Table 4-1 shows the input/output circuit types of pins and the recommended conditions for unused pins. Refer to **Figure 4-1** for the configuration of the input/output circuit of each type.

Table 4-1. Pin Input/Output Circuit Types (1/2)

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
P00/INTP0/TI00	2	Input	Connect to V _{SS} .
P01/INTP1/TI01	8-A	Input/Output	Connect independently via a resistor to V _{SS} .
P02/INTP2			
P03/INTP3			
P04/INTP4			
P05/INTP5			
P06/INTP6			
P07/XT1	16	Input	Connect to V _{DD} .
P10/ANI0 to P17/ANI7	11	Input/Output	Connect independently via a resistor to V _{DD} or V _{SS} .
P20/SI1	8-A		
P21/SO1	5-A		
P22/ $\overline{\text{SCK}}1$	8-A		
P23/STB	5-A		
P24/BUSY	8-A		
P25/SI0/SB0/SDA0	10-A		
P26/SO0/SB1/SDA1			
P27/ $\overline{\text{SCK}}0$ /SCL			
P30/TO0	5-A		
P31/TO1			
P32/TO2			
P33/TI1	8-A		
P34/TI2			
P35/PCL	5-A		
P36/BUZ			
P37			
P40/AD0 to P47/AD7		5-E	Input/Output

Table 4-1. Pin Input/Output Circuit Types (2/2)

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
P50/A8 to P57/A15	5-A	Input/output	Connect independently via a resistor to V_{DD} or V_{SS} .
P60 to P63 (Mask ROM version)	13-B	Input/output	Connect independently via a resistor to V_{DD} .
P60 to P63 (μ PD78P078Y)	13-D		
P64/ \overline{RD}	5-A	Input/output	Connect independently via a resistor to V_{DD} or V_{SS} .
P65/ \overline{WR}			
P66/ \overline{WAIT}			
P67/ASTB			
P70/SI2/RxD	8-A	Input/output	Connect independently via a resistor to V_{DD} or V_{SS} .
P71/SO2/TxD	5-A		
P72/ $\overline{SCK2}$ /ASCK	8-A		
P80/A0 to P87/A7	5-A		
P90 to P93 (Mask ROM version)	13-B		
P90 to P93 (μ PD78P078Y)	13-D		
P94 to P96	5-A	Input/output	Connect independently via a resistor to V_{DD} or V_{SS} .
P100/TI5/TO5	8-A		
P101/TI6/TO6	5-A		
P102, P103			
P120/RTP0 to P127/RTP7	5-A		
P130/ANO0, P131/ANO1	12-A	Input/output	Connect independently via a resistor to V_{SS} .
\overline{RESET}	2	Input	—
XT2	16	—	Open
AV_{REF0}	—	—	Connect to V_{SS} .
AV_{REF1}			Connect to V_{DD} .
AV_{DD}			Connect to V_{SS} .
AV_{SS}			
IC (Mask ROM version)			Connect directly to V_{SS} .
V_{PP} (μ PD78P078Y)			

Figure 4-1. List of Pin Input/Output Circuits (1/2)

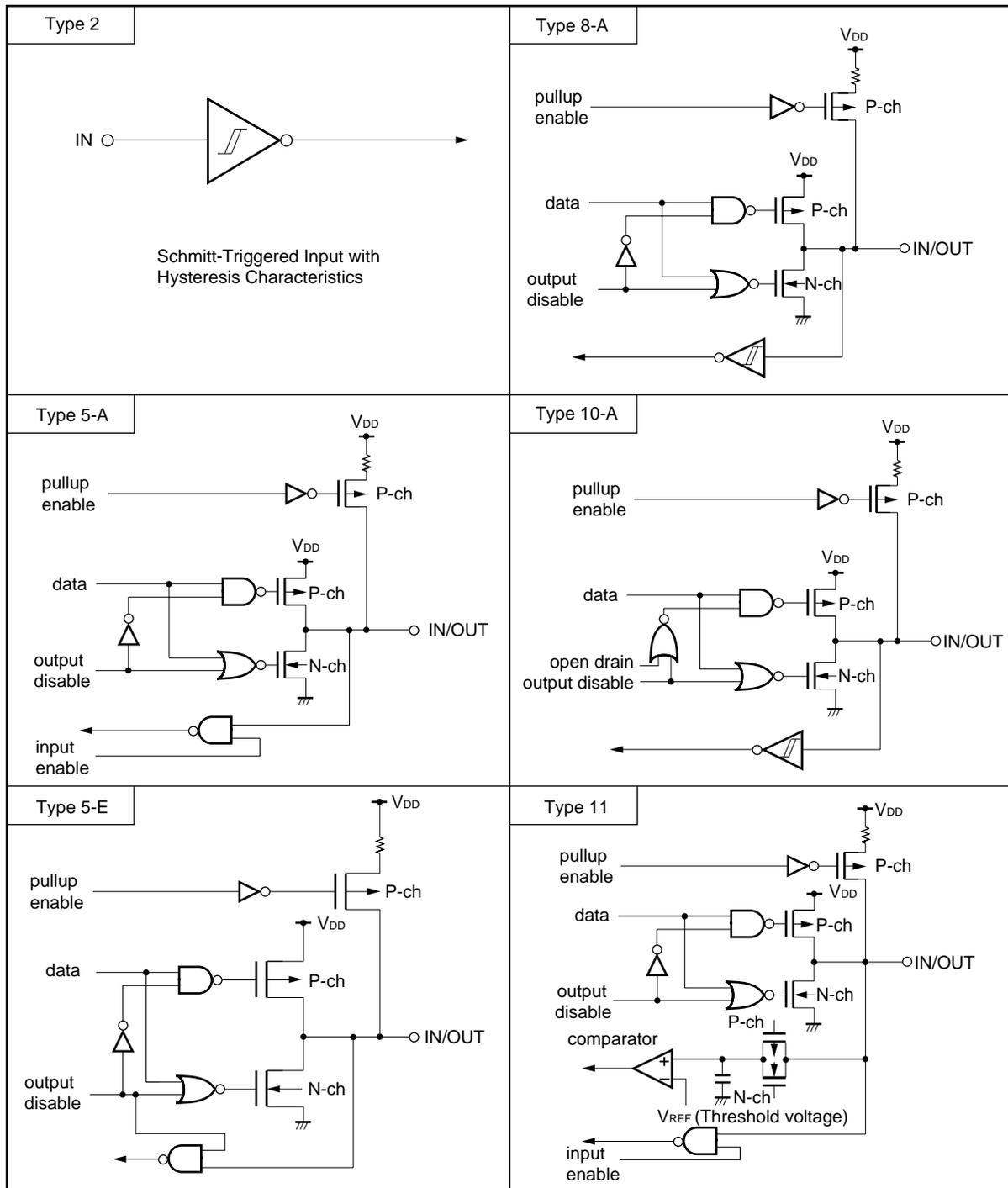
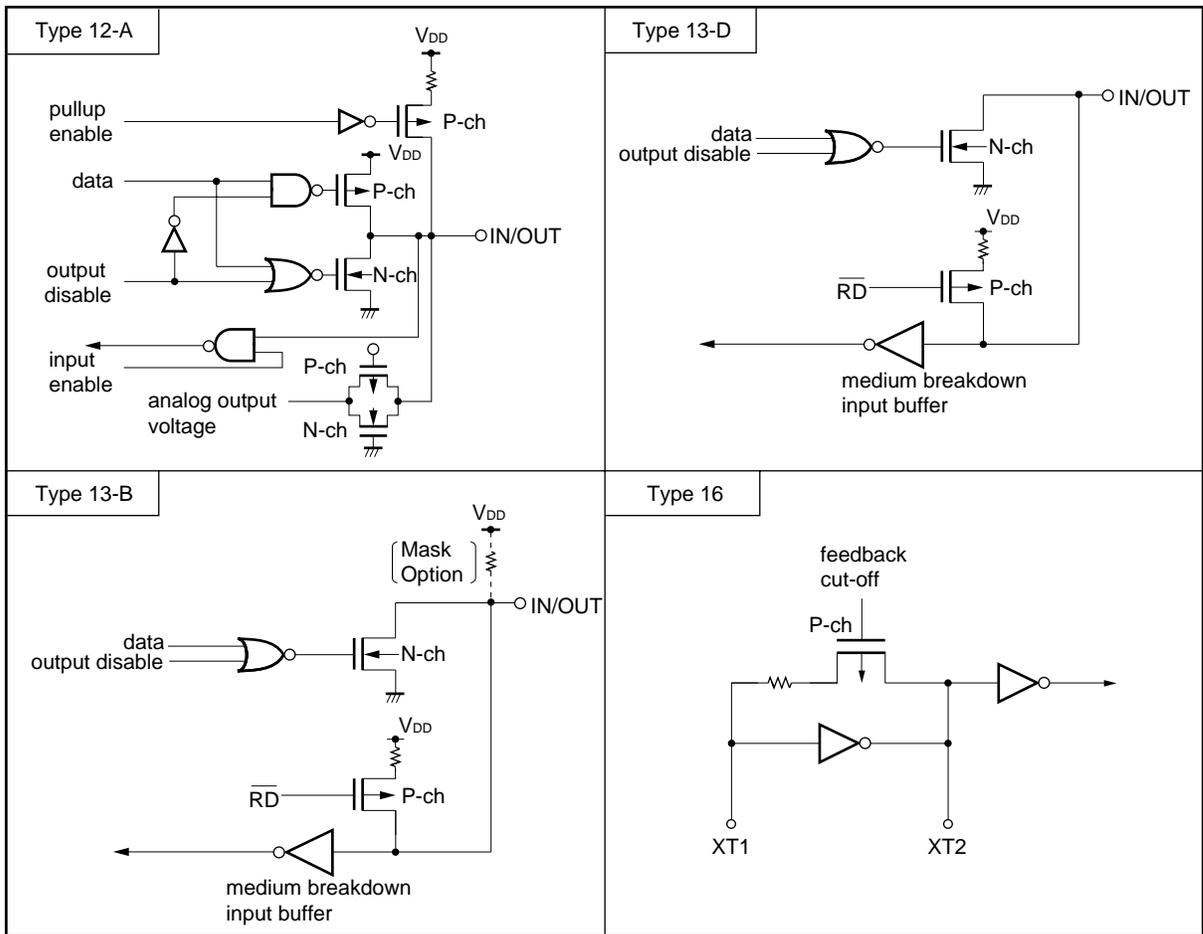


Figure 4-1. List of Pin Input/Output Circuits (2/2)



[MEMO]

CHAPTER 5 CPU ARCHITECTURE

5.1 Memory Spaces

The μ PD78078 and 78078Y Subseries allow access to a memory space of 64 Kbytes. Figures 5-1 to 5-3 shows memory maps.

Figure 5-1. Memory Map (μ PD78076, 78076Y)

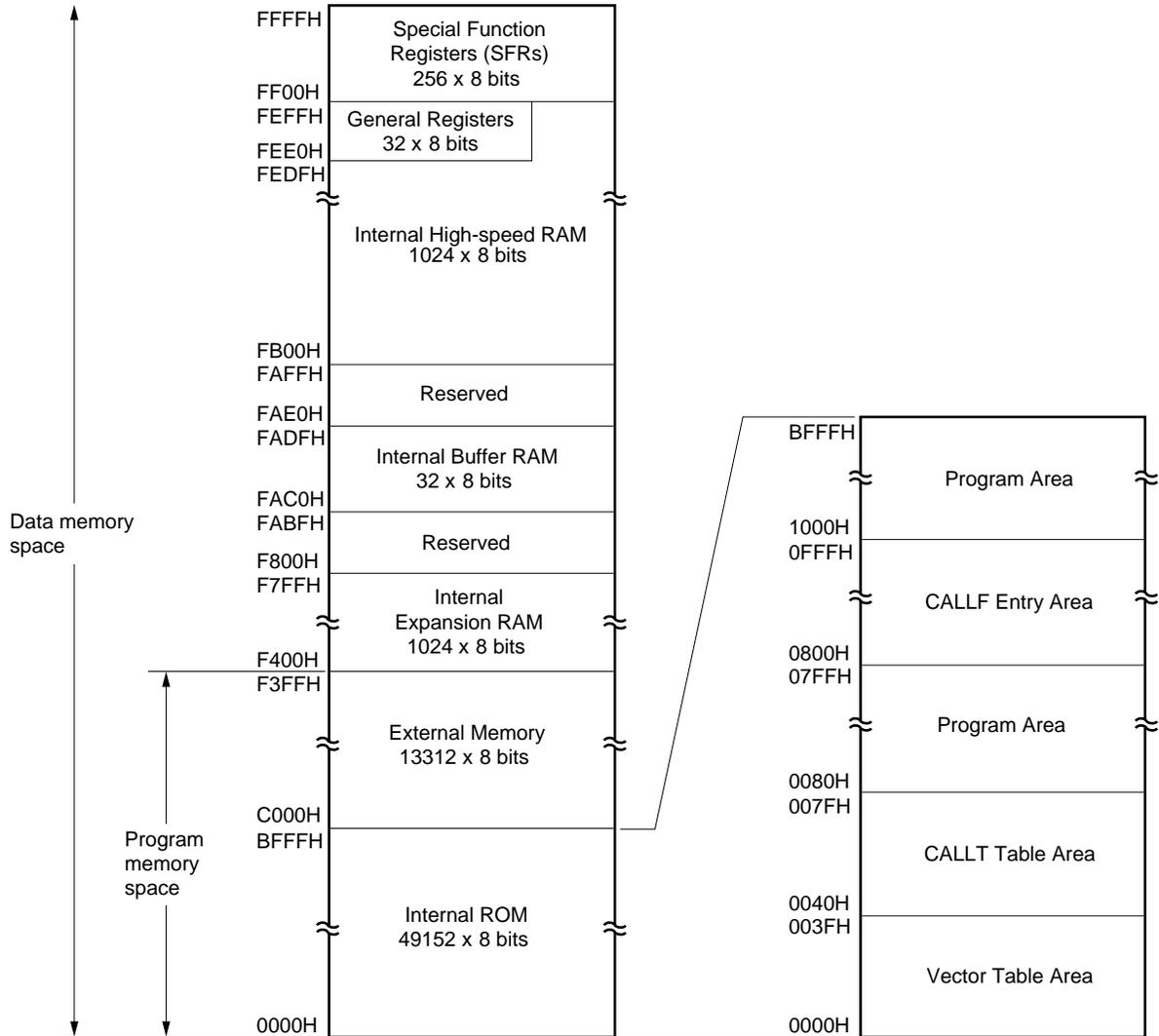
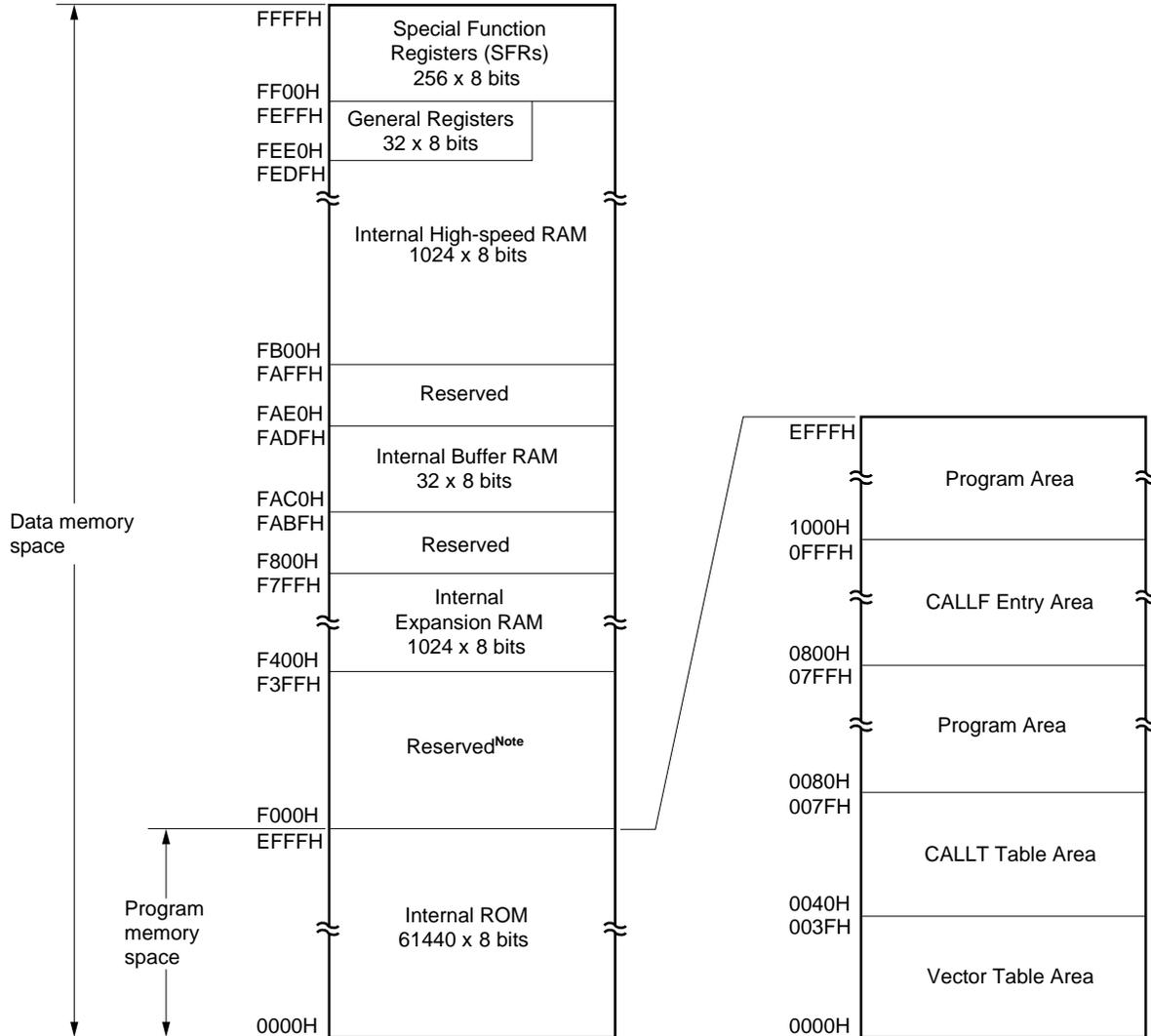
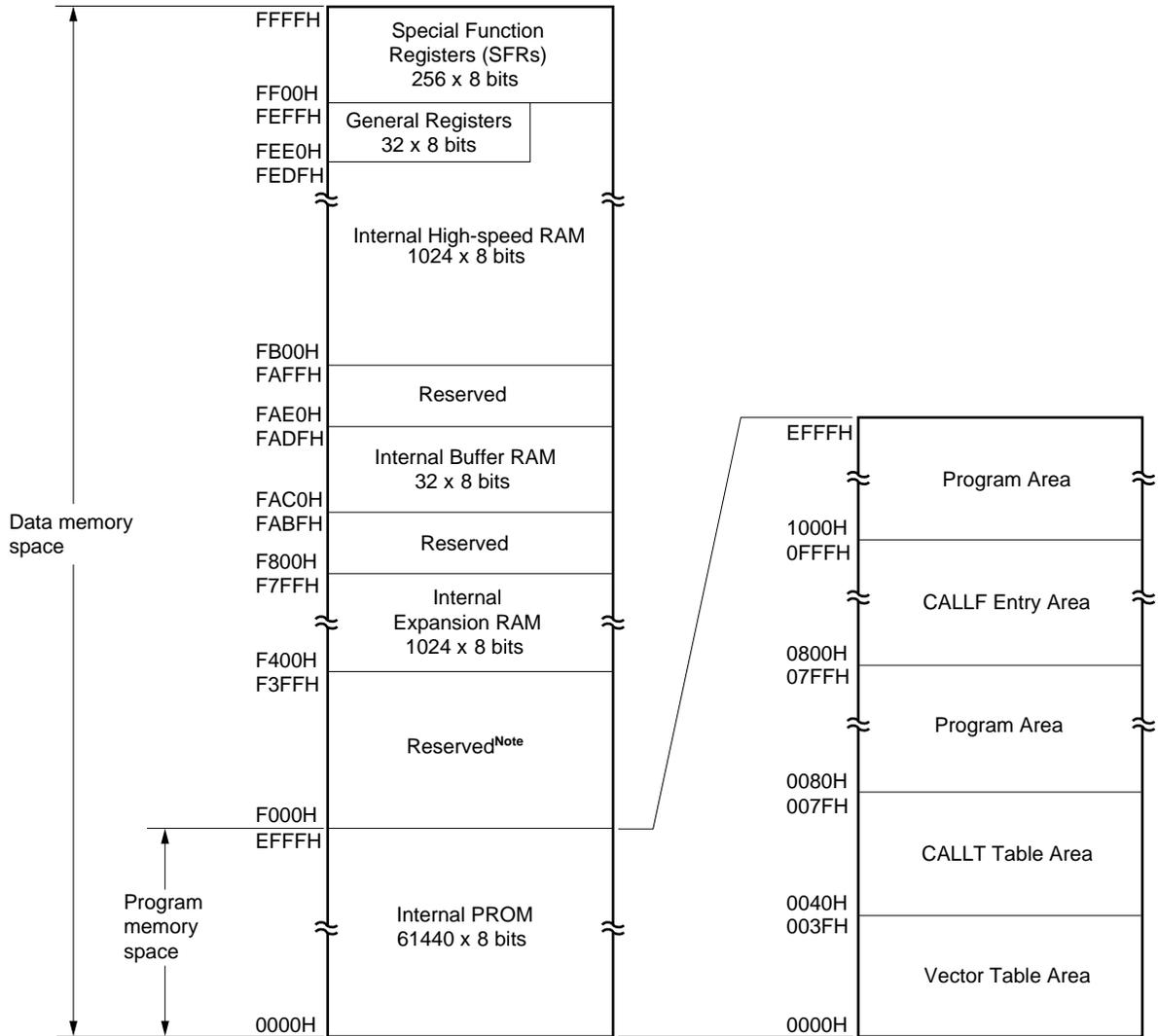


Figure 5-2. Memory Map (μ PD78078, 78078Y)



Note When internal ROM size is 60 Kbytes, the area F000H to F3FFH cannot be used. F000H to F3FFH can be used as external memory by setting the internal ROM size to less than 56 Kbytes by the internal memory size switching register.

Figure 5-3. Memory Map (μ PD78P078, μ PD78P078Y)



Note When internal PROM size is 60 Kbytes, the area F000H to F3FFH cannot be used. F000H to F3FFH can be used as external memory by setting the internal ROM size to less than 56 Kbytes by the internal memory size switching register.

5.1.1 Internal program memory space

The internal program memory space stores programs and table data. This is generally accessed by the program counter (PC). The μ PD78078 and 78078Y Subseries have various size of internal ROMs or PROM as shown below.

Table 5-1. Internal ROM Capacities

Part number	Internal ROM	
	Type	Capacity
μ PD78076, 78076Y	Mask ROM	49152 x 8 bits
μ PD78078, 78078Y		61440 x 8 bits
μ PD78P078, 78P078Y	PROM	

The internal program memory is divided into three areas: vector table area, CALLT instruction table area, and CALLF instruction table area. These areas are described on the next page.

(1) Vector table area

The 64-byte area 0000H to 003FH is reserved as a vector table area. The $\overline{\text{RESET}}$ 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 5-2. Vector Table

Vector Table Address	Interrupt Request
0000H	$\overline{\text{RESET}}$ input
0004H	INTWDT
0006H	INTP0
0008H	INTP1
000AH	INTP2
000CH	INTP3
000EH	INTP4
0010H	INTP5
0012H	INTP6
0014H	INTCSI0
0016H	INTCSI1
0018H	INTSER
001AH	INTSR/INTCSI2
001CH	INTST
001EH	INTTM3
0020H	INTTM00
0022H	INTTM01
0024H	INTTM1
0026H	INTTM2
0028H	INTAD
002AH	INTTM5
002CH	INTTM6
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).

5.1.2 Internal data memory space

The μ PD78078 and 78078Y Subseries units incorporate the following RAMs.

(1) Internal high-speed RAM

This is a 1024 x 8-bit configuration in the area FB00H to FEFH. 4 banks of general registers, each bank consisting of eight 8-bit registers, are allocated in the 32-byte area FEE0H to FEFH.

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

(2) Internal buffer RAM

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

(3) Internal expansion RAM

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

5.1.3 Special function register (SFR) area

An on-chip peripheral hardware special function register (SFR) is allocated in the area FF00H to FFFFH. (Refer to **Table 5-3. Special Function Register List** of **5.2.3 Special function register (SFR)**).

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

5.1.4 External memory space

The external memory space is accessible by setting the memory expansion mode register (MM). External memory space can store program, table data, etc. and allocate peripheral devices.

5.1.5 Data memory addressing

Addressing is a method to specify the instruction address to be executed next and the register and memory address to be manipulated when instructions are executed. The instruction address to be executed next is addressed by the program counter (PC) (for details, refer to **5.3 Instruction Address Addressing**).

For the addressing of the memory to be manipulated when instructions are executed, the μ PD78078 and 78078Y Subseries are provided with various addressing modes for optimum addressing. Special addressing methods are possible to meet the functions of the special function registers (SFRs) and general registers. The data memory space is the entire 64-Kbyte space (0000H to FFFFH). Figures 5-4 to 5-6 show the data memory addressing modes.

For details of addressing, refer to **5.4 Operand Address Addressing**.

Figure 5-4. Data Memory Addressing (μ PD78076, 78076Y)

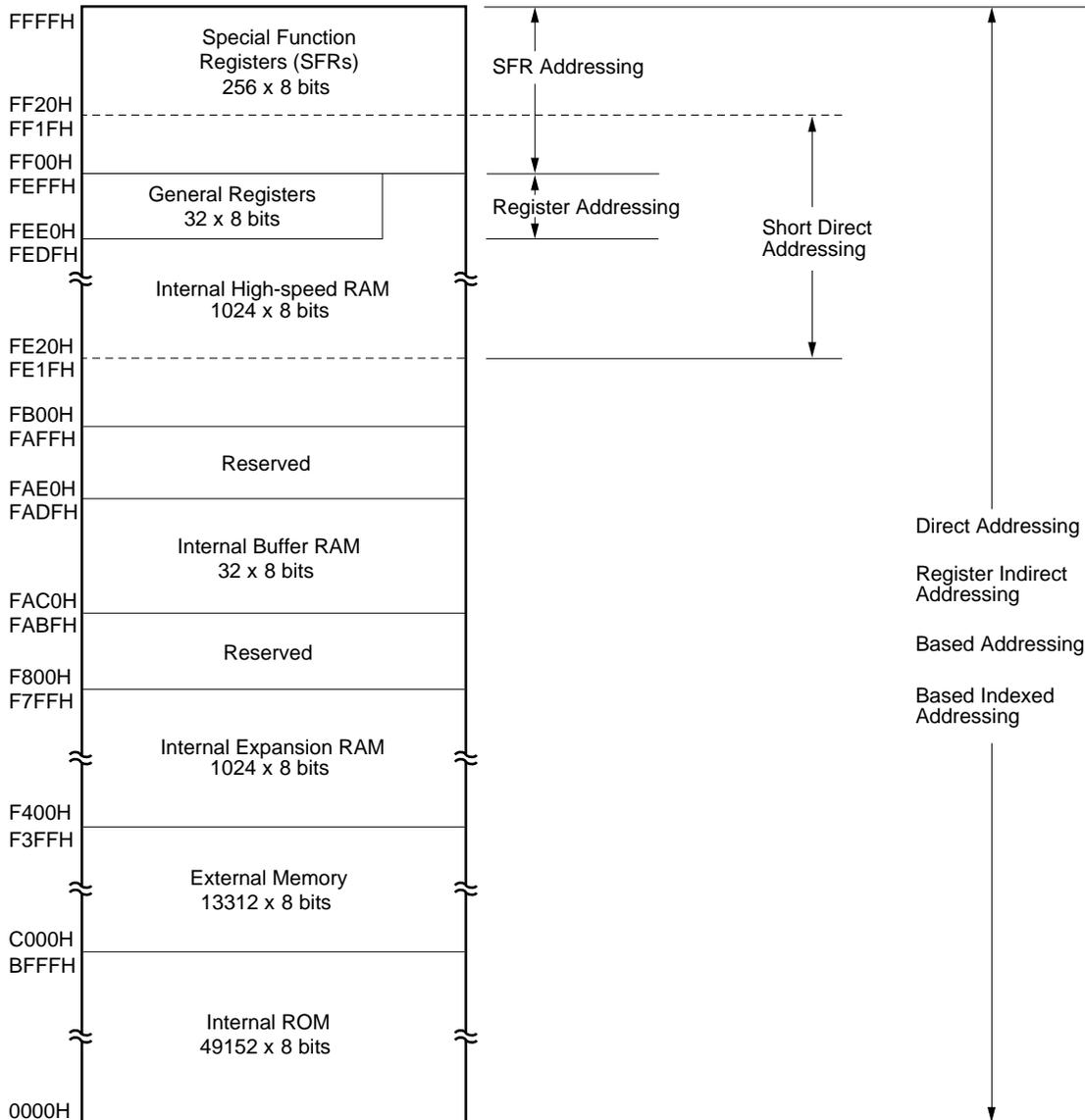
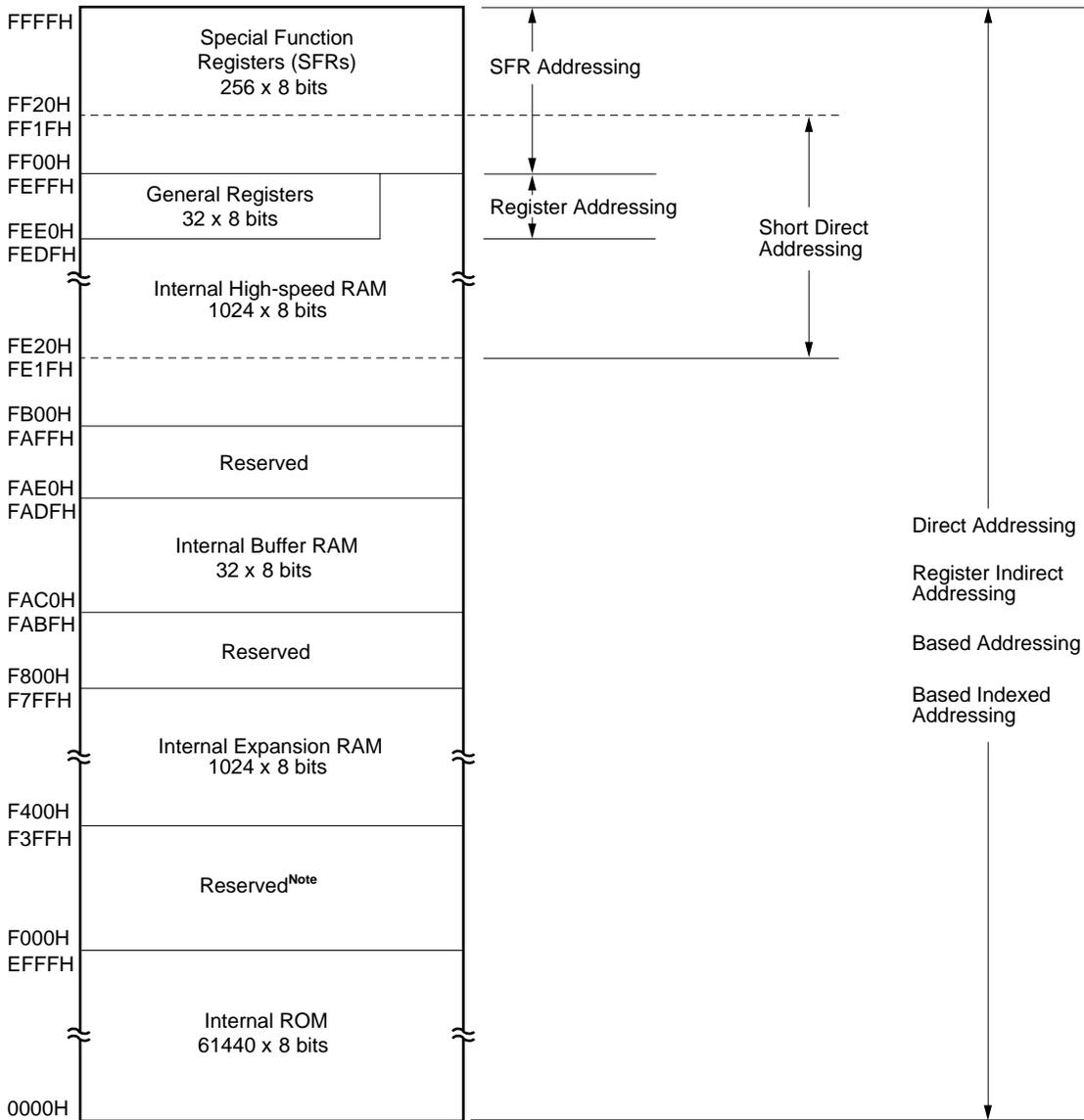
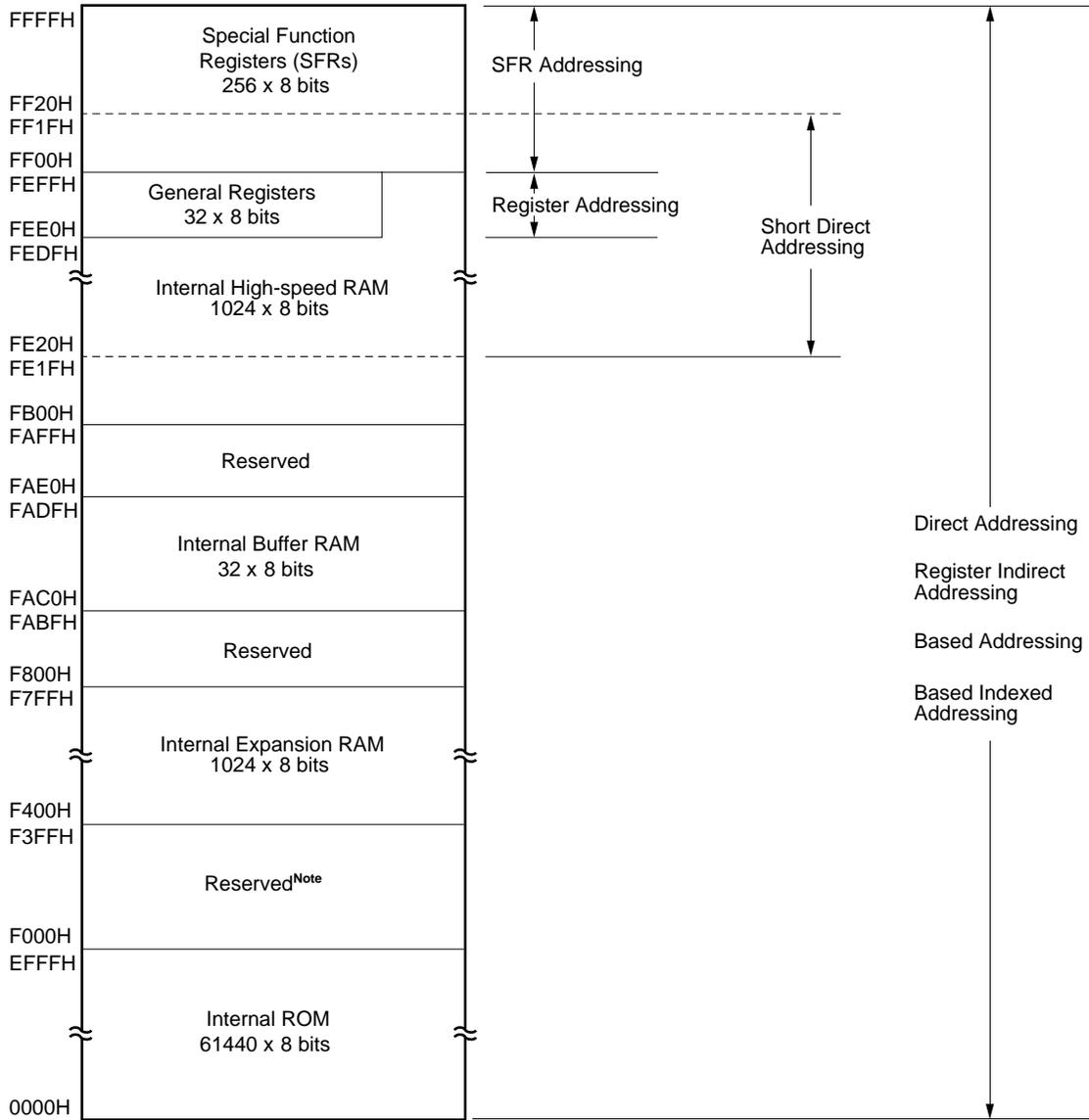


Figure 5-5. Data Memory Addressing (μ PD78078, 78078Y)



Note When internal ROM size is 60 Kbytes, the area F000H to F3FFH cannot be used. F000H to F3FFH can be used as external memory by setting the internal ROM size to less than 56 Kbytes by the internal memory size switching register (IMS).

Figure 5-6. Data Memory Addressing (μ PD78P078, 78P078Y)



Note When internal PROM size is 60 Kbytes, the area F000H to F3FFH cannot be used. F000H to F3FFH can be used as external memory by setting the internal ROM size to less than 56 Kbytes by the internal memory size switching register (IMS).

5.2 Processor Registers

The μ PD78078 and 78078Y Subseries units incorporate the following processor registers.

5.2.1 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).

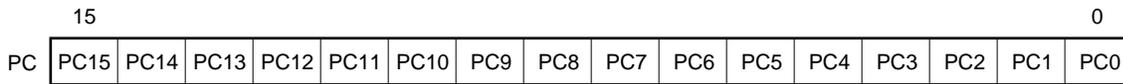
(1) Program counter (PC)

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.

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

Figure 5-7. Program Counter Configuration



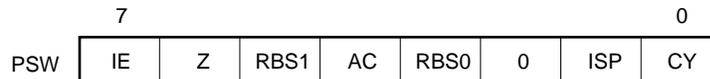
(2) Program status word (PSW)

The program status word is an 8-bit register consisting of various flags to be set/reset by instruction execution.

Program status word contents are automatically stacked upon interrupt request generation or PUSH PSW instruction execution and are automatically reset upon execution of the RETB, RETI and POP PSW instructions.

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

Figure 5-8. Program Status Word Configuration



(a) Interrupt enable flag (IE)

This flag controls the interrupt request acknowledge operations of the CPU.

When IE = 0, the IE is set to interrupt disabled (DI) status. All interrupts except non-maskable interrupt are disabled.

When IE = 1, the IE is set to interrupt enabled (EI) status and interrupt request acknowledge is controlled with an in-service priority flag (ISP), an interrupt mask flag for various interrupt sources, and a priority specification flag.

The IE is reset to (0) upon DI instruction execution or interrupt request 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, acknowledgment of the vectored interrupt request specified to low-order priority with the priority specify flag registers (PR0L, PR0H, and PR1L) (refer to **22.3 (3) Priority specify flag registers (PR0L, PR0H, and PR1L)**) is disabled. Whether an actual interrupt request is acknowledged or not is controlled with the interrupt enable flag (IE).

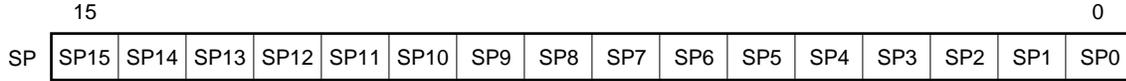
(f) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Figure 5-9. Stack Pointer Configuration



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

Each stack operation saves/resets data as shown in Figures 5-10 and 5-11.

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

Figure 5-10. Data to be Saved to Stack Memory

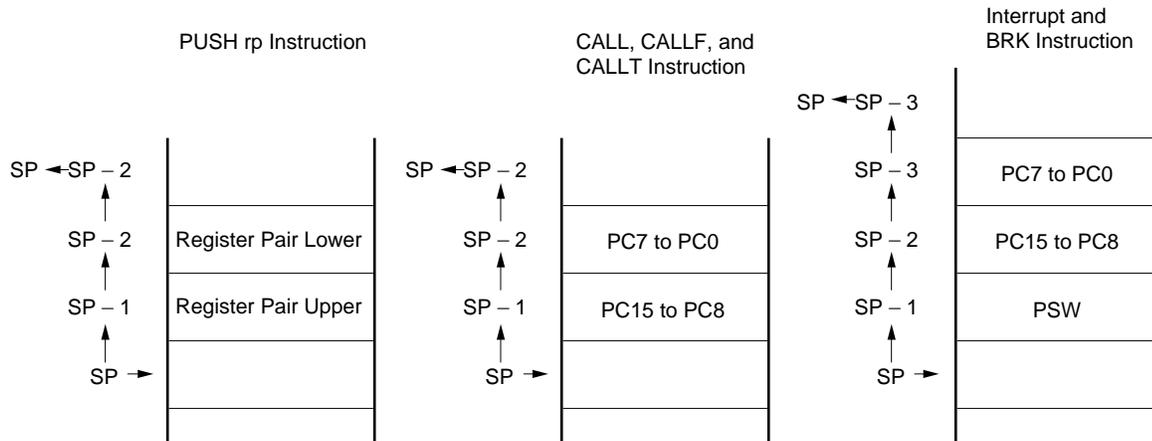
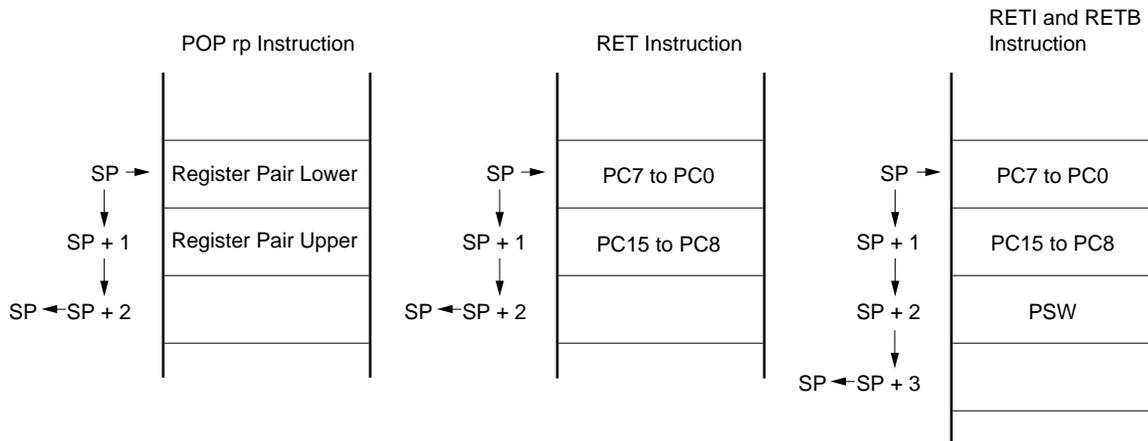


Figure 5-11. Data to be Reset from Stack Memory



5.2.2 General registers

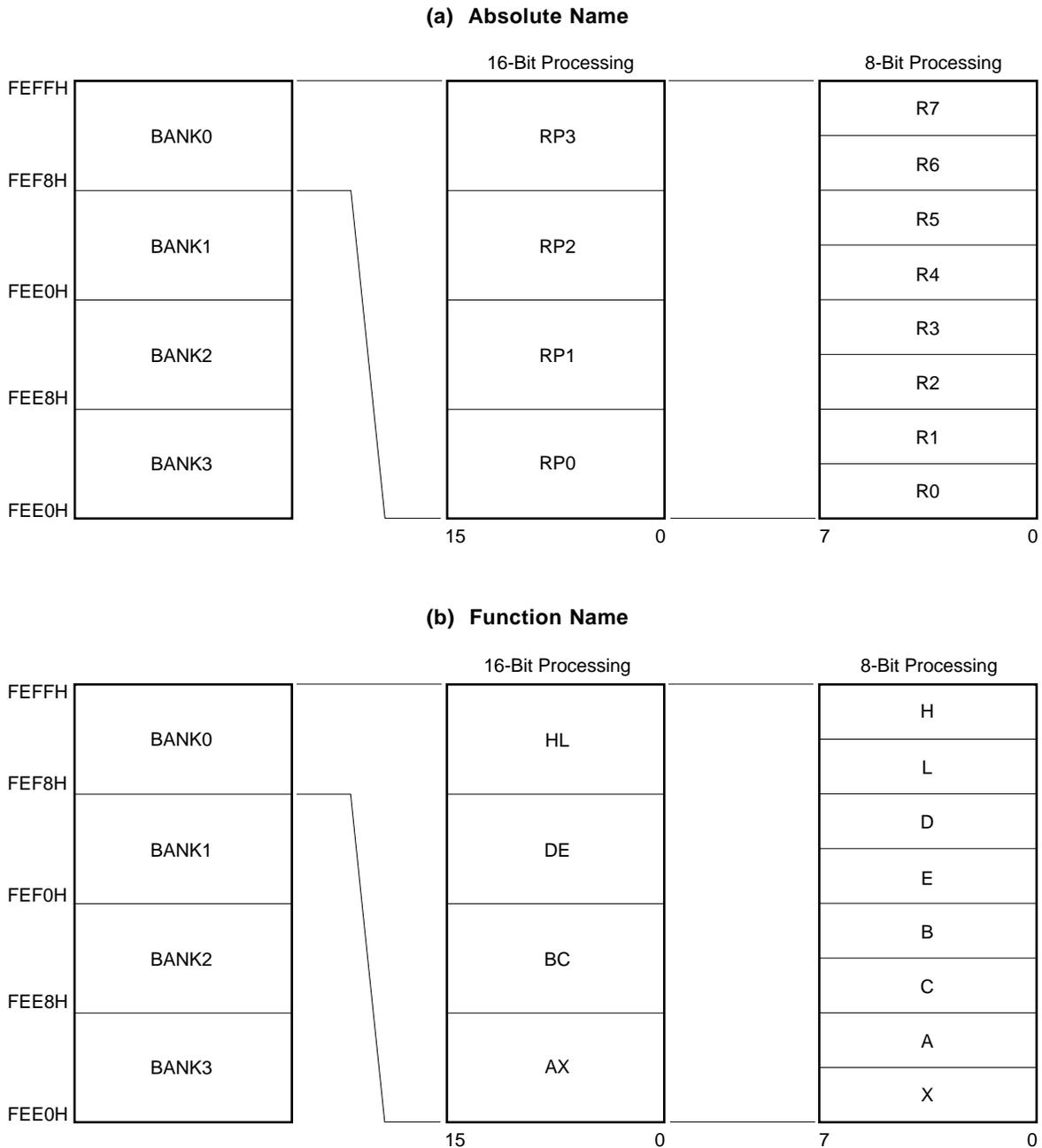
A general register is mapped at particular addresses (FEE0H to FEFFH) of the data memory. It consists of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

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

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

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

Figure 5-12. General Register Configuration



5.2.3 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 registers can be manipulated in a similar way as the general registers, by using operation, transfer, or bit-manipulate instructions. The special function registers are read from and written to in specified manipulation bit units (1, 8, and/or 16) depending on the register type.

Each manipulation bit unit can be specified as follows.

- **1-bit manipulation**

Describe the symbol reserved with assembler for the 1-bit manipulation instruction operand (sfr.bit).

This manipulation can also be specified with an address.

- **8-bit manipulation**

Describe the symbol reserved with assembler for the 8-bit manipulation instruction operand (sfr).

This manipulation can also be specified with an address.

- **16-bit manipulation**

Describe the symbol reserved with assembler for the 16-bit manipulation instruction operand (sfrp).

When addressing an address, describe an even address.

Table 5-3 gives a list of special function registers. The meaning of items in the table is as follows.

- **Symbol**

This column shows the addresses of the special function registers.

They have been defined as reserved words in the RA78K/0 and as the header file, sfrbit.h, in the CC78K/0.

★ They can be described as instruction operands when the RA78K/0, ID78K0-NS, ID78K0, and SM78K/0 are used.

- **R/W**

This column shows whether the corresponding special function register can be read or written.

R/W : Both reading and writing are enabled.

R : The value in the register can read out. A write to this register is ignored.

W : A value can be written to the register. Reading values from the register is impossible.

- **Manipulation**

The register can be manipulated in bit units marked with a check (✓) mark.

The register cannot be manipulated in bit units marked with “—”.

- **After reset**

The register is set to the value immediately after the $\overline{\text{RESET}}$ signal is input.

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulation			After Reset
				1 bit	8 bits	16 bits	
FF00H	Port0	P0	R/W	√	√	—	00H
FF01H	Port1	P1		√	√	—	
FF02H	Port2	P2		√	√	—	
FF03H	Port3	P3		√	√	—	
FF04H	Port4	P4		√	√	—	Undefined
FF05H	Port5	P5		√	√	—	
FF06H	Port6	P6		√	√	—	
FF07H	Port7	P7		√	√	—	00H
FF08H	Port8	P8		√	√	—	
FF09H	Port9	P9		√	√	—	
FF0AH	Port10	P10		√	√	—	
FF0CH	Port12	P12		√	√	—	
FF0DH	Port13	P13		√	√	—	
FF10H	Capture/compare register 00	CR00	—	—	√	Undefined	
FF11H			—	—	√		
FF12H	Capture/compare register 01	CR01	—	—	√	0000H	
FF13H			—	—	√		
FF14H	16-bit timer register	TM0	R	—	—	√	0000H
FF15H			—	—	√		
FF16H	Compare register 10	CR10	R/W	—	√	—	Undefined
FF17H	Compare register 20	CR20		—	√	—	
FF18H	8-bit timer register 1	TMS	R	—	√	√	00H
FF19H	8-bit timer register 2	TM2		—	√	—	
FF1AH	Serial I/O shift register 0	SIO0	R/W	—	√	—	Undefined
FF1BH	Serial I/O shift register 1	SIO1		—	√	—	
FF1FH	A/D conversion result register	ADCR	R	—	√	—	FFH
FF20H	Port mode register 0	PM0	R/W	√	√	—	
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		√	√	—	
FF27H	Port mode register 7	PM7		√	√	—	
FF28H	Port mode register 8	PM8		√	√	—	
FF29H	Port mode register 9	PM9		√	√	—	
FF2AH	Port mode register 10	PM10		√	√	—	
FF2CH	Port mode register 12	PM12		√	√	—	
FF2DH	Port mode register 13	PM13		√	√	—	

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulation			After Reset	
				1 bit	8 bits	16 bits		
FF30H	Real-time output buffer register L	RTBL	R/W	—	√	—	00H	
FF31H	Real-time output buffer register H	RTBH		—	√	—		
FF34H	Real-time output port mode register	RTPM		√	√	—		
FF36H	Real-time output port control register	RTPC		√	√	—		
FF38H FF39H	Correction address register 0	CORAD0		—	—	√		
FF3AH FF3BH	Correction address register 1	CORAD1		—	—	√		
FF3FH	External bus type selection register	EBTS		—	√	—		
FF40H	Timer clock select register 0	TCL0		√	√	—		
FF41H	Timer clock select register 1	TCL1		—	√	—		
FF42H	Timer clock select register 2	TCL2		—	√	—		
FF43H	Timer clock select register 3	TCL3		—	√	—		88H
FF47H	Sampling clock select register	SCS		—	√	—		00H
FF48H	16-bit timer mode control register	TMC0		√	√	—		
FF49H	8-bit timer mode control register 1	TMC1		√	√	—		
FF4AH	Watch timer mode control register	TMC2	√	√	—			
FF4CH	Capture/compare control register 0	CRC0	√	√	—	04H		
FF4EH	16-bit timer output control register	TOC0	√	√	—	00H		
FF4FH	8-bit timer output control register	TOC1	√	√	—			
FF50H	Compare register 50	CR50	—	√	—			
FF51H	8-bit timer register 5	TM5	R	—	√	—		
FF52H	Timer clock selection register 5	TCL5	R/W	—	√	—		
FF53H	8-bit timer mode control register 5	TMC5		√	√	—		
FF54H	Compare register 60	CR60		—	√	—		
FF55H	8-bit timer register 6	TM6	R	—	√	—		
FF56H	Timer clock selection register 6	TCL6	R/W	—	√	—		
FF57H	8-bit timer mode control register 6	TMC6		√	√	—		
FF60H	Serial operating mode register 0	CSIM0		√	√	—		
FF61H	Serial bus interface control register	SBIC	√	√	—			
FF62H	Slave address register	SVA	—	√	—	Undefined		
FF63H	Interrupt timing specify register	SINT	√	√	—	00H		
FF68H	Serial operating mode register 1	CSIM1	√	√	—			
FF69H	Automatic data transmit/receive control register	ADTC	√	√	—			
FF6AH	Automatic data transmit/receive address pointer	ADTP	—	√	—			
FF6BH	Automatic data transmit/receive interval specify register	ADTI	√	√	—			
FF70H	Asynchronous serial interface mode register	ASIM	√	√	—			
FF71H	Asynchronous serial interface status register	ASIS	R	—	√		—	
FF72H	Serial operating mode register 2	CSIM2	R/W	√	√		—	
FF73H	Baud rate generator control register	BRGC		—	√		—	

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulation			After Reset
					1 bit	8 bits	16 bits	
FF74H	Transmit shift register	TXS	SIO2	W	—	√	—	FFH
	Receive buffer register	RXB		R				
FF80H	A/D converter mode register	ADM		R/W	√	√	—	01H
FF84H	A/D converter input select register	ADIS			—	√	—	00H
FF8AH	Correction control register	CORCN			√	√	—	
FF90H	D/A conversion value set register 0	DACS0			—	√	—	
FF91H	D/A conversion value set register 1	DACS1			—	√	—	
FF98H	D/A converter mode register	DAM			√	√	—	
FFD0H to FDFH	External access area ^{Note 1}				√	√	—	Undefined
FFE0H	Interrupt request flag register 0L	IF0	IF0L		√	√	√	00H
FFE1H	Interrupt request flag register 0H		IF0H		√	√		
FFE2H	Interrupt request flag register 1L	IF1L			√	√	—	
FFE4H	Interrupt mask flag register 0L	MK0	MK0L	√	√	√	FFH	
FFE5H	Interrupt mask flag register 0H		MK0H	√	√			
FFE6H	Interrupt mask flag register 1L	MK1L		√	√	—		
FFE8H	Priority order specify flag register 0L	PR0	PR0L	√	√	√		
FFE9H	Priority order specify flag register 0H		PR0H	√	√			
FFEAH	Priority order specify flag register 1L	PR1L		√	√	—		
FFECH	External interrupt mode register 0	INTM0		—	√	—	00H	
FFEDH	External interrupt mode register 1	INTM1		—	√	—		
FFF0H	Internal memory size switching register	IMS		—	√	—	Note 2	
FFF2H	Oscillation mode selection register	OSMS		W	—	√	00H	
FFF3H	Pull-up resistor option register H	PUOH		R/W	√	√		
FFF4H	Internal expansion RAM size switching register	IXS		W	—	√	0AH ^{Note 3}	
FFF6H	Key return mode register	KRM		R/W	√	√	—	02H
FFF7H	Pull-up resistor option register L	PUOL			√	√	—	00H
FFF8H	Memory expansion mode register	MM			√	√	—	10H
FFF9H	Watchdog timer mode register	WDTM			√	√	—	00H
FFFAH	Oscillation stabilization time select register	OSTS			—	√	—	04H
FFFBH	Processor clock control register	PCC			√	√	—	

- Notes**
- The external access area cannot be accessed in SFR addressing. Access the area through direct addressing.
 - The value after reset depends on products.
 μ PD78076,78076Y: CCH, μ PD78078, 78078Y: CFH, μ PD78P078, 78P078Y: CFH
 When a mask ROM version is used, do not set to IMS any value other than that determined at reset, unless external device expansion function is used with the μ PD78078 or 78078Y.
 - When a mask ROM version is used, do not set to IXS any value other than that determined at reset.

5.3 Instruction Address Addressing

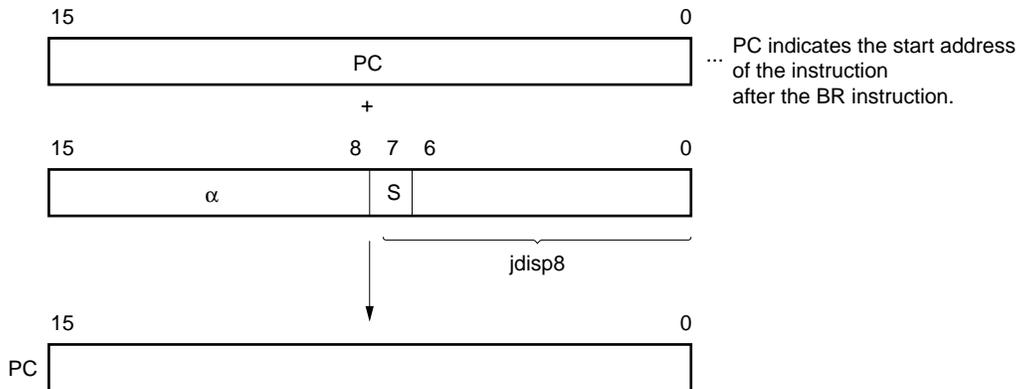
An instruction address is determined by program counter (PC) contents. The PC contents are normally incremented (+1 for each byte) automatically according to the number of bytes of an instruction to be fetched each time another instruction is executed. However, when a branch instruction is executed, 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—Instructions (U12326E)**).

5.3.1 Relative addressing

[Function]

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

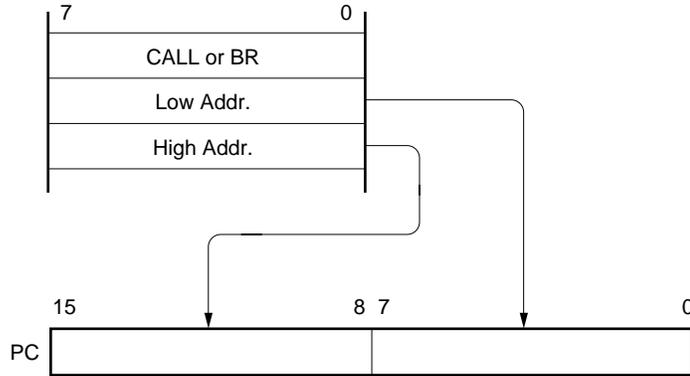
5.3.2 Immediate addressing

[Function]

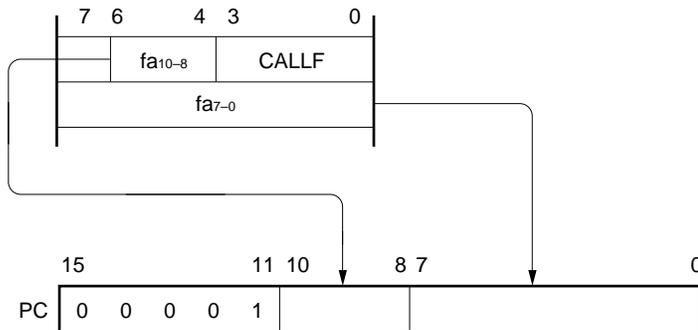
Immediate data in the instruction word is transferred to the program counter (PC) and branched. This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed. CALL !addr16 and BR !addr16 instructions can branch to all the memory space. CALLF !addr11 instruction branches to the area from 0800H to 0FFFH.

[Illustration]

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



In the case of CALLF !addr11 instruction



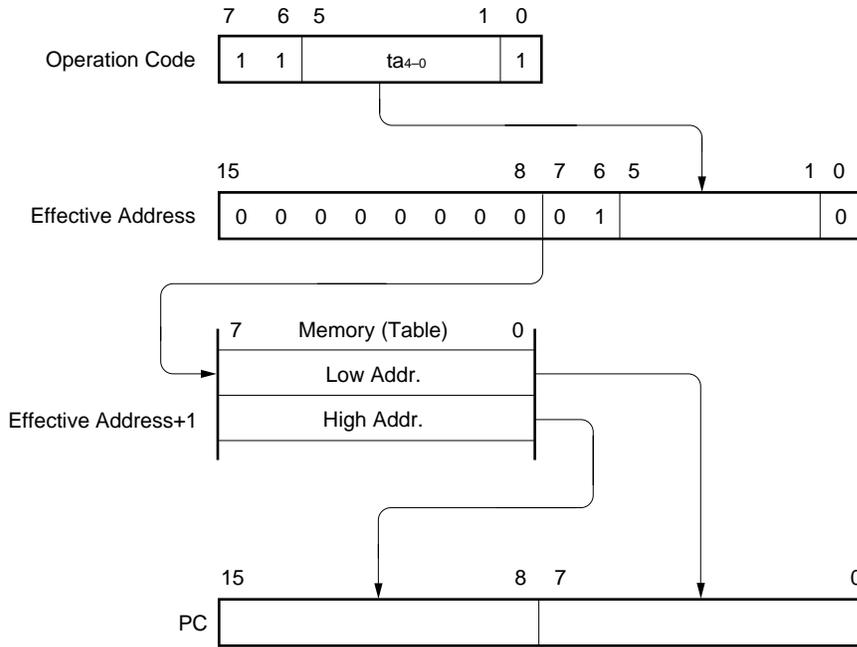
5.3.3 Table indirect addressing

[Function]

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

Table indirect addressing is carried out when the CALLT [addr5] instruction is executed. This instruction can refer to the address stored in the memory table 40H to 7FH and branch to all the memory space.

[Illustration]

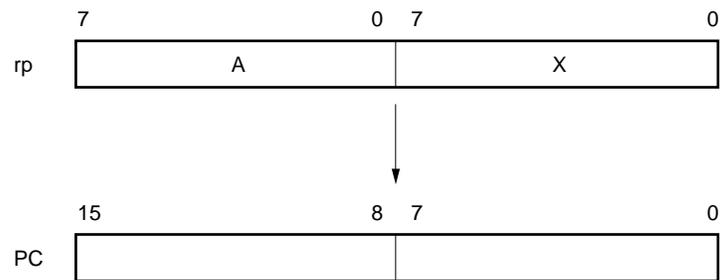


5.3.4 Register addressing

[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]

5.4 Operand Address Addressing

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

5.4.1 Implied addressing

[Function]

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

Of the μ PD78078 and 78078Y 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.

[Description example]

In the case of MULU X

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

5.4.2 Register addressing

[Function]

The general register is accessed as an operand. The general register to be accessed is specified with register bank select flags (RBS0 and RBS1) and register specify code (Rn, Rpn) in the instruction code.

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

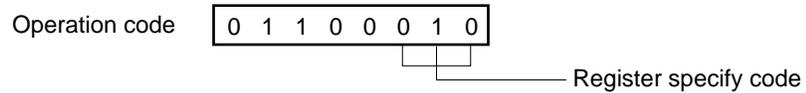
[Operand format]

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

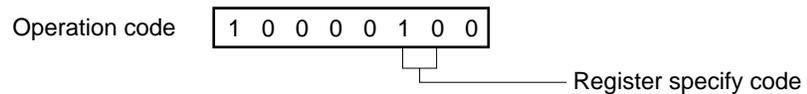
'r' and 'rp' can be described with function names (X, A, C, B, E, D, L, H, AX, BC, DE and HL) as well as absolute names (R0 to R7 and RP0 to RP3).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



5.4.3 Direct addressing

[Function]

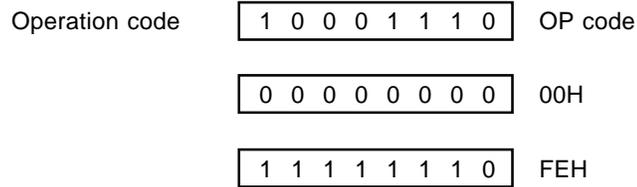
The memory indicated by immediate data in an instruction word is directly addressed.

[Operand format]

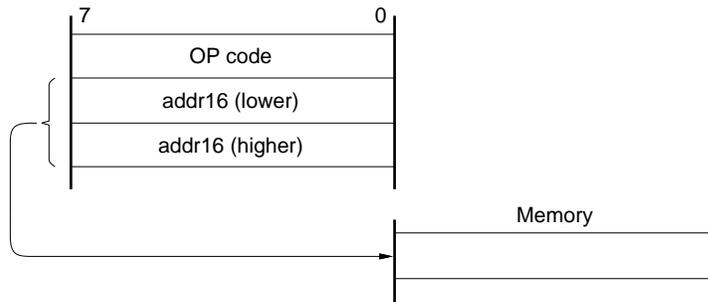
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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



[Illustration]



5.4.4 Short direct addressing

[Function]

The memory to be manipulated in the fixed space is directly addressed with 8-bit data in an instruction word. The fixed space to which this addressing is applied to is the 256-byte space, from FE20H to FF1FH. An internal high-speed RAM and a special function register (SFR) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.

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

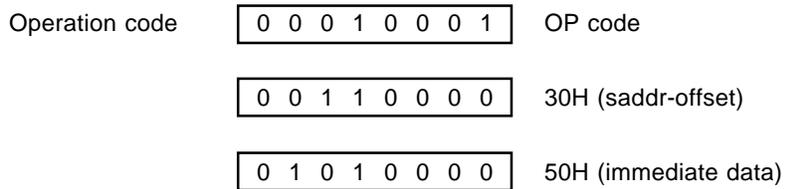
When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is set to 0. When it is at 00H to 1FH, bit 8 is set to 1. Refer to [Illustration] below.

[Operand format]

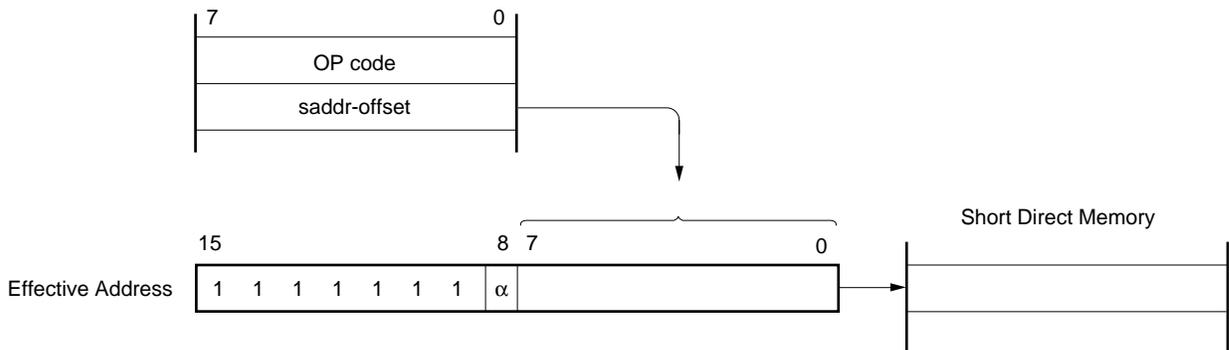
Identifier	Description
saddr	Label of FE20H to FF1FH immediate data
saddrp	Label of FE20H to FF1FH immediate data (even address only)

[Description example]

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



[Illustration]



When 8-bit immediate data is 20H to FFH, $\alpha = 0$

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

5.4.6 Register indirect addressing

[Function]

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

[Operand format]

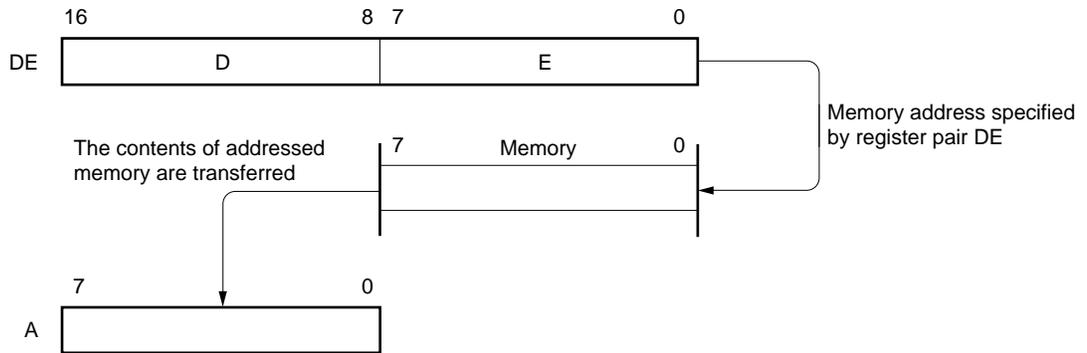
Identifier	Description
—	[DE], [HL]

[Description example]

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

Operation code 1 0 0 0 0 1 0 1

[Illustration]



5.4.7 Based addressing

[Function]

8-bit immediate data is added to the contents of the base register, that is, the HL register pair, and the sum is used to address the memory. The HL register pair to be accessed is in the register bank specified with the register bank select flags (RBS0 and RBS1). Addition is performed by expanding the offset data as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all the memory spaces.

[Operand format]

Identifier	Description
—	[HL + byte]

[Description example]

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

Operation code

1 0 1 0 1 1 1 0

0 0 0 1 0 0 0 0

5.4.8 Based indexed addressing

[Function]

The B or C register contents specified in an instruction are added to the contents of the base register, that is, the HL register pair, and the sum is used to address the memory. The HL, B, and C registers to be accessed are registers in the register bank specified with the register bank select flag (RBS0 and RBS1).

Addition is performed by expanding the contents of the B or C register as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all the memory spaces.

[Operand format]

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

[Description example]

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

Operation code

1 0 1 0 1 0 1 1

5.4.9 Stack addressing

[Function]

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

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

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

[Description example]

In the case of PUSH DE

Operation code

1 0 1 1 0 1 0 1

[MEMO]

CHAPTER 6 PORT FUNCTIONS

6.1 Port Functions

The μ PD78078 and 78078Y Subseries units incorporate two input ports and eighty-six input/output ports. Figure 6-1 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 6-1. Port Types

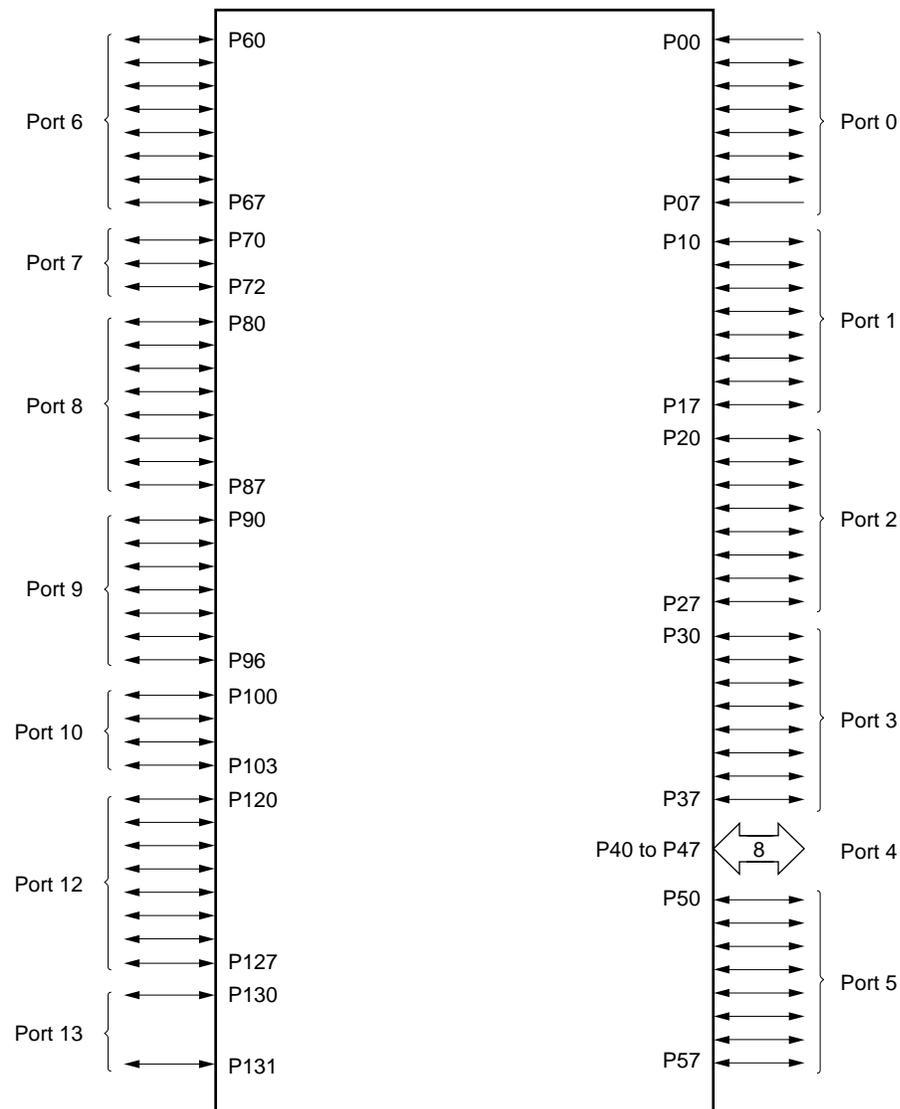


Table 6-1. Port Functions (μ PD78078 Subseries) (1/2)

Pin Name	Function	Alternate Function	
P00	Port 0. 8-bit input/output port.	Input only	
P01		INTP0/TI00	
P02		Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.	INTP1/TI01
P03			INTP2
P04			INTP3
P05			INTP4
P06		INTP5	
P07		INTP6	
P07	Input only	XT1	
P10 to P17	Port 1. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.	ANI0 to ANI7	
P20	Port 2. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.	SI1	
P21		SO1	
P22		$\overline{\text{SCK1}}$	
P23		STB	
P24		BUSY	
P25		SI0/SB0	
P26		SO0/SB1	
P27		$\overline{\text{SCK0}}$	
P30	Port 3. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.	TO0	
P31		TO1	
P32		TO2	
P33		TI1	
P34		TI2	
P35		PCL	
P36		BUZ	
P37		—	
P40 to P47	Port 4. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software. Test input flag (KRIF) is set to 1 by falling edge detection.	AD0 to AD7	
P50 to P57	Port 5. 8-bit input/output port. LEDs can be driven directly. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.	A8 to A15	

Table 6-1. Port Functions (μ PD78078 Subseries) (2/2)

Pin Name	Function		Alternate Function
P60	Port 6. 8-bit input/output port. Input/output mode can be specified bit-wise.	N-ch open drain input/output port. On-chip pull-up resistor can be specified by mask option. (Mask ROM version only). LEDs can be driven directly.	—
P61		If used as an input port, an on-chip pull-up resistor can be connected by software.	
P62			
P63			
P64			\overline{RD}
P65		\overline{WR}	
P66		\overline{WAIT}	
P67		ASTB	
P70	Port 7. 3-bit input/output port.	If used as an input port, an on-chip pull-up resistor can be connected by software.	SI2/RxD
P71	Input/output mode can be specified bit-wise.		SO2/TxD
P72	If used as an input port, an on-chip pull-up resistor can be connected by software.		$\overline{SCK2/ASCK}$
P80 to P87	Port 8. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		A0 to A7
P90	Port 9. 7-bit input/output port. Input/output mode can be specified bit-wise.	N-ch open-drain input/output port. On-chip pull-up resistor can be specified by mask option. (Mask ROM version only). LEDs can be driven directly.	—
P91		If used as an input port, an on-chip pull-up resistor can be connected by software.	
P92			
P93			
P94			
P95			
P96			
P100	Port 10. 4-bit input/output port.	If used as an input port, an on-chip pull-up resistor can be connected by software.	TI5/TO5
P101	Input/output mode can be specified bit-wise.		TI6/TO6
P102, 103			—
P120 to P127	Port 12. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		RTP0 to RTP7
P130, P131	Port 13. 2-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		ANO0, ANO1

Table 6-2. Port Functions (μ PD78078Y Subseries) (1/2)

Pin Name	Function	Alternate Function	
P00	Port 0. 8-bit input/output port.	Input only	
P01		INTP0/TI00	
P02		Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.	INTP1/TI01
P03			INTP2
P04			INTP3
P05			INTP4
P06			INTP5
P07		Input only	INTP6
P10 to P17	Port 1. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.	XT1	
P20	Port 2. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.	ANI0 to ANI7	
P21		SI1	
P22		SO1	
P23		SCK1	
P24		STB	
P25		BUSY	
P26		SI0/SB0/SDA0	
P27		SO0/SB1/SDA1	
P30	Port 3. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.	SCK0/SCL	
P31		TO0	
P32		TO1	
P33		TO2	
P34		TI1	
P35		TI2	
P36		PCL	
P37		BUZ	
P40 to P47	Port 4. 8-bit input/output port. Input/output mode can be specified 8-bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software. Test input flag (KRIF) is set to 1 by falling edge detection.	—	
P50 to P57	Port 5. 8-bit input/output port. LEDs can be driven directly. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.	AD0 to AD7	
		A8 to A15	

Table 6-2. Port Functions (μ PD78078Y Subseries) (2/2)

Pin Name	Function		Alternate Function
P60	Port 6. 8-bit input/output port. Input/output mode can be specified bit-wise.	N-ch open drain input/output port. On-chip pull-up resistor can be specified by mask option. (Mask ROM version only). LEDs can be driven directly.	—
P61		If used as an input port, an on-chip pull-up resistor can be connected by software.	
P62			
P63			
P64			
P65			
P66			
P67			
P70	Port 7. 3-bit input/output port. Input/output mode can be specified bit-wise.		
P71			SO2/TxD
P72		If used as an input port, an on-chip pull-up resistor can be connected by software.	$\overline{\text{SCK2/ASCK}}$
P80 to P87	Port 8. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		A0 to A7
P90	Port 9. 7-bit input/output port. Input/output mode can be specified bit-wise.	N-ch open drain input/output port. On-chip pull-up resistor can be specified by mask option. (Mask ROM version only). LEDs can be driven directly.	—
P91		If used as an input port, an on-chip pull-up resistor can be connected by software.	
P92			
P93			
P94			
P95			
P96			
P100	Port 10. 4-bit input/output port. Input/output mode can be specified bit-wise.		
P101			TI6/TO6
P102, 103		If used as an input port, an on-chip pull-up resistor can be connected by software.	—
P120 to P127	Port 12. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		RTP0 to RTP7
P130, P131	Port 13. 2-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.		ANO0, ANO1

6.2 Port Configuration

A port consists of the following hardware:

Table 6-3. Port Configuration

Item	Configuration
Control register	Port mode register (PMm: m = 0 to 3, 5 to 10, 12, 13) Pull-up resistor option register (PUOH, PUOL) Memory expansion mode register (MM) ^{Note} Key return mode register (KRM)
Port	Total: 88 ports (2 inputs, 86 inputs/outputs)
Pull-up resistor	<ul style="list-style-type: none"> • Mask ROM versions..... Total: 86 pins (software-specifiable for 78 pins and mask option for 8 pins) • μPD78P078 (or 78P078Y) Total: 78 pins

Note MM specifies port 4 input/output.

6.2.1 Port 0

Port 0 is an 8-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 and P07 pins are input-only ports. When P01 to P06 pins are used as input ports, an on-chip pull-up resistor can be used in 6-bit units with a pull-up resistor option register L (PUOL).

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

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

Figures 6-2 and 6-3 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 6-2. Block Diagram of P00 and P07

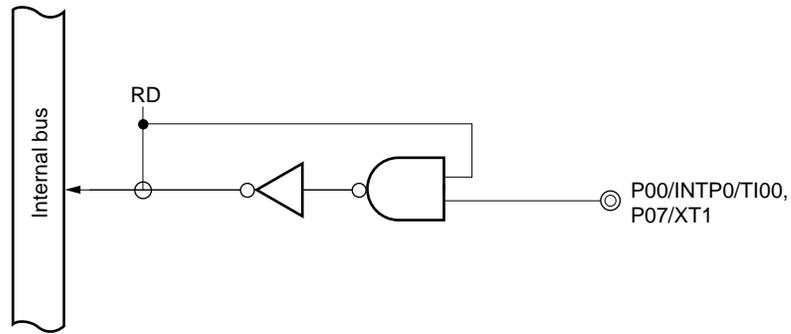
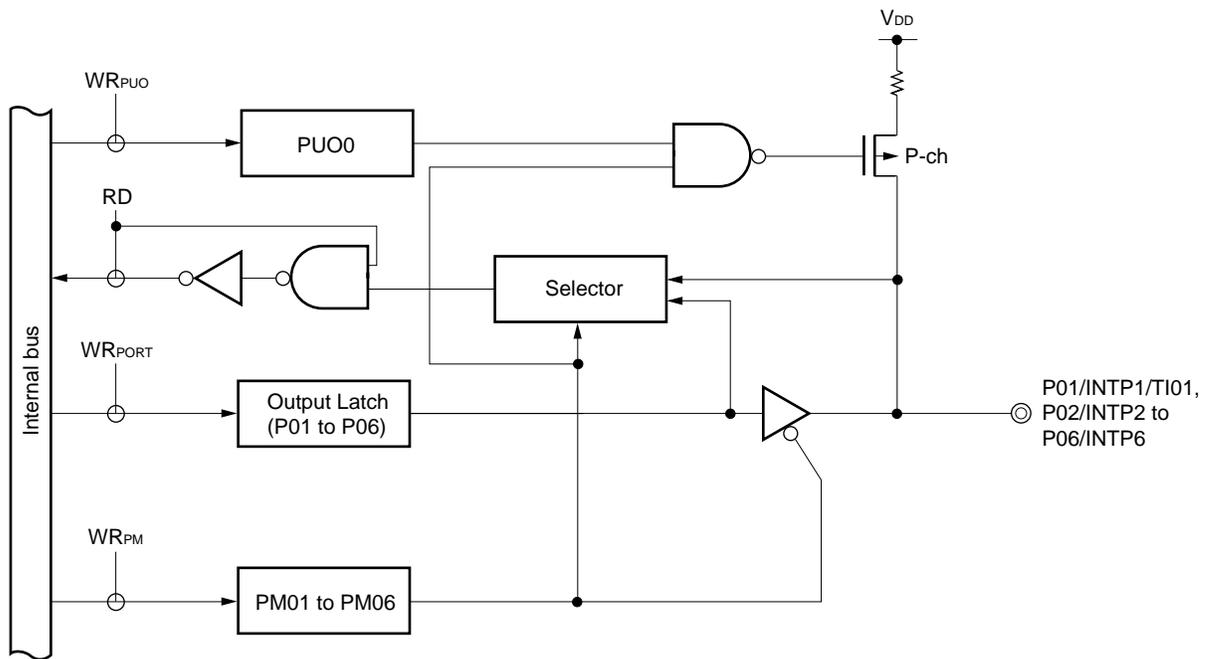


Figure 6-3. Block Diagram of P01 to P06



- PUO : Pull-up resistor option register
- PM : Port mode register
- RD : Port 0 read signal
- WR : Port 0 write signal

6.2.2 Port 1

Port 1 is an 8-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). When P10 to P17 pins are used as input ports, an on-chip pull-up resistor can be connected to them in 8-bit units with a pull-up resistor option register L (PUOL).

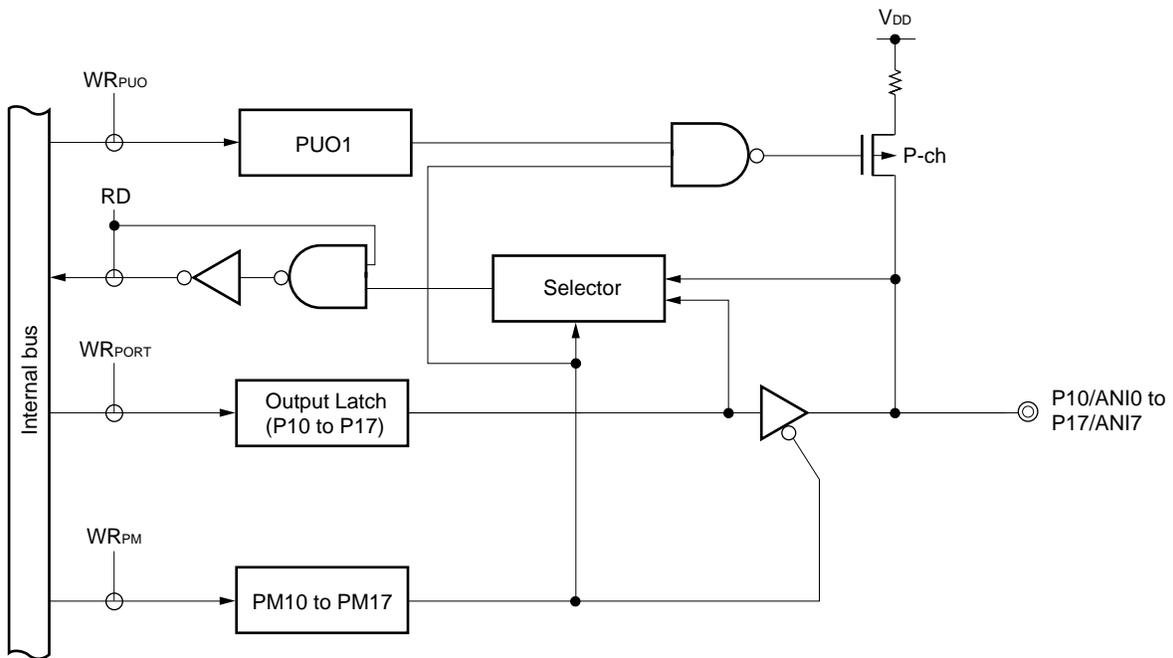
Dual-functions include an A/D converter analog input.

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

Figure 6-4 shows a block diagram of port 1.

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

Figure 6-4. Block Diagram of P10 to P17



PUO : Pull-up resistor option register

PM : Port mode register

RD : Port 1 read signal

WR : Port 1 write signal

6.2.3 Port 2 (μ PD78078 Subseries)

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

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

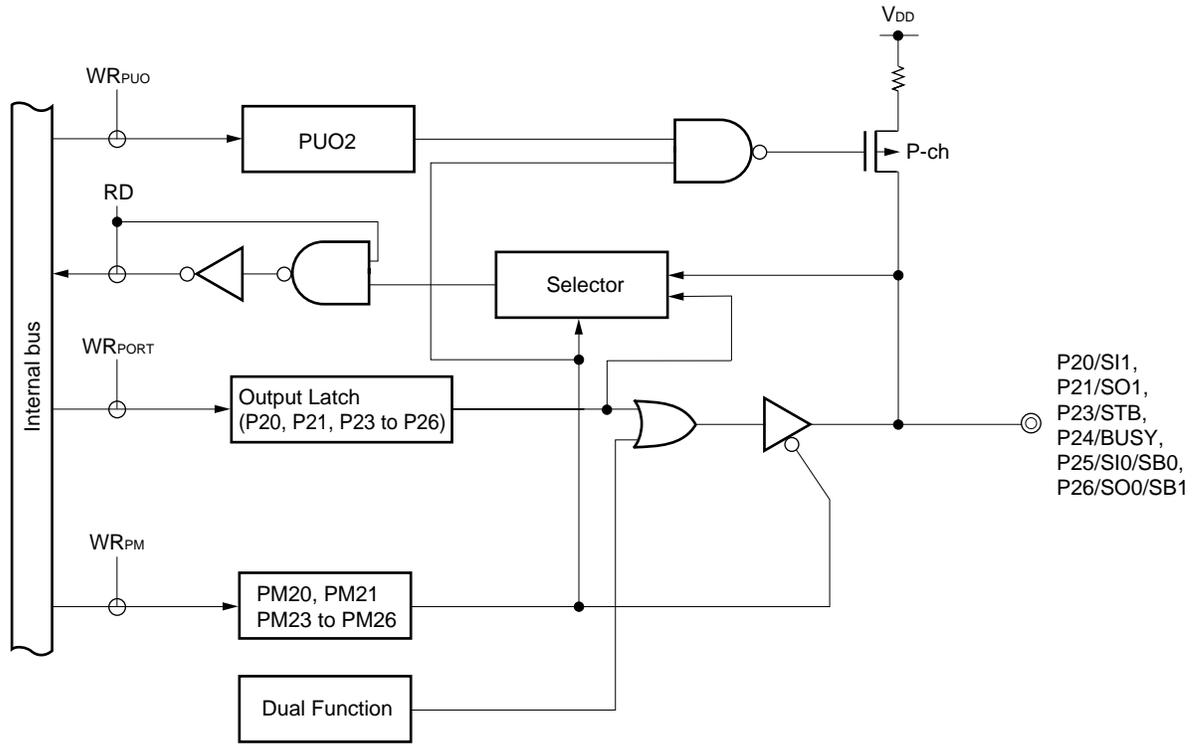
$\overline{\text{RESET}}$ input sets port 2 to input mode.

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

- Cautions 1.** When used as a serial interface, set the input/output and output latch according to its functions. For the setting method, refer to Figure 17-4. Serial Operating Mode Register 0 Format and Figure 19-3. 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 (11) SBI mode precautions (e) in 17.4.3 SBI mode operation).

★

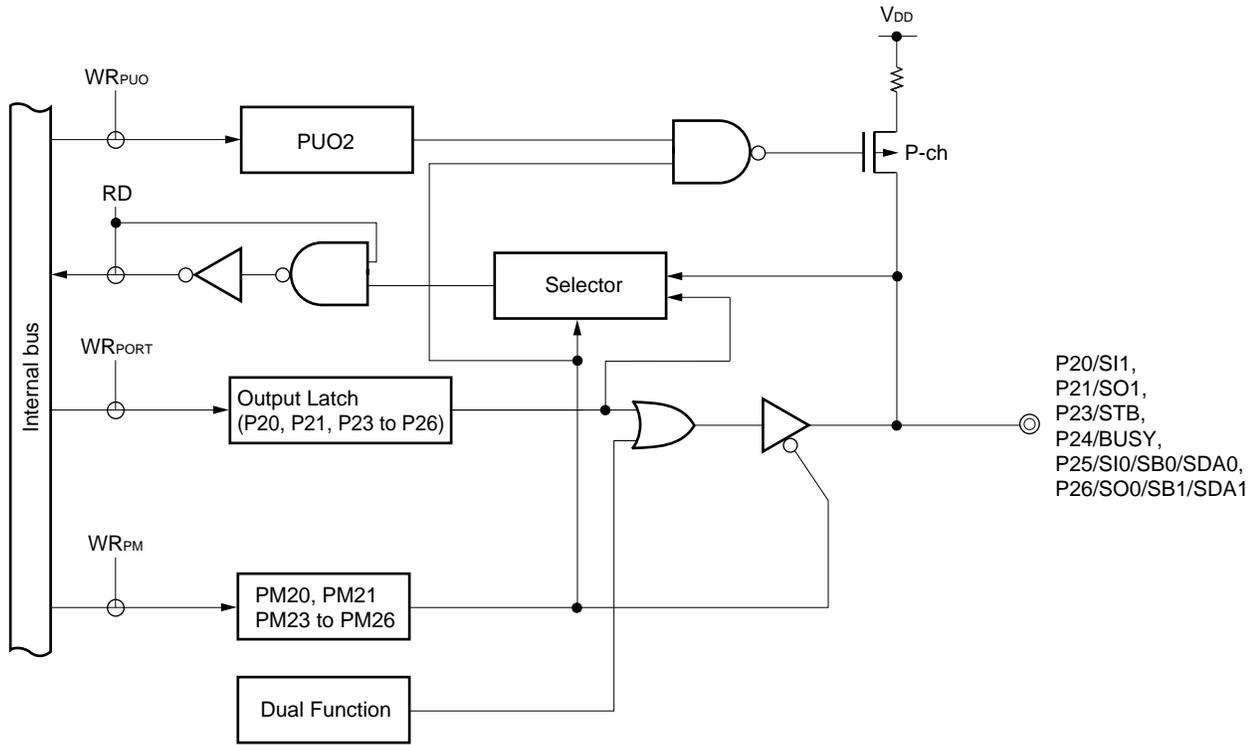
Figure 6-5. Block Diagram of P20, P21, P23 to P26



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

★

Figure 6-6. Block Diagram of P22 and P27



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

P20/S11,
 P21/SO1,
 P23/STB,
 P24/BUSY,
 P25/SI0/SB0/SDA0,
 P26/SO0/SB1/SDA1

6.2.4 Port 2 (μ PD78078Y Subseries)

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

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

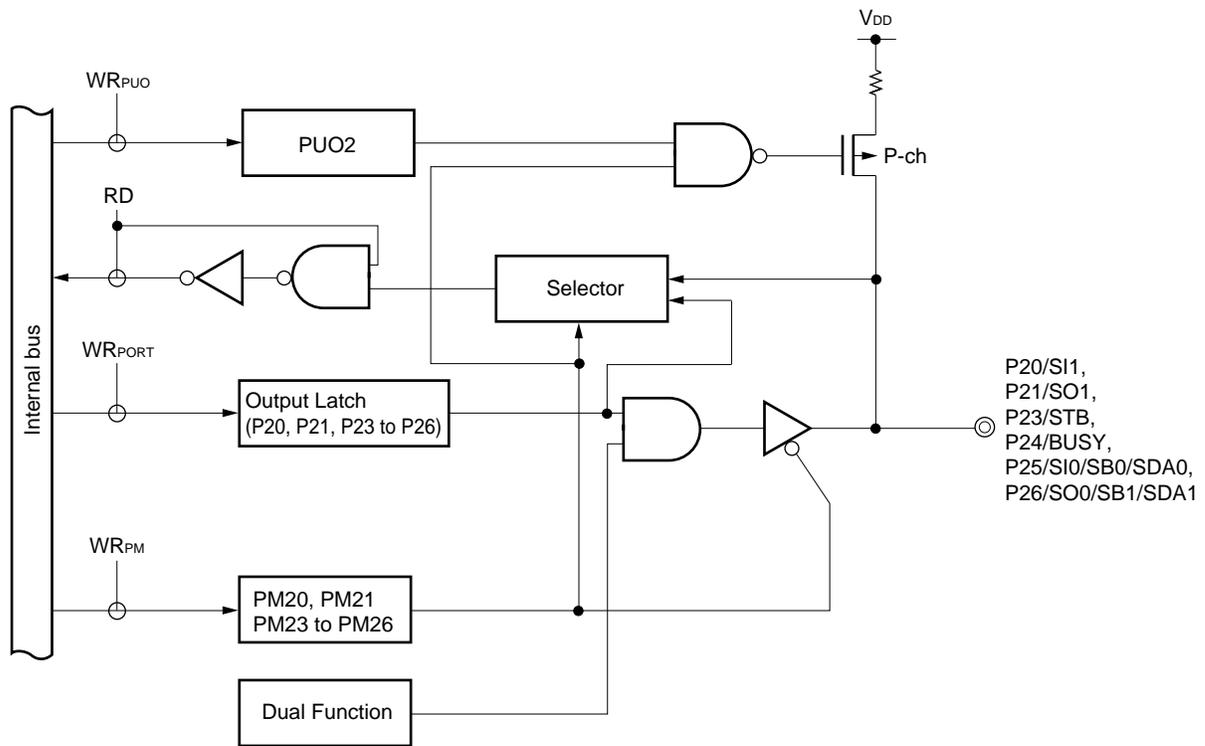
$\overline{\text{RESET}}$ input sets port 2 to input mode.

Figures 6-7 and 6-8 show a block diagram of port 2.

Caution When used as a serial interface, set the input/output and output latch according to its functions. For the setting method, refer to Figure 18-4. Serial Operating Mode Register 0 Format and Figure 19-3. Serial Operating Mode Register 1 Format.

★

Figure 6-7. Block Diagram of P20, P21, P23 to P26



PUO : Pull-up resistor option register

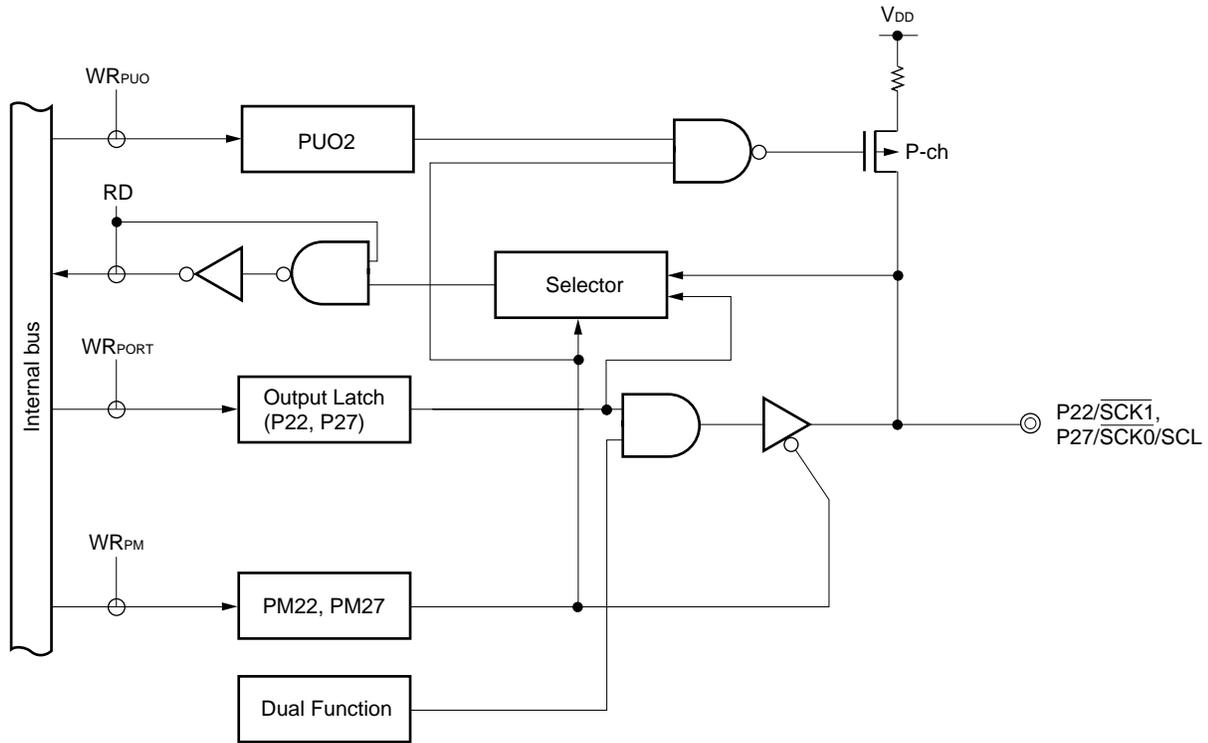
PM : Port mode register

RD : Port 2 read signal

WR : Port 2 write signal

★

Figure 6-8. Block Diagram of P22 and P27



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

6.2.5 Port 3

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

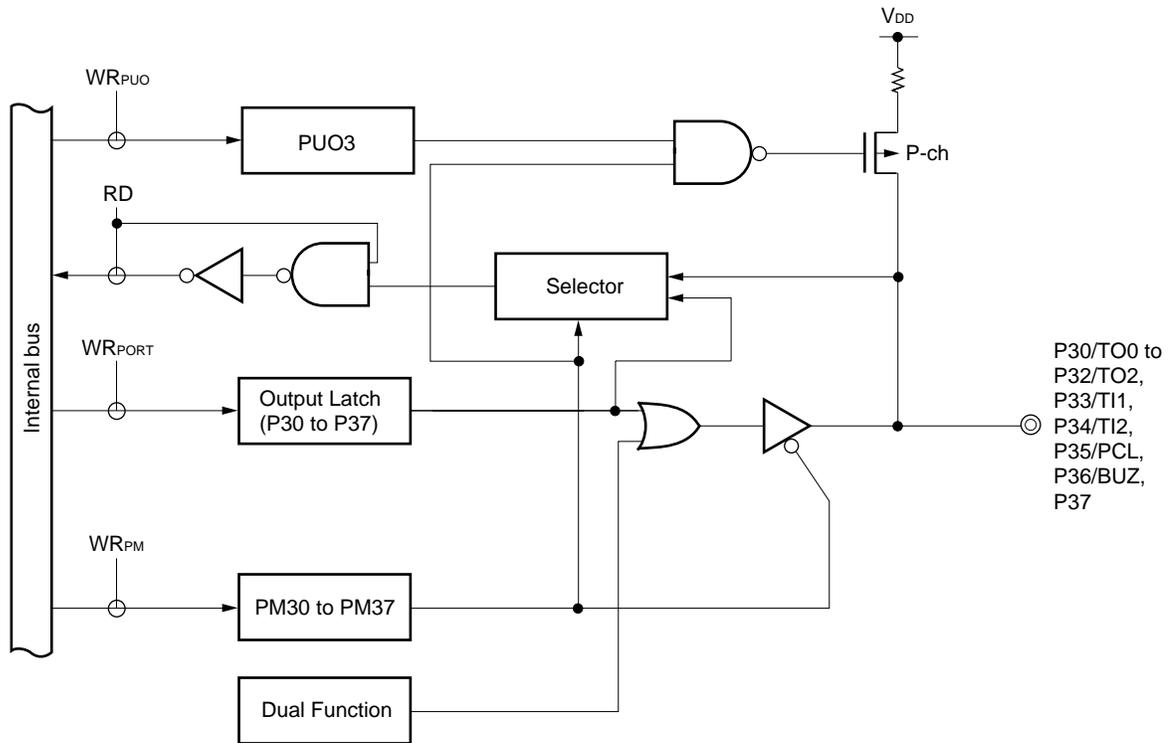
Dual-functions include timer input/output, clock output and buzzer output.

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

Figure 6-9 shows a block diagram of port 3.

★

Figure 6-9. Block Diagram of P30 to P37



- PUO : Pull-up resistor option register
- PM : Port mode register
- RD : Port 3 read signal
- WR : Port 3 write signal

6.2.6 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 memory expansion mode register (MM). When P40 to P47 pins are used as input ports, an on-chip pull-up resistor can be connected to them in 8-bit units with a pull-up resistor option register L (PUOL).

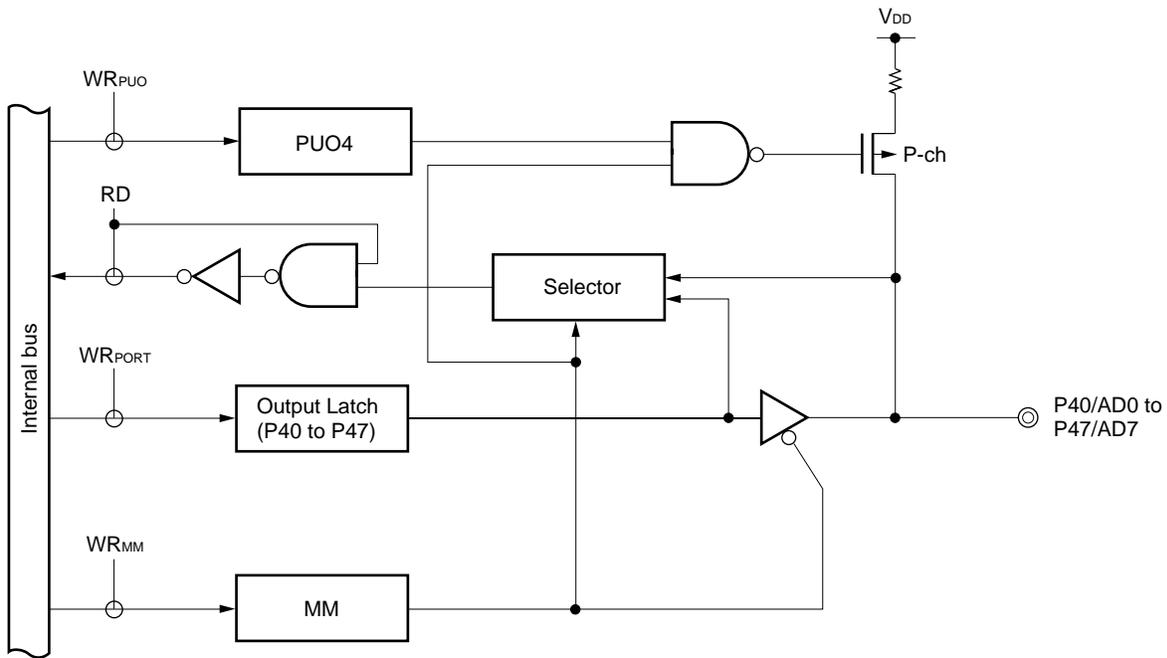
The test input flag (KRIF) can be set to 1 by detecting falling edges.

Dual-functions include address/data bus function in external memory expansion mode.

$\overline{\text{RESET}}$ input sets port 4 to input mode.

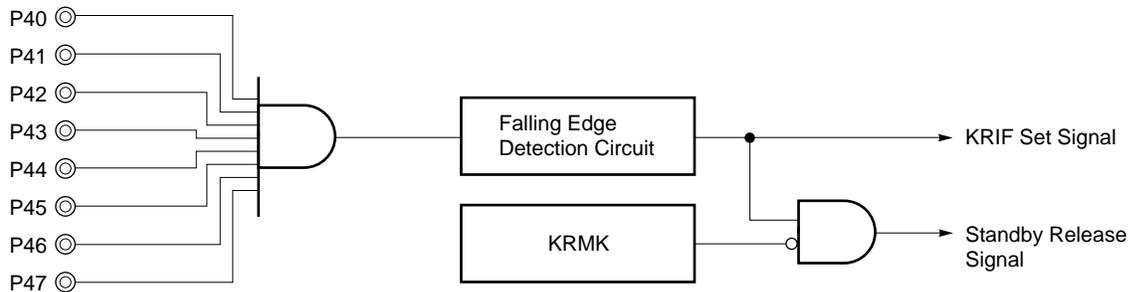
Figures 6-10 and 6-11 show a block diagram of port 4 and block diagram of falling edge detection circuit, respectively.

Figure 6-10. Block Diagram of P40 to P47



- PUO : Pull-up resistor option register
- MM : Memory expansion mode register
- RD : Port 4 read signal
- WR : Port 4 write signal

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



6.2.7 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). When P50 to P57 pins are used as input ports, an on-chip pull-up resistor can be connected to them in 8-bit units with a pull-up resistor option register L (PUOL).

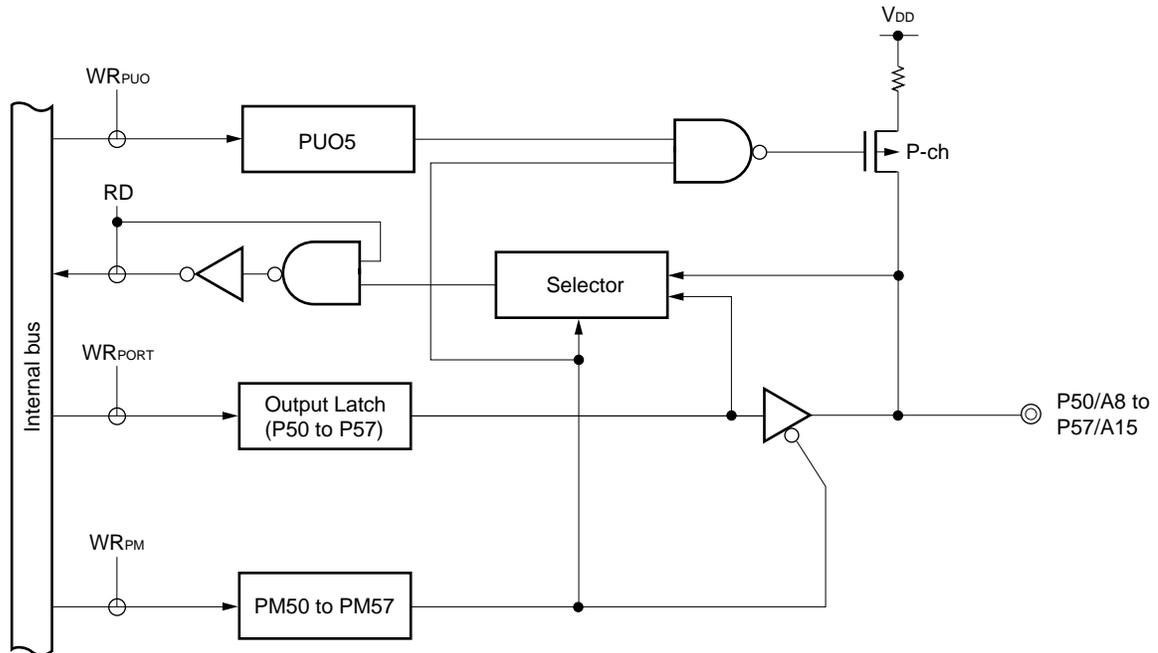
Port 5 can drive LEDs directly.

Dual-functions include address bus function in external memory expansion mode.

$\overline{\text{RESET}}$ input sets port 5 to input mode.

Figure 6-12 shows a block diagram of port 5.

Figure 6-12. Block Diagram of P50 to P57



PUO : Pull-up resistor option register

PM : Port mode register

RD : Port 5 read signal

WR : Port 5 write signal

6.2.8 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).

This port has pull-up resistor options as shown below. However, the option specification method differs depending on the port pin and the device version.

Table 6-4. Pull-up Resistor Options for Port 6

	Higher 4 bits (P64 to P67)	Lower 4 bits (P60 to P63)
Mask ROM version	Internal pull-up resistors can be specified by setting PU06 in 4-bit units.	Internal mask-option pull-up resistors can be specified bitwise.
PROM version		No pull-up resistor options

PU06 : Bit 6 of the pull-up resistor option register L (PUOL)

Pins P60 to P63 can drive LEDs directly.

Dual-functions include the control signal output function in external memory expansion mode.

$\overline{\text{RESET}}$ input sets port 6 to input mode.

Figures 6-13 and 6-14 show block diagrams of port 6.

Cautions 1. When external wait is not used in external memory expansion mode, P66 can be used as an input/output port.

2. When the low level is input to the P60 to P63 pins, the leakage current that flows through each of these pins depends on the following conditions:

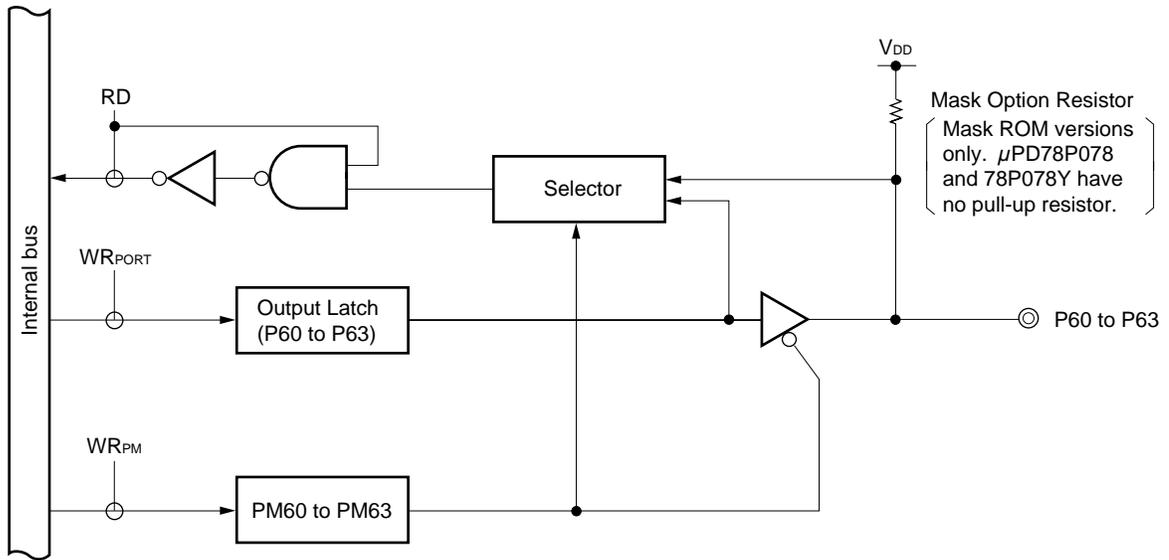
[Mask ROM version]

- When a pull-up resistor is connected,
–3 μA (max.)
- When a pull-up resistor is not connected,
–200 μA (max.) for 1.5 clock cycles during which a read instruction is executed to port 6 (P6) or port mode register 6 (PM6) (It is assumed that no wait cycles are inserted.)
–3 μA (max.) during periods other than the above

[PROM version]

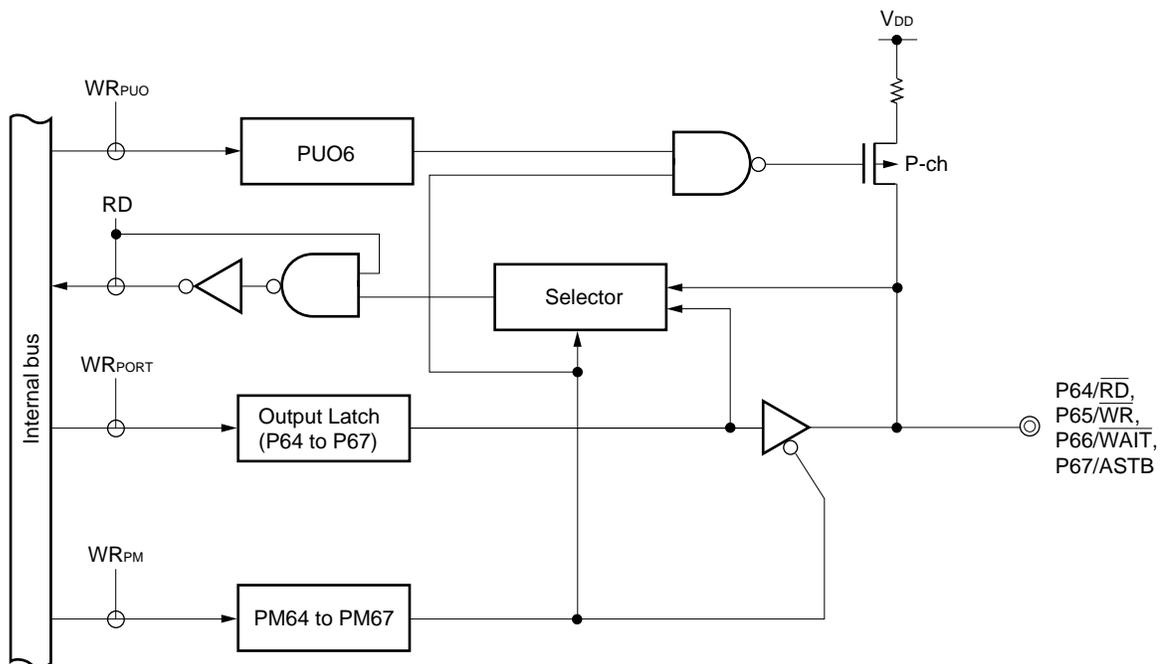
- When a pull-up resistor is not connected,
–200 μA (max.) for 1.5 clock cycles during which a read instruction is executed to port 6 (P6) or port mode register 6 (PM6) (It is assumed that no wait cycles are inserted.)
–3 μA (max.) during periods other than the above

Figure 6-13. Block Diagram of P60 to P63



PM : Port mode register
 RD : Port 6 read signal
 WR : Port 6 write signal

Figure 6-14. Block Diagram of P64 to P67



PUO : Pull-up resistor option register
 PM : Port mode register
 RD : Port 6 read signal
 WR : Port 6 write signal

6.2.9 Port 7

This is a 3-bit input/output port with output latches. Input mode/output mode can be specified in 1-bit units with a port mode register 7 (PM7). When pins P70 to P72 are used as input port pins, an on-chip pull-up resistor can be connected in 3-bit units with a pull-up resistor option register L (PUOL).

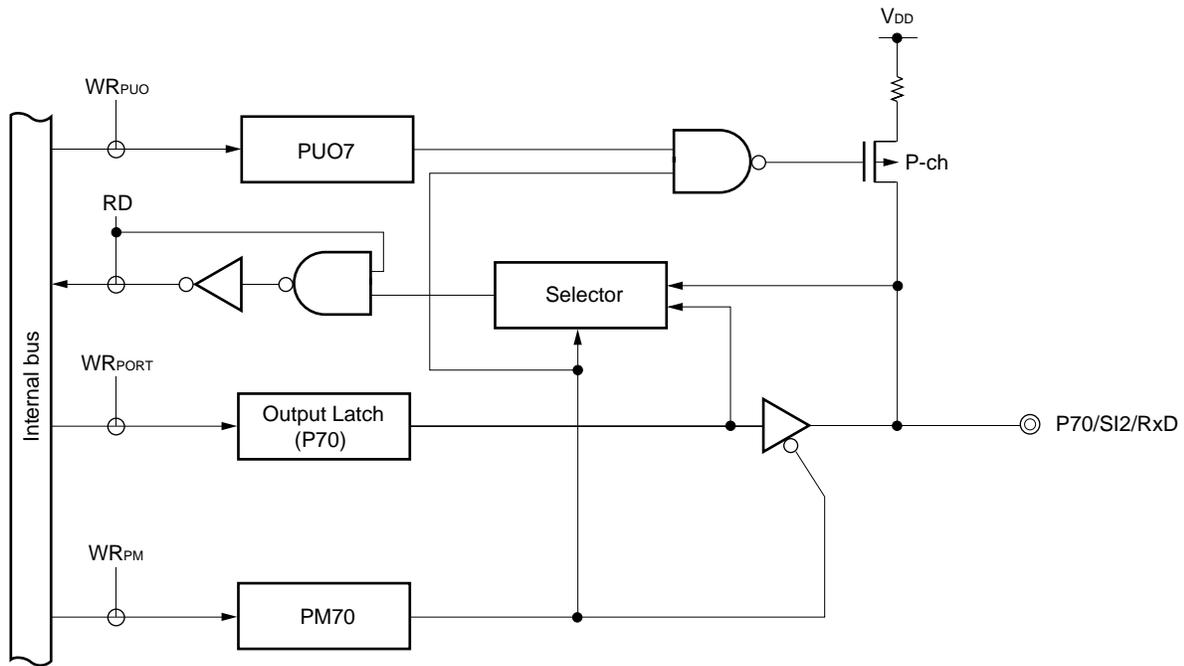
Dual-functions include serial interface channel 2 data input/output and clock input/output.

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

Port 7 block diagrams are shown in Figures 6-15 and 6-16.

Caution When used as a serial interface, set the input/output and output latch according to its functions. For the setting method, refer to Table 20-2. Serial Interface Channel 2 Operating Mode Setting.

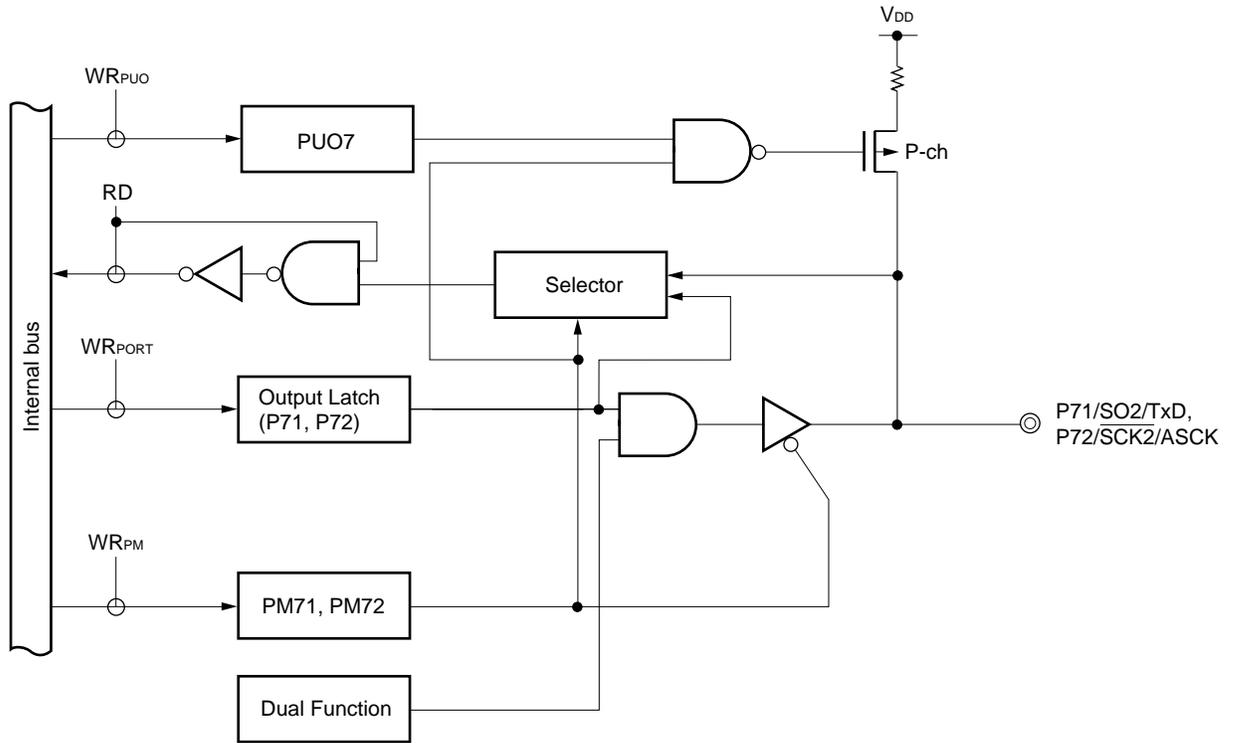
Figure 6-15. Block Diagram of P70



- PUO : Pull-up resistor option register
- PM : Port mode register
- RD : Port 7 read signal
- WR : Port 7 write signal

★

Figure 6-16. Block Diagram of P71 and P72



- PUO : Pull-up resistor option register
- PM : Port mode register
- RD : Port 7 read signal
- WR : Port 7 write signal

6.2.10 Port 8

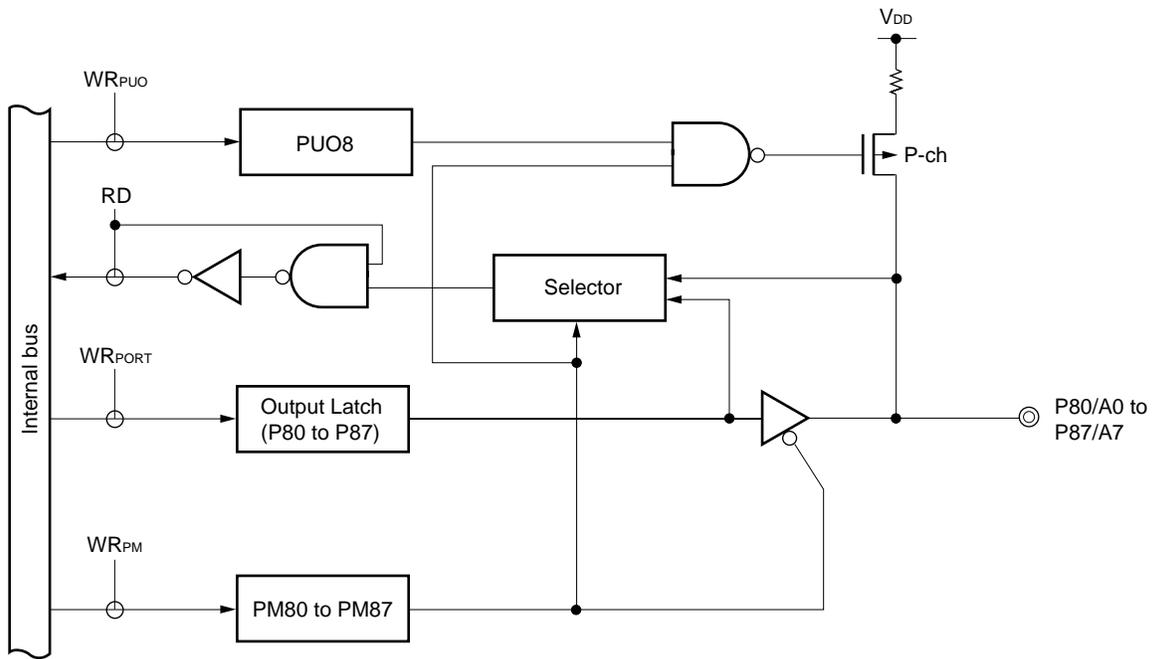
Port 8 is an 8-bit input/output port with output latch. P80 to P87 pins can specify the input mode/output mode in 1-bit units with the port mode register 8 (PM8). When pins P80 to P87 are used as input ports, an on-chip pull-up resistor can be connected to them in 8-bit units with a pull-up resistor option register H (PUOH).

Dual-functions include the address bus function in external memory expansion mode.

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

Figure 6-17 shows block diagram of port 8.

Figure 6-17. Block Diagram of P80 to P87



PUO : Pull-up resistor option register

PM : Port mode register

RD : Port 8 read signal

WR : Port 8 write signal

6.2.11 Port 9

Port 9 is an 7-bit input/output port with output latch. P90 to P96 pins can specify the input mode/output mode in 1-bit units with the port mode register 9 (PM9).

This port has pull-up resistor options as shown below. However, the option specification method differs depending on the port pin and the device version.

Table 6-5. Pull-up Resistor Options for Port 9

	Higher 3 bits (P94 to P97)	Lower 4 bits (P90 to P93)
Mask ROM version	Internal pull-up resistors can be specified by setting PU06 in 3-bit units.	Internal mask-option pull-up resistors can be specified bitwise.
PROM version		No pull-up resistor options

PU09 : Bit 1 of the pull-up resistor option register H (PUOH)

Pins P90 to P93 can drive LEDs directly.

Dual-functions include the control signal output function in external memory expansion mode.

$\overline{\text{RESET}}$ input sets port 6 to input mode.

Figures 6-18 and 6-19 show block diagrams of port 9.

Caution When the low level is input to the P90 to P93 pins, the leakage current that flows through each of these pins depends on the following conditions:

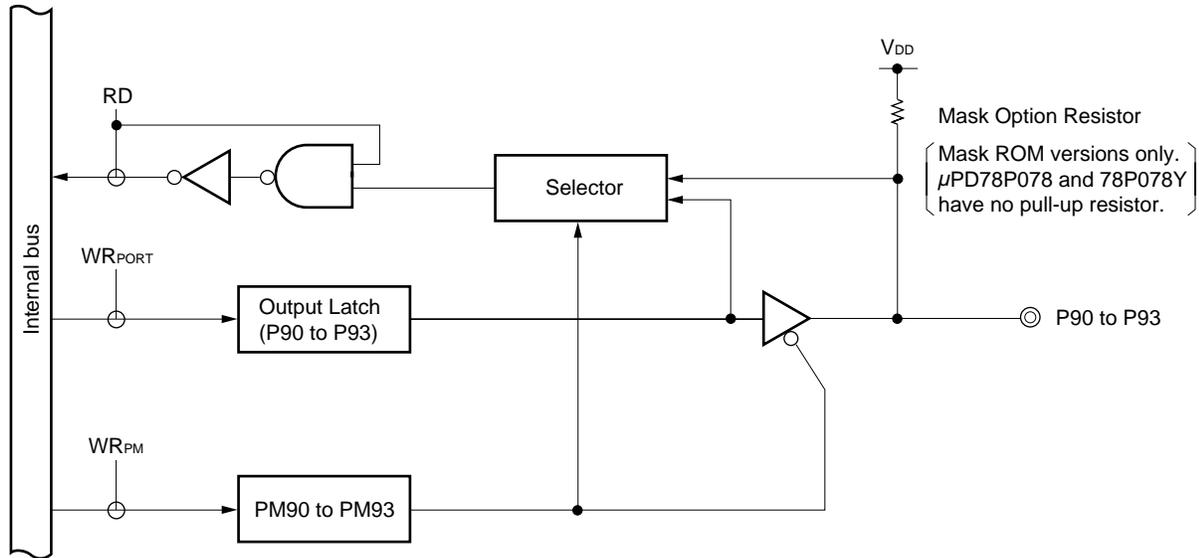
[Mask ROM version]

- When a pull-up resistor is connected,
-3 μA (max.)
- When a pull-up resistor is not connected,
-200 μA (max.) for 1.5 clock cycles during which a read instruction is executed to port 9 (P9) or port mode register 9 (PM9) (It is assumed that no wait cycles are inserted.)
-3 μA (max.) during periods other than the above

[PROM version]

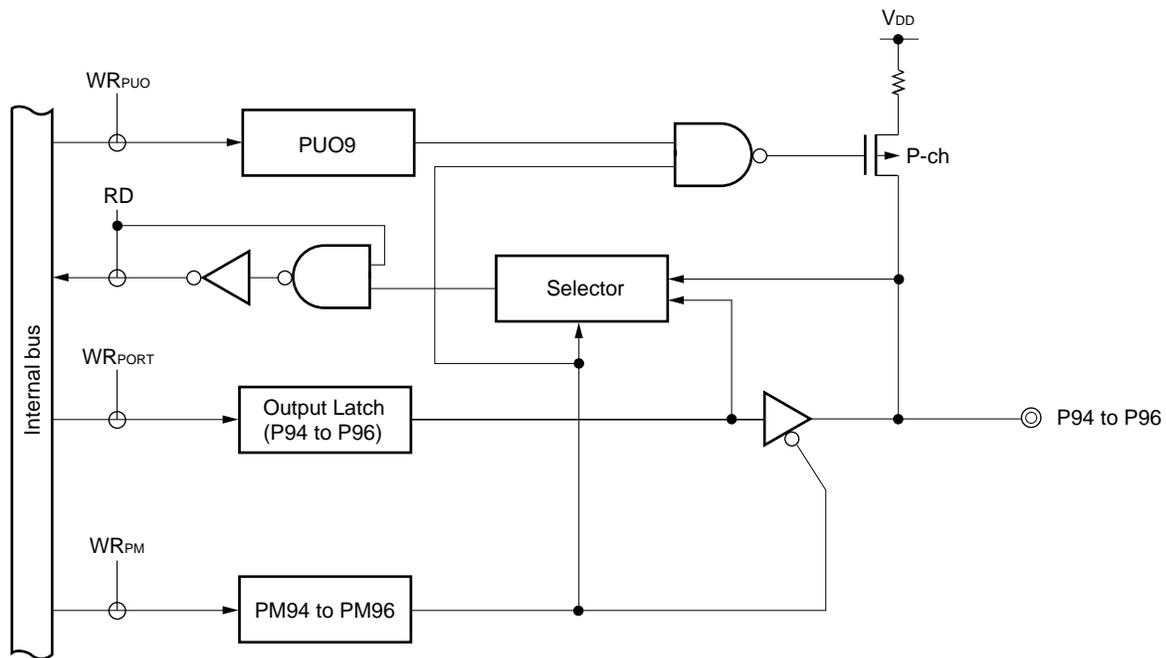
- When a pull-up resistor is not connected,
-200 μA (max.) for 1.5 clock cycles during which a read instruction is executed to port 9 (P9) or port mode register 9 (PM9) (It is assumed that no wait cycles are inserted.)
-3 μA (max.) during periods other than the above

Figure 6-18. Block Diagram of P90 to P93



PM : Port mode register
 RD : Port 9 read signal
 WR : Port 9 write signal

Figure 6-19. Block Diagram of P94 to P96



P_{UO} : Pull-up resistor option register
 PM : Port mode register
 RD : Port 9 read signal
 WR : Port 9 write signal

6.2.12 Port 10

Port 10 is a 4-bit input/output port with output latch. P100 to P103 pins can specify the input mode/output mode in 1-bit units with the port mode register 10 (PM10). When pins P100 to P103 are used as input ports, an on-chip pull-up resistor can be connected to them in 4-bit units with a pull-up resistor option register H (PUOH).

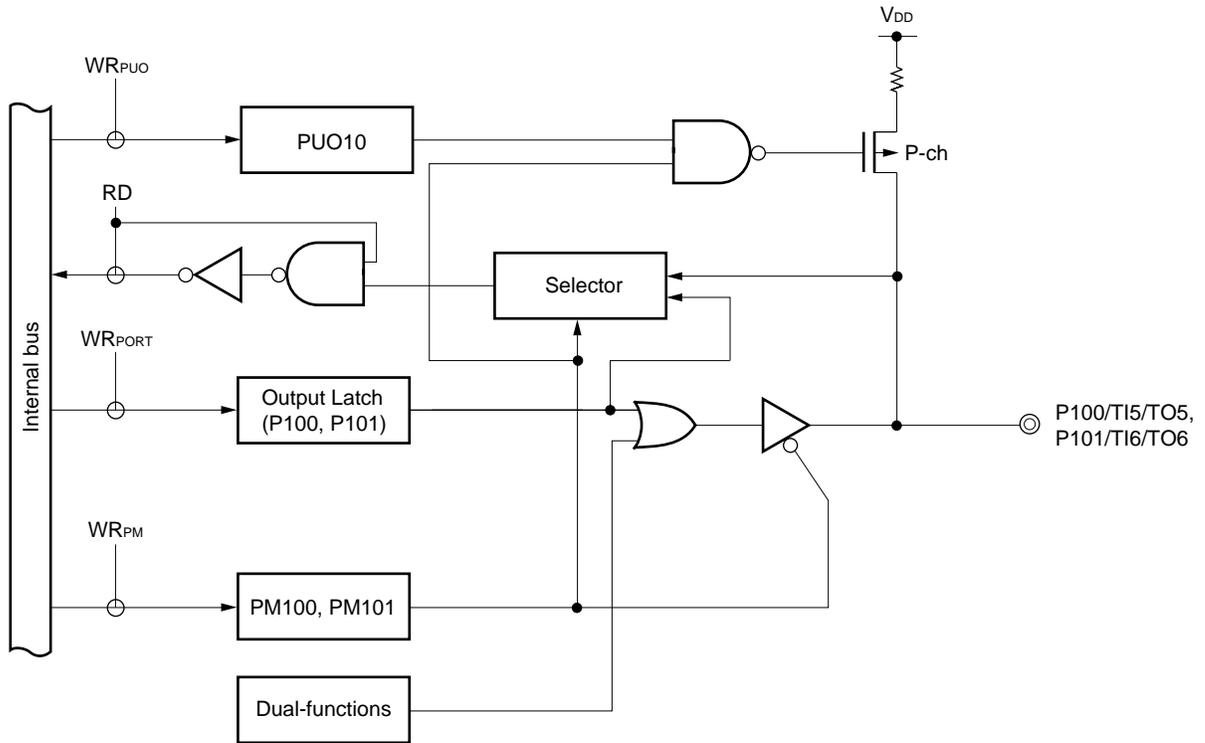
Dual-functions include the timer input/output.

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

Figures 6-20 and 6-21 show block diagrams of port 10.

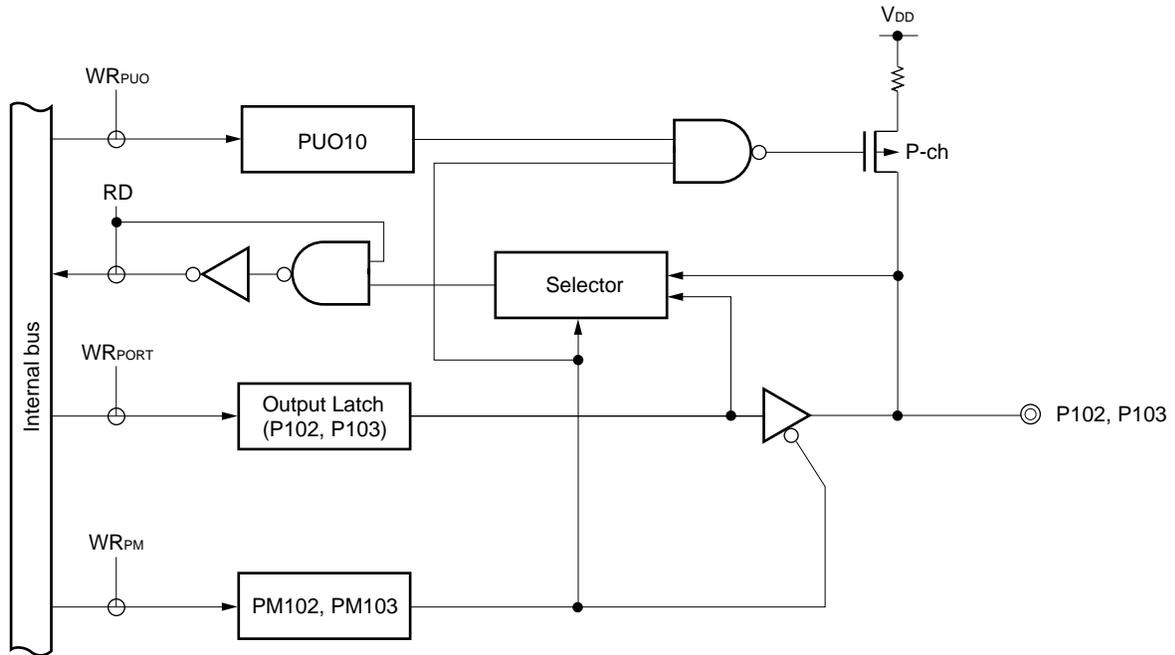
★

Figure 6-20. Block Diagram of P100 and P101



- PUO : Pull-up resistor option register
- PM : Port mode register
- RD : Port 10 read signal
- WR : Port 10 write signal

Figure 6-21. Block Diagram of P102 and P103



- PUO : Pull-up resistor option register
- PM : Port mode register
- RD : Port 6 read signal
- WR : Port 6 write signal

6.2.13 Port 12

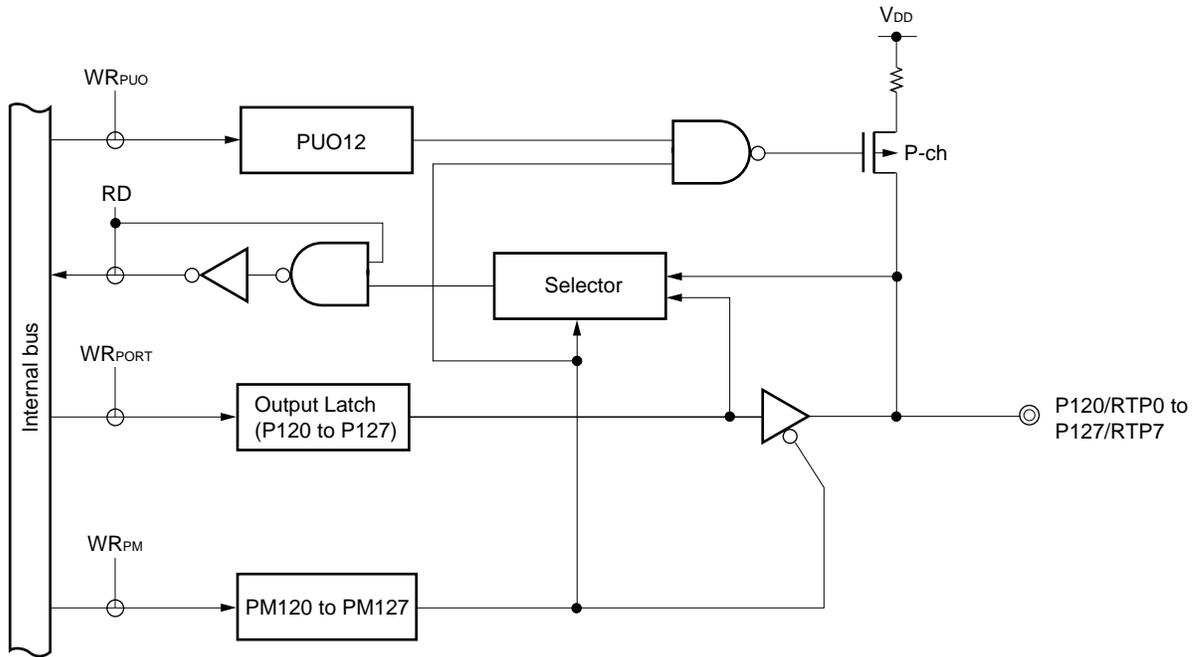
This is an 8-bit input/output port with output latches. Input mode/output mode can be specified in 1-bit units with the port mode register 12 (PM12). When pins P120 to P127 are used as input port pins, an on-chip pull-up resistor can be connected in 8-bit units with a pull-up resistor option register H (PUOH).

These pins are dual function pin and serve as real-time outputs.

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

The port 12 block diagram is shown in Figure 6-22.

Figure 6-22. Block Diagram of P120 to P127



- PUO : Pull-up resistor option register
- PM : Port mode register
- RD : Port 12 read signal
- WR : Port 12 write signal

6.2.14 Port 13

This is a 2-bit input/output port with output latches. Input mode/output mode can be specified in 1-bit units with the port mode register 13 (PM13). When pins P130 and P131 are used as input port pins, an on-chip pull-up resistor can be connected in 2-bit units with a pull-up resistor option register H (PUOH).

These pins are dual function pin and serve as D/A converter analog outputs.

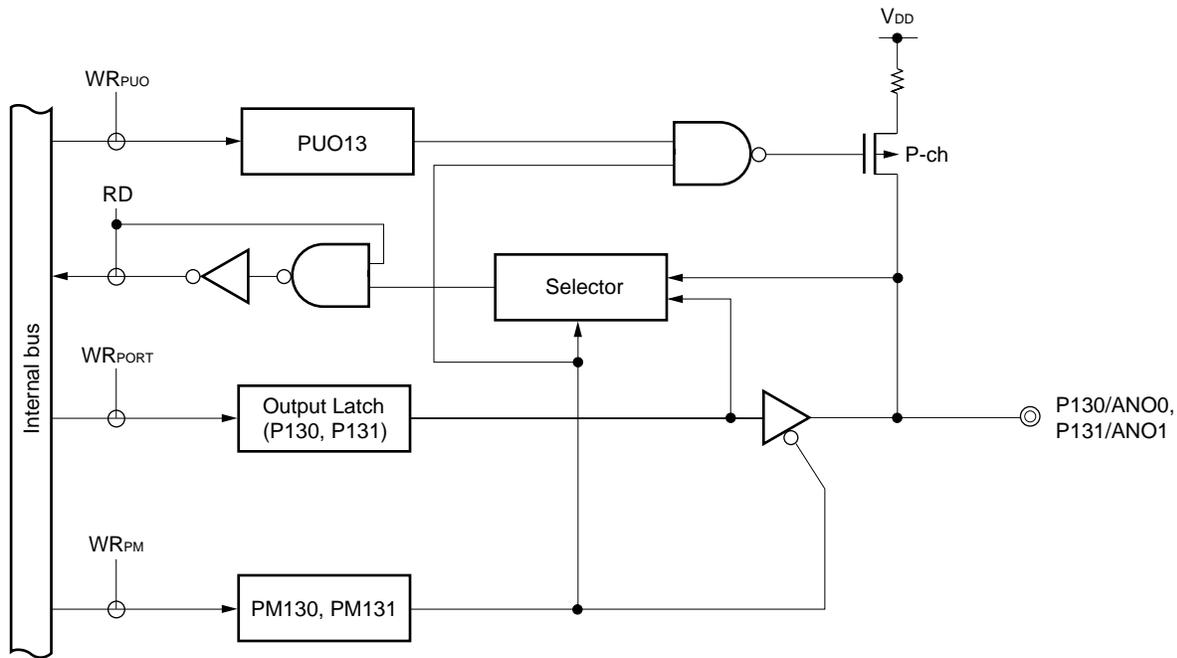
$\overline{\text{RESET}}$ input sets the input mode.

The port 13 block diagram is shown in Figure 6-23.

Caution When only either one of the D/A converter channels is used with $\text{AV}_{\text{REF1}} < \text{V}_{\text{DD}}$, the other pins that are not used as analog outputs must be set as follows:

- Set PM13x bit of the port mode register 13 (PM13) to 1 (input mode) and connect the pin to V_{SS} .
- Set PM13x bit of the port mode register 13 (PM13) to 0 (output mode) and the output latch to 0, to output low level from the pin.

Figure 6-23. Block Diagram of P130 and P131



PUO : Pull-up resistor option register

PM : Port mode register

RD : Port 13 read signal

WR : Port 13 write signal

6.3 Port Function Control Registers

The following four types of registers control the ports.

- Port mode registers (PM0 to PM3, PM5 to PM10, PM12, PM13)
- Pull-up resistor option register (PUOH, PUOL)
- Memory expansion mode register (MM)
- Key return mode register (KRM)

(1) Port mode registers (PM0 to PM3, PM5 to PM10, PM12, PM13)

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

PM0 to PM3, PM5 to PM10, PM12, and PM13 are independently set with a 1-bit or 8-bit memory manipulation instruction

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

When port pins are used as alternate-function pins, set the port mode register and output latch according to Table 6-6.

Cautions 1. Pins P00 and P07 are input-only pins.

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

3. The memory expansion mode register specifies P40 to P47 as input/output pins.

Table 6-6. Port Mode Register and Output Latch Settings when Using Alternate Function

Pin Name	Alternate Function		PMxx	Pxx	Pin Name	Alternate Function		PMxx	Pxx
	Name	Input/Output				Name	Input/Output		
P00	INTP0	Input	1 (Fixed)	None	P50 to P57	A8 to A15	Output	x ^{Note 2}	
	TI00	Input	1 (Fixed)	None	P64	\overline{RD}	Output	x ^{Note 2}	
P01	INTP1	Input	1	x	P65	\overline{WR}	Output	x ^{Note 2}	
	TI01	Input	1	x	P66	\overline{WAIT}	Input	x ^{Note 2}	
P02 to P06	INTP2 to INTP6	Input	1	x	P67	ASTB	Output	x ^{Note 2}	
P07 ^{Note 1}	XT1	Input	1 (Fixed)	None	P80 to P87	A0 to A7	Output	x ^{Note 3}	
P10 to P17 ^{Note 1}	ANI0 to ANI7	Input	1	x	P100	TI5	Input	1	x
P30 to P32	TO0 to TO2	Output	0	0		TO5	Output	0	0
P33, P34	TI1, TI2	Input	1	x	P101	TI6	Input	1	x
P35	PCL	Output	0	0		TO6	Output	0	0
P36	BUZ	Output	0	0	P120 to P127	RTP0 to RTP7	Output	0	Desired value
P40 to P47	AD0 to AD7	Input/Output	x ^{Note 2}		P130,P131 ^{Note 1}	ANO0, ANO1	Output	1	x

- Notes**
1. If these ports are read out when these pins are used in the alternative function mode, undefined values are read.
 2. When the P40 to P47 pins P50 to P57 pins, and P64 to P67 pins are used for alternate functions, set the function with the memory extension mode register (MM).
 3. When the P80 to P87 pins are used for alternate functions, set the function with the external bus type selection register (EBTS).

- Cautions**
1. When not using external wait in the external memory extension mode, the P66 pin can be used as an I/O port.
 2. When port 2 and port 7 are used for serial interface, the I/O latch or output latch must be set according to its function. For the setting methods, see Figure 17-4. Serial Operation Mode Register 0 Format, Figure 18-4. Serial Operation Mode Register 0 Format, Figure 19-3. Serial Operation Mode Register 1 Format, and Table 20-2. Operating Mode Setting for Serial Interface Channel 2.

Remarks

- x : Don't care
- PMxx : Port mode register
- Pxx : Port output latch

Figure 6-24. 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	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FF25H	FFH	R/W
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60	FF26H	FFH	R/W
PM7	1	1	1	1	1	PM72	PM71	PM70	FF27H	FFH	R/W
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80	FF28H	FFH	R/W
PM9	1	PM96	PM95	PM94	PM93	PM92	PM91	PM90	FF29H	FFH	R/W
PM10	1	1	1	1	PM103	PM102	PM101	PM100	FF2AH	FFH	R/W
PM12	PM127	PM126	PM125	PM124	PM123	PM122	PM121	PM120	FF2CH	FFH	R/W
PM13	1	1	1	1	1	1	PM131	PM130	FF2DH	FFH	R/W

PMmn	Pmn Pin Input/Output Mode Selection (m = 0 to 3, 5 to 10, 12, 13 : n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

(2) Pull-up resistor option register (PUOH, PUOL)

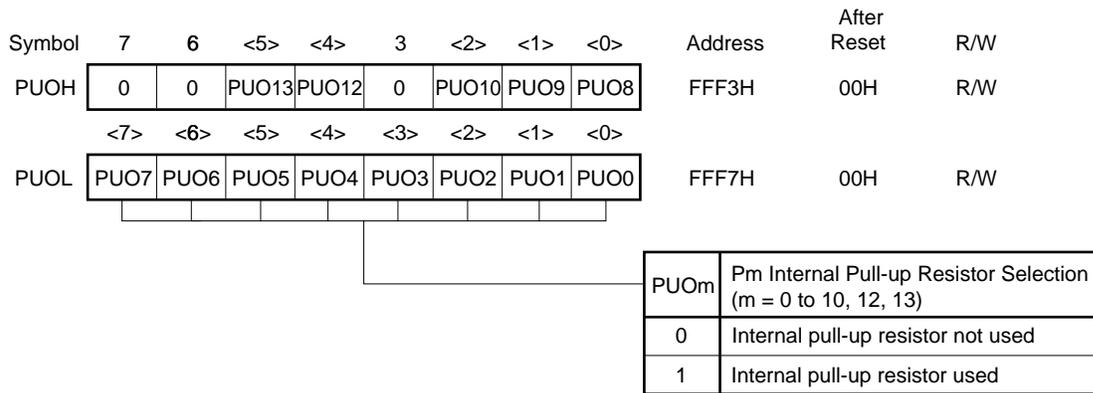
This register is used to set whether to use an internal pull-up resistor at each port or not. A pull-up resistor is internally used at bits which are set to the input mode at a port where on-chip pull-up resistor use has been specified with PUOH, PUOL. No on-chip pull-up resistors can be used to the bits set to the output mode or to the bits used as an analog input pin, irrespective of PUOH or PUOL setting.

PUOH and PUOL are set with a 1-bit or 8-bit memory manipulation instruction.

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

- Cautions**
1. P00 and P07 pins do not incorporate a pull-up resistor.
 2. When ports 1, 4, 5, and P64 to P67 pins, or port 8 pins are used as dual-function pins, an on-chip pull-up resistor cannot be used even if 1 is set in PUOm bit of PUOH and PUOL (m = 1, 4 to 6, or 8).
 3. Pins P60 to P63 and P90 to P93 can be incorporated with pull-up resistor by mask option only for mask ROM version.

Figure 6-25. Pull-Up Resistor Option Register Format



Caution Bits 3, 6, and 7 of PUOH should be set to 0.

(3) Memory expansion mode register (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.

$\overline{\text{RESET}}$ input sets this register to 10H.

Figure 6-26. Memory Expansion Mode Register Format

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

MM2	MM1	MM0	Single-chip/Memory Expansion Mode Selection		P40 to P47, P50 to P57, P64 to P67, P80 to P87 Pin State					
					P40 to P47		P80 to P87	P50 to P53	P54, P55	P56, P57
0	0	0	Single-chip mode		Port mode	Input	Port mode			
0	0	1								
0	1	1	Memory expansion mode	256-byte mode	AD0 to AD7	A0 to A7 ^{Note 1}	Port mode			P64 = $\overline{\text{RD}}$ P65 = $\overline{\text{WR}}$ P66 = $\overline{\text{WAIT}}$ P67 = $\overline{\text{ASTB}}$
1	0	0		4-Kbyte mode			A8 to A11	Port mode		
1	0	1		16-Kbyte mode				A12, A13	Port mode	
1	1	1		Full ^{Note 2} address mode			A14, A15		Port mode	
Other than above			Setting prohibited							

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

- Notes**
1. These pins can be used only in the separate bus mode. They enter the port mode when the device is switched to the multiplexed bus mode.
 2. The full address mode allows external expansion for all areas of the 64-Kbyte address space, except the internal ROM, RAM, SFR, and use-prohibited areas.

- Remarks**
1. P60 to P63 pins enter the port mode in both the single-chip and memory expansion mode.
 2. Besides setting port 4 input/output, MM also sets the wait count and external expansion area.

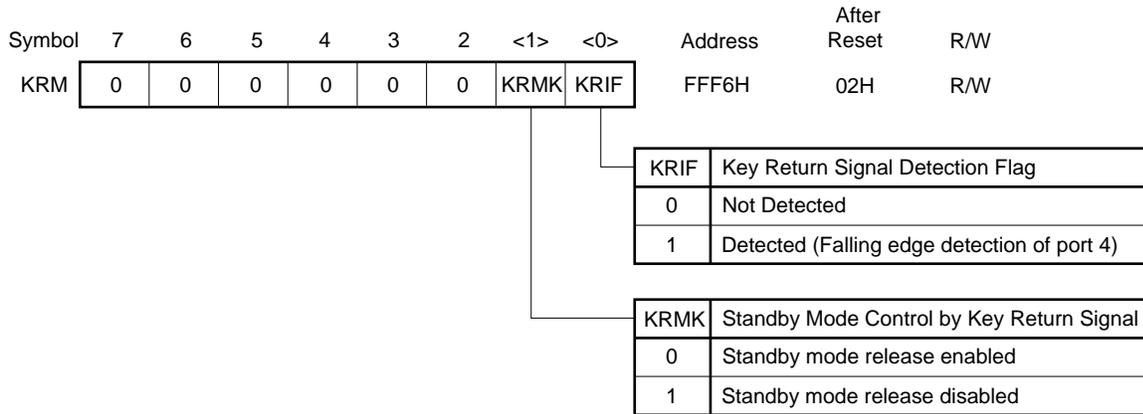
(4) Key return mode register (KRM)

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.

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

Figure 6-27. Key Return Mode Register Format



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

6.4 Port Function Operations

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

6.4.1 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.

6.4.2 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.

6.4.3 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.

6.5 Selection of Mask Option

The following mask option is provided in mask ROM version. The μ PD78P078 and 78P078Y have no mask option.

Table 6-7. Comparison between Mask ROM Version and the μ PD78P078 and 78P078Y

Pin Name	Mask ROM Version	μ PD78P078 and 78P078Y
Mask option for pins P60 to P63 and P90 to P93	Bitwise-selectable on-chip pull-up resistors	No on-chip pull-up resistor

CHAPTER 7 CLOCK GENERATOR

7.1 Clock Generator Functions

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

(1) Main system clock oscillator

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

(2) Subsystem clock oscillator

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

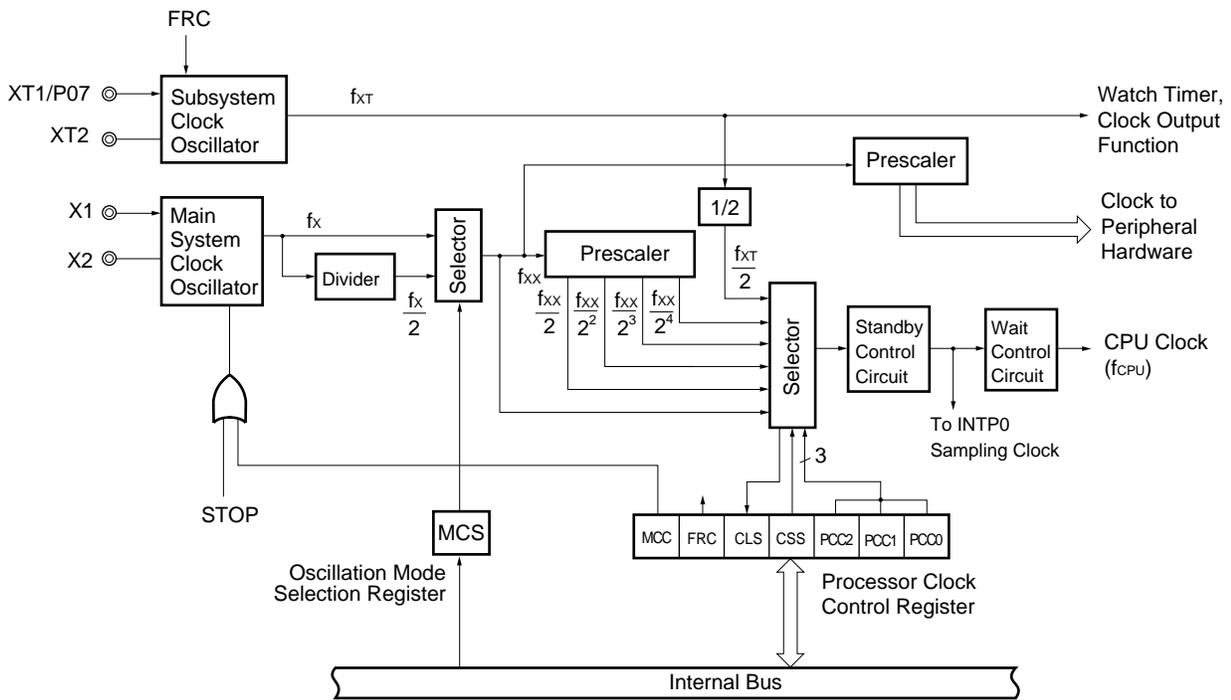
7.2 Clock Generator Configuration

The clock generator consists of the following hardware.

Table 7-1. Clock Generator Configuration

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

Figure 7-1. Block Diagram of Clock Generator



7.3 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 selects a CPU clock and the division ratio, determines whether to make the main system clock oscillator operate or stop, and enables or disables the subsystem clock oscillator internal feedback resistor. The PCC is set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 7-2. Subsystem Clock Feedback Resistor

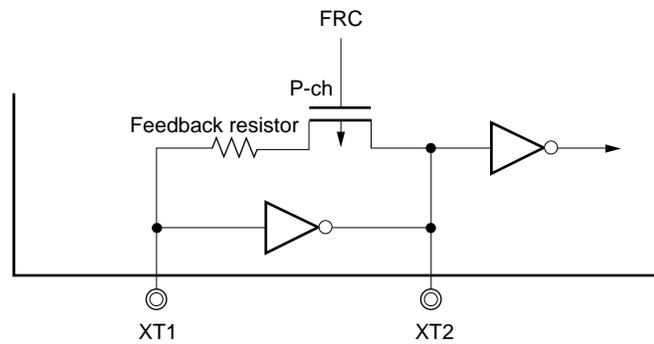


Figure 7-3. Processor Clock Control Register Format

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

R/W	CSS	PCC2	PCC1	PCC0	CPU Clock (f_{CPU}) Selection		
					MCS = 1		MCS = 0
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	$f_{xx}/2^4$	$f_x/2^4$	$f_x/2^5$	
1	0	0	0	$f_{xt}/2$			
	0	0	1				
	0	1	0				
	0	1	1				
	1	0	0				
Other than above				Setting prohibited			

R	CLS	CPU Clock Status
	0	Main system clock
	1	Subsystem clock

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

R/W	MCC	Main System Clock Oscillation Control ^{Note 2}
	0	Oscillation possible
	1	Oscillation stopped

- Notes**
1. Bit 5 is a read-only bit.
 2. When the CPU is operating on the subsystem clock, MCC should be used to stop the main system clock oscillation. A STOP instruction should not be used.

Caution Bit 3 must be set to 0.

- Remarks**
1. f_{xx} : Main system clock frequency (f_x or $f_x/2$)
 2. f_x : Main system clock oscillation frequency
 3. f_{xt} : Subsystem clock oscillation frequency
 4. MCS : Bit 0 of oscillation mode selection register (OSMS)

- ★ The fastest instruction of the μ PD78078 and 78078Y Subseries can be executed in two clocks of the CPU clock. The relationship between the CPU clock (f_{CPU}) and the minimum instruction execution time is shown in Table 7-2.

Table 7-2. Relationship between CPU Clock and Minimum Instruction Execution Time

CPU Clock (f_{CPU})	Minimum Instruction Execution Time: $2/f_{CPU}$
f_X	0.4 μs
$f_X/2$	0.8 μs
$f_X/2^2$	1.6 μs
$f_X/2^3$	3.2 μs
$f_X/2^4$	6.4 μs
$f_X/2^5$	12.8 μs
$f_{XT}/2$	122 μs

$f_X = 5.0 \text{ MHz}$, $f_{XT} = 32.768 \text{ kHz}$

f_X : Main system clock oscillation frequency

f_{XT} : Subsystem clock oscillation frequency

(2) Oscillation mode selection register (OSMS)

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

$\overline{\text{RESET}}$ input sets OSMS to 00H.

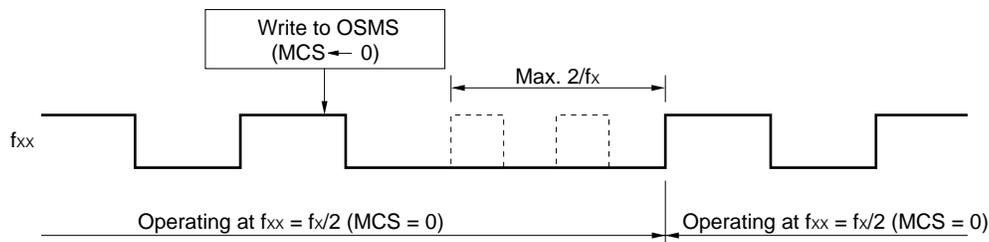
Figure 7-4. Oscillation Mode Selection Register Format

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

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

- Cautions**
- As shown in Figure 7-5 below, writing data (including same data as previous) to OSMS cause delay of main 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 main system clock, a temporary error occurs in the count clock cycle of timer, etc. In addition, because the oscillation mode is changed by this register, the clocks for peripheral hardware as well as that for the CPU are switched. Therefore, writing to OSMS should be performed only immediately after reset signal release and before peripheral hardware operation starts.
 - When writing "1" to MCS, V_{DD} must be 2.7 V or higher before the write execution.

Figure 7-5. Main System Clock Waveform due to Writing to OSMS



Remarks f_{xx} : Main system clock frequency (f_x or $f_x/2$)
 f_x : Main system clock oscillation frequency

7.4 System Clock Oscillator

7.4.1 Main system clock oscillator

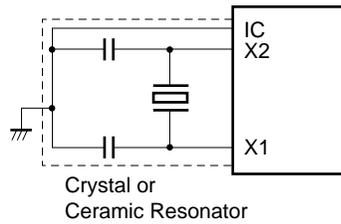
The main system clock oscillator oscillates with a crystal resonator or a ceramic resonator (standard: 5.0 MHz) connected to the X1 and X2 pins.

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

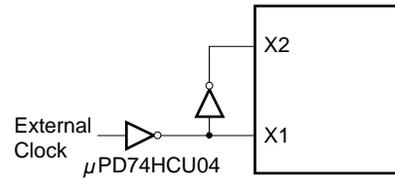
Figure 7-6 shows an external circuit of the main system clock oscillator.

Figure 7-6. External Circuit of Main System Clock Oscillator

(a) Crystal and ceramic oscillation



(b) External clock



Caution Do not execute the STOP instruction nor set MCC (bit 7 of the processor clock control register (PCC)) to 1 while an external clock is input. This is because the operation of main system clock is stopped and X2 pin is pulled-up to V_{DD} when a STOP instruction is executed and MCC is set to 1.

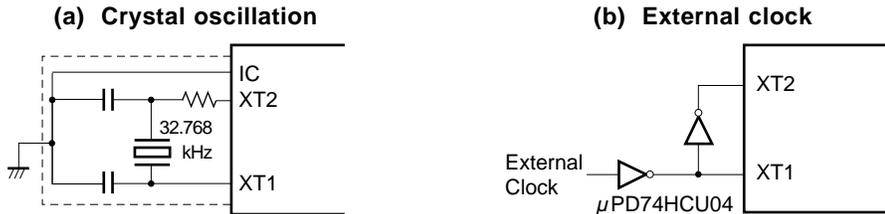
7.4.2 Subsystem clock oscillator

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

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

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

Figure 7-7. External Circuit of Subsystem Clock Oscillator



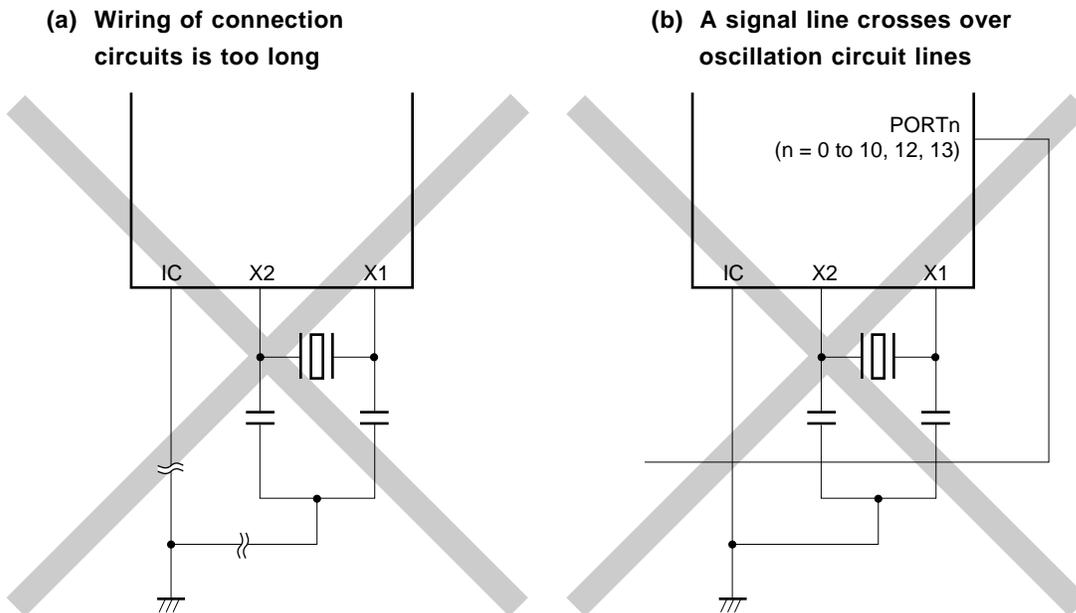
Cautions 1. When using a main system clock oscillator and a subsystem clock oscillator, carry out wiring in the broken-line area in Figures 7-6 and 7-7 as follows to prevent any effects from wiring capacities.

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

Take special note of the fact that the subsystem clock oscillator is a circuit with low-level amplification so that current consumption is maintained at low levels.

Figure 7-8 shows examples of oscillator having bad connection.

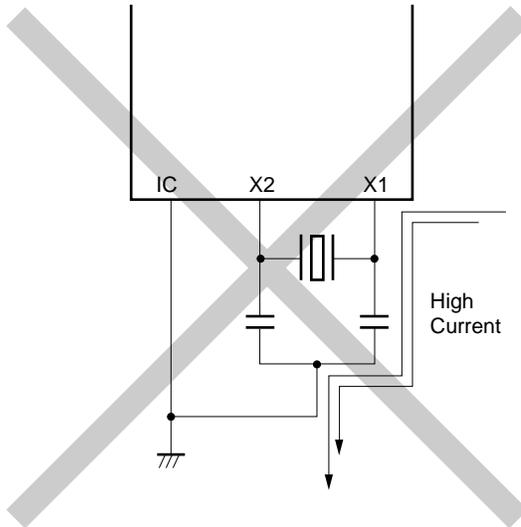
Figure 7-8. Examples of Oscillator with Bad Connection (1/2)



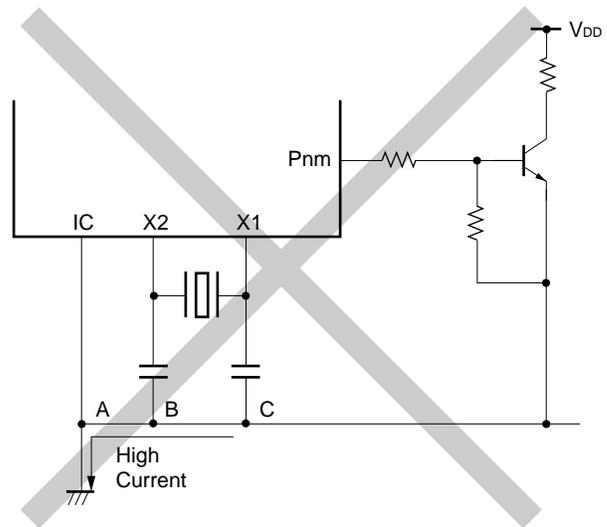
Remark When using a subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Further, insert resistors in series on the side of XT2.

Figure 7-8. Examples of Oscillator with Bad Connection (2/2)

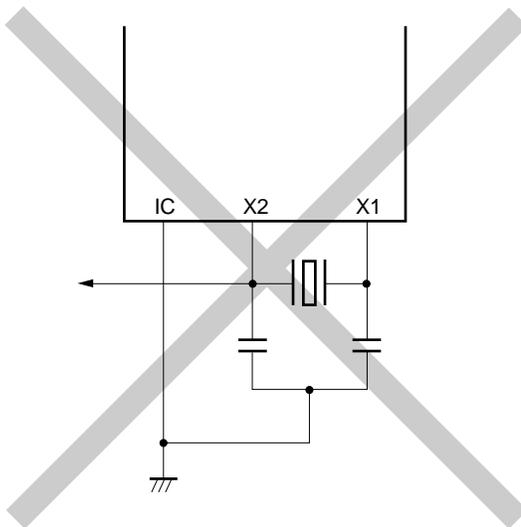
(c) Changing high current is too near a signal conductor



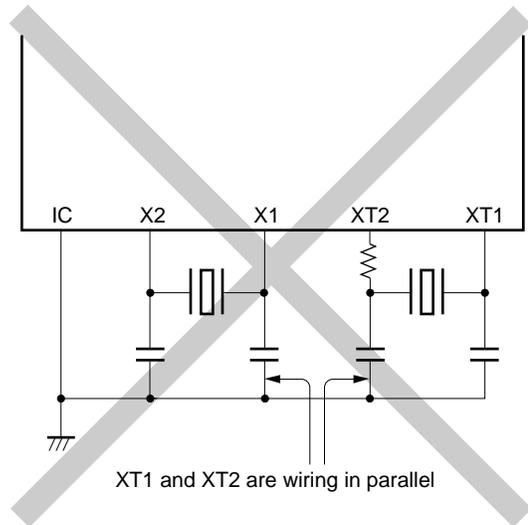
(d) Current flows through the grounding line of the oscillator (potential at points A, B, and C fluctuate)



(e) Signals are fetched



(f) Signal conductors of the main and sub-system clock are parallel and near each other



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

Cautions 2. In Figure 7-8 (f), XT2 and X1 are wired in parallel. Thus, the cross-talk noise of X1 may increase with XT2, resulting in malfunctioning. To prevent that from occurring, it is recommended to wire XT2 and X1 so that they are not in parallel, and to connect the IC pin between XT1 and X1 directly to V_{SS}.

7.4.3 Divider

The divider generates various clocks by dividing the main system clock oscillator output (f_{xx}).

7.4.4 When no subsystem clocks are used

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

XT1: Connect to V_{DD}

XT2: Open

In this state, however, some current may leak via the internal feedback resistor of the subsystem clock oscillator when the main system clock stops. To minimize leakage current, set bit 6 (FRC) of the processor clock control register (PCC) so that the above internal feedback resistor is not used. In this case also, connect the XT1 and XT2 pins as described above.

7.5 Clock Generator Operations

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

- Main system clock f_{XX}
- Subsystem clock f_{XT}
- 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 $\overline{\text{RESET}}$ signal, the lowest speed mode of the main system clock (12.8 μs when operated at 5.0 MHz) is selected (PCC = 04H, OSMS = 00H). Main system clock oscillation stops while the low level is applied to $\overline{\text{RESET}}$ pin.
- (b) With the main system clock selected, one of the five CPU clock stages (f_{XX} , $f_{XX}/2$, $f_{XX}/2^2$, $f_{XX}/2^3$ or $f_{XX}/2^4$) can be selected by setting the PCC and OSMS.
- (c) With the main system clock selected, two standby modes, the STOP and HALT modes, are available. In a system which is not using the subsystem clock, the current consumption in the STOP mode can be reduced further by setting bit 6 (FRC) of the PCC so as not to use the on-chip feedback resistor.
- (d) The PCC can be used to select the subsystem clock and to operate the system with low current consumption (122 μs when operated at 32.768 kHz).
- (e) With the subsystem clock selected, main system clock oscillation can be stopped with the PCC. The HALT mode can be used. However, the STOP mode cannot be used. (Subsystem clock oscillation cannot be stopped.)
- (f) The main system clock is divided and supplied to the peripheral hardware. The subsystem clock is supplied to 16-bit timer/event counter, the watch timer, and clock output functions only. Thus, 16-bit timer/event counter (when selecting watch timer output for count clock operating with subsystem clock), the watch function, and the clock output function can also be continued in the standby state. However, since all other peripheral hardware operate with the main system clock, the peripheral hardware also stops if the main system clock is stopped. (Except external input clock operation)

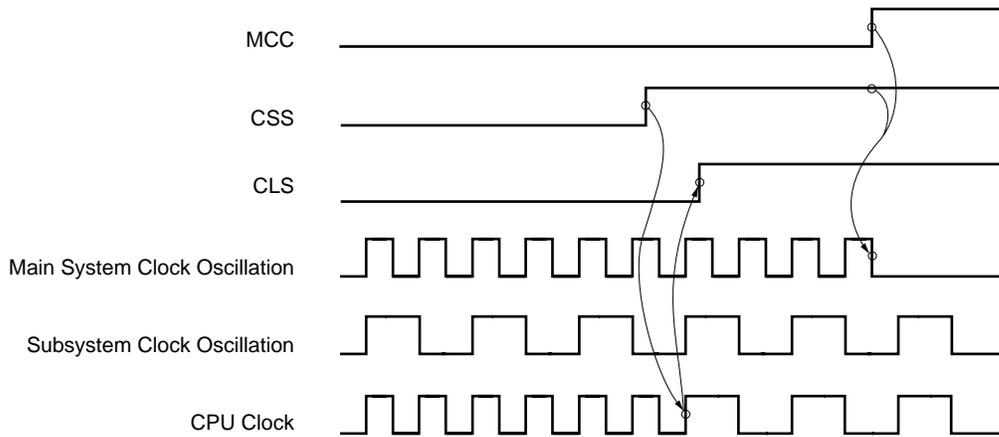
7.5.1 Main system clock operations

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

- (a) Because the operation guarantee instruction execution speed depends on the power supply voltage, the minimum instruction execution time can be changed by bits 0 to 2 (PCC0 to PCC2) of the PCC.
- (b) If bit 7 (MCC) of the PCC is set to 1 when operated with the main system clock, the main system clock oscillation does not stop. When bit 4 (CSS) of the PCC is set to 1 and the operation is switched to subsystem clock operation (CLS = 1) after that, the main system clock oscillation stops (see **Figure 7-9**).

Figure 7-9. Main System Clock Stop Function (1/2)

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



(b) Operation when MCC is set in case of main system clock operation

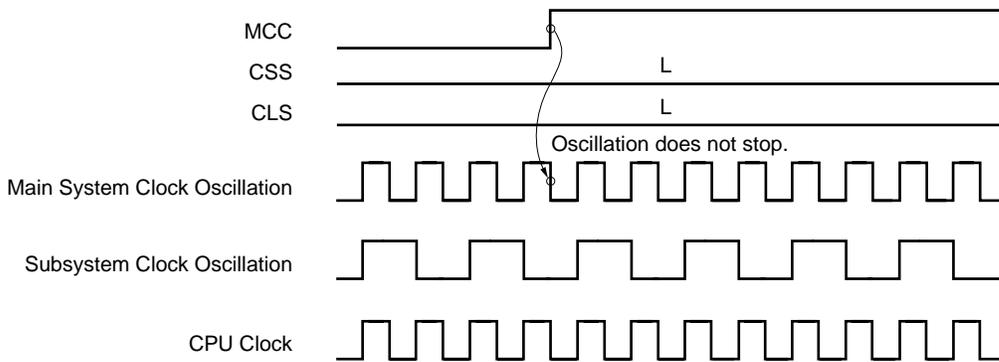
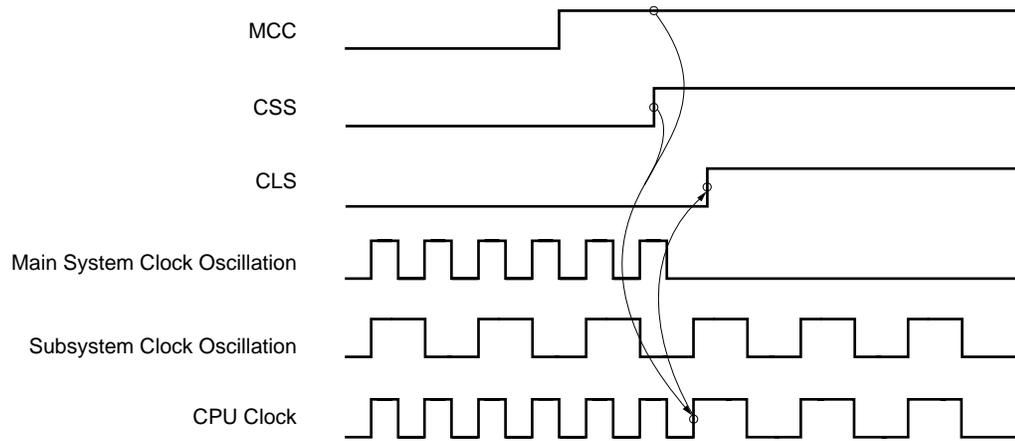


Figure 7-9. Main System Clock Stop Function (2/2)

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

**7.5.2 Subsystem clock operations**

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

- The minimum instruction execution time remains constant ($122 \mu\text{s}$ when operated at 32.768 kHz) irrespective of bits 0 to 2 (PCC0 to PCC2) of the PCC.
- Watchdog timer counting stops.

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

7.6 Changing System Clock and CPU Clock Settings

7.6.1 Time required for switchover between system clock and CPU clock

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

The actual switchover operation is not performed directly after writing to the PCC, but operation continues on the pre-switchover clock for several instructions (see Table 7-3).

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

Table 7-3. Maximum Time Required for CPU Clock Switchover

Set Values after Switchover					Set Values before Switchover																							
MCS	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0				
					0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	X	X	X
X	0	0	0	0	/				8 instructions				4 instructions				2 instructions				1 instruction				1 instruction			
		0	0	1					16 instructions				4 instructions				2 instructions				1 instruction				1 instruction			
		0	1	0					16 instructions				8 instructions				2 instructions				1 instruction				1 instruction			
		0	1	1					16 instructions				8 instructions				4 instructions				1 instruction				1 instruction			
		1	0	0					16 instructions				8 instructions				4 instructions				2 instructions				1 instruction			
1	1	X	X	X	f _x /2f _{xT} instruction (77 instructions)				f _x /4f _{xT} instruction (39 instructions)				f _x /8f _{xT} instruction (20 instructions)				f _x /16f _{xT} instruction (10 instructions)				f _x /32f _{xT} instruction (5 instructions)				/			
					0	f _x /4f _{xT} instruction (39 instructions)				f _x /8f _{xT} instruction (20 instructions)				f _x /16f _{xT} instruction (10 instructions)				f _x /32f _{xT} instruction (5 instructions)				f _x /64f _{xT} instruction (3 instructions)						

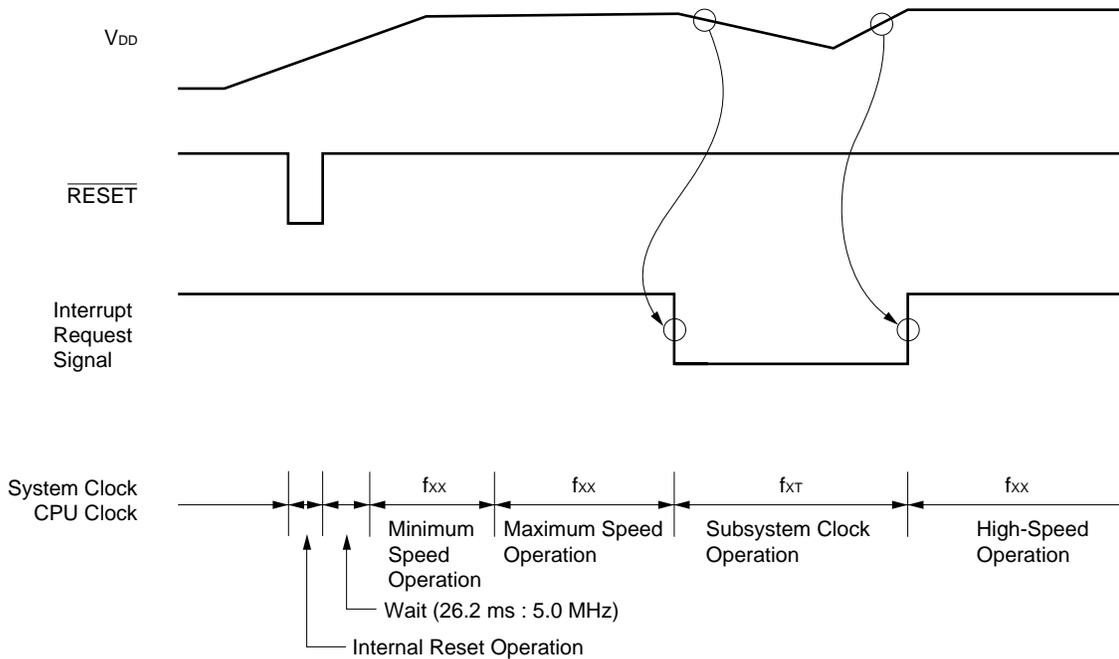
Caution Selection of the CPU clock cycle scaling factor (PCC0 to PCC2) and switchover from the main system clock to the subsystem clock (changing CSS from 0 to 1) should not be performed simultaneously. Simultaneous setting is possible, however, for selection of the CPU clock cycle scaling factor (PCC0 to PCC2) and switchover from the subsystem clock to the main system clock (changing CSS from 1 to 0).

- Remarks**
- One instruction is the minimum instruction execution time with the pre-switchover CPU clock.
 - Figures in parentheses apply to operation with f_x = 5.0 MHz and f_{xT} = 32.768 kHz.

7.6.2 System clock and CPU clock switching procedure

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

Figure 7-10. System Clock and CPU Clock Switching



- (1) The CPU is reset by setting the $\overline{\text{RESET}}$ signal to low level after power-on. After that, when reset is released by setting the $\overline{\text{RESET}}$ signal to high level, main system clock starts oscillation. At this time, oscillation stabilization time ($2^{17}/f_x$) is secured automatically. After that, the CPU starts executing the instruction at the minimum speed of the main system clock ($12.8 \mu\text{s}$ when operated at 5.0 MHz).
- (2) After the lapse of a sufficient time for the V_{DD} voltage to increase to enable operation at maximum speeds, the processor clock control register (PCC) and oscillation mode selection register (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 signal, the main system clock is switched to the subsystem clock (which must be in an oscillation stable state).
- (4) Upon detection of V_{DD} voltage reset due to an interrupt request signal, 0 is set to bit 7 (MCC) of PCC and oscillation of the main system clock is started. After the lapse of time required for stabilization of oscillation, the PCC and OSMS are rewritten and the maximum-speed operation is resumed.

Caution When subsystem clock is being operated while main system clock was stopped, if switching to the main system clock is made again, be sure to switch after securing oscillation stable time by software.

[MEMO]

CHAPTER 8 16-BIT TIMER/EVENT COUNTER

★ 8.1 Outline of Timers Incorporated into μ PD78078, 78078Y Subseries

This chapter explains the 16-bit timer/event counter. First of all, the timers incorporated into the μ PD78078, 78078Y Subseries and the related circuits are outlined below.

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

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

(2) 8-bit timers/event counters 1 and 2 (TM1 and TM2)

TM1 and TM2 can be used to serve as an interval timer and an external event counter and to output square waves with any selected frequency. Two 8-bit timer/event counters can be used as one 16-bit timer/event counter (See **CHAPTER 9 8-BIT TIMER/EVENT COUNTERS 1 AND 2**).

(3) 8-bit timer/event counters 5 and 6 (TM5 and TM6)

TM5 and TM6 can be used to serve as an interval timer and external event counter and to output any frequency square waves. These cannot be used as 16-bit timer/event counters (See **CHAPTER 10 8-BIT TIMER/EVENT COUNTER 5 AND 6**).

(4) Watch timer (TM3)

This timer can set a flag every 0.5 sec. and simultaneously generates interrupt requests at the preset time intervals (See **CHAPTER 11 WATCH TIMER**).

(5) Watchdog timer (WDTM)

WDTM can perform the watchdog timer function or generate non-maskable interrupt requests, maskable interrupt requests and $\overline{\text{RESET}}$ at the preset time intervals (See **CHAPTER 12 WATCHDOG TIMER**).

(6) Clock output control circuit

This circuit supplies other devices with the divided main system clock and the subsystem clock (See **CHAPTER 13 CLOCK OUTPUT CONTROL CIRCUIT**).

(7) Buzzer output control circuit

This circuit outputs the buzzer frequency obtained by dividing the main system clock (See **CHAPTER 14 BUZZER OUTPUT CONTROL CIRCUIT**).

Table 8-1. Timer/Event Counter Operations

		16-bit timer/ event counter	8-bit timer/event counters 1 and 2	8-bit timer/event counters 5 and 6	Watch timer	Watchdog timer
Operation mode	Interval timer	2 channels ^{Note 3}	2 channels	2 channels	1 channel ^{Note 1}	1 channel ^{Note 2}
	External event counter	√	√	√	—	—
Function	Timer output	√	√	√	—	—
	PWM output	√	—	√	—	—
	Pulse width measurement	√	—	—	—	—
	Square-wave output	√	√	√	—	—
	One-shot pulse output	√	—	—	—	—
	Interrupt request	√	√	√	√	√
	Test input	—	—	—	√	—

- Notes**
1. The watch timer can perform both watch timer and interval timer functions at the same time.
 2. The watchdog timer can perform either the watchdog timer function or the interval timer function.
 3. When capture/compare registers (CR00, CR01) are specified as compare registers.

8.2 16-Bit Timer/Event Counter Functions

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

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

TM0 can perform both PWM output and pulse width measurement at the same time.

(1) Interval timer

TM0 generates interrupt requests at the preset time interval.

Table 8-2. 16-Bit Timer/Event Counter Interval Times

Minimum Interval Time		Maximum Interval Time		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
2 x TI00 input cycle		2 ¹⁶ x TI00 input cycle		TI00 input edge cycle	
—	2 x 1/f _x (400 ns)	—	2 ¹⁶ x 1/f _x (13.1 ms)	—	1/f _x (200 ns)
2 x 1/f _x (400 ns)	2 ² x 1/f _x (800 ns)	2 ¹⁶ x 1/f _x (13.1 ms)	2 ¹⁷ x 1/f _x (26.2 ms)	1/f _x (200 ns)	2 x 1/f _x (400 ns)
2 ² x 1/f _x (800 ns)	2 ³ x 1/f _x (1.6 μs)	2 ¹⁷ x 1/f _x (26.2 ms)	2 ¹⁸ x 1/f _x (52.4 ms)	2 x 1/f _x (400 ns)	2 ² x 1/f _x (800 ns)
2 ³ x 1/f _x (1.6 μs)	2 ⁴ x 1/f _x (3.2 μs)	2 ¹⁸ x 1/f _x (52.4 ms)	2 ¹⁹ x 1/f _x (104.9 ms)	2 ² x 1/f _x (800 ns)	2 ³ x 1/f _x (1.6 μs)
2 x watch timer output cycle		2 ¹⁶ x watch timer output cycle		Watch timer output edge cycle	

- Remarks**
1. f_x : Main system clock oscillation frequency
 2. MCS : Bit 0 of oscillation mode selection register (OSMS)
 3. Figures in parentheses apply to operation with f_x = 5.0 MHz.

(2) PWM output

TM0 can generate 14-bit resolution PWM output.

(3) Pulse width measurement

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

(4) External event counter

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

(5) Square-wave output

TM0 can output a square wave with any selected frequency.

Table 8-3. 16-Bit Timer/Event Counter Square-Wave Output Ranges

Minimum Pulse Width		Maximum Pulse Width		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
2 x T100 input cycle		2 ¹⁶ x T100 input cycle		T100 input edge cycle	
—	2 x 1/f _x (400 ns)	—	2 ¹⁶ x 1/f _x (13.1 ms)	—	1/f _x (200 ns)
2 x 1/f _x (400 ns)	2 ² x 1/f _x (800 ns)	2 ¹⁶ x 1/f _x (13.1 ms)	2 ¹⁷ x 1/f _x (26.2 ms)	1/f _x (200 ns)	2 x 1/f _x (400 ns)
2 ² x 1/f _x (800 ns)	2 ³ x 1/f _x (1.6 μs)	2 ¹⁷ x 1/f _x (26.2 ms)	2 ¹⁸ x 1/f _x (52.4 ms)	2 x 1/f _x (400 ns)	2 ² x 1/f _x (800 ns)
2 ³ x 1/f _x (1.6 μs)	2 ⁴ x 1/f _x (3.2 μs)	2 ¹⁸ x 1/f _x (52.4 ms)	2 ¹⁹ x 1/f _x (104.9 ms)	2 ² x 1/f _x (800 ns)	2 ³ x 1/f _x (1.6 μs)
2 x watch timer output cycle		2 ¹⁶ x watch timer output cycle		Watch timer output edge cycle	

- Remarks**
1. f_x : Main system clock oscillation frequency
 2. MCS : Bit 0 of oscillation mode selection register (OSMS)
 3. Figures in parentheses operation with f_x = 5.0 MHz.

(6) One-shot pulse output

TM0 is able to output one-shot pulse which can set any width of output pulse.

8.3 16-Bit Timer/Event Counter Configuration

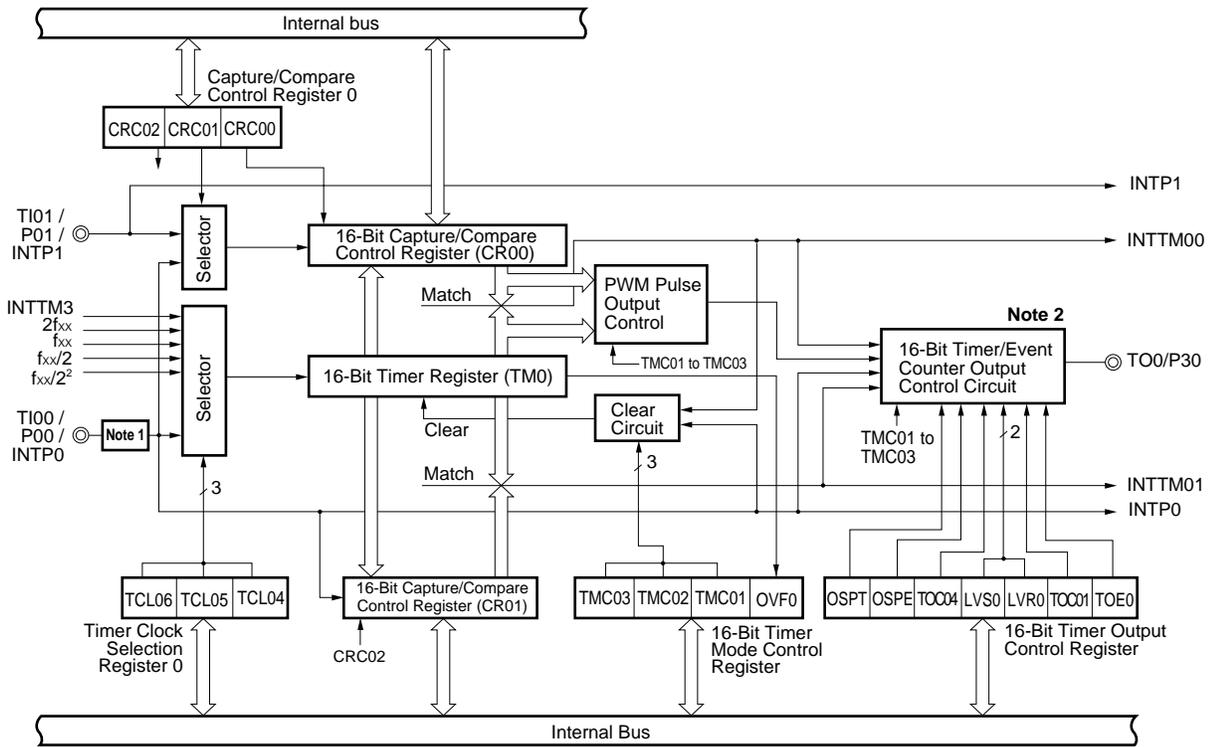
The 16-bit timer/event counter consists of the following hardware.

Table 8-4. 16-Bit Timer/Event Counter Configuration

Item	Configuration
Timer register	16 bits x 1 (TM0)
Register	Capture/compare register: 16 bits x 2 (CR00, CR01)
Timer output	1 (TO0)
Control register	Timer clock select register 0 (TCL0) 16-bit timer mode control register (TMC0) Capture/compare control register 0 (CRC0) 16-bit timer output control register (TOC0) Port mode register 3 (PM3) External interrupt mode register 0 (INTM0) Sampling clock select register (SCS) ^{Note}

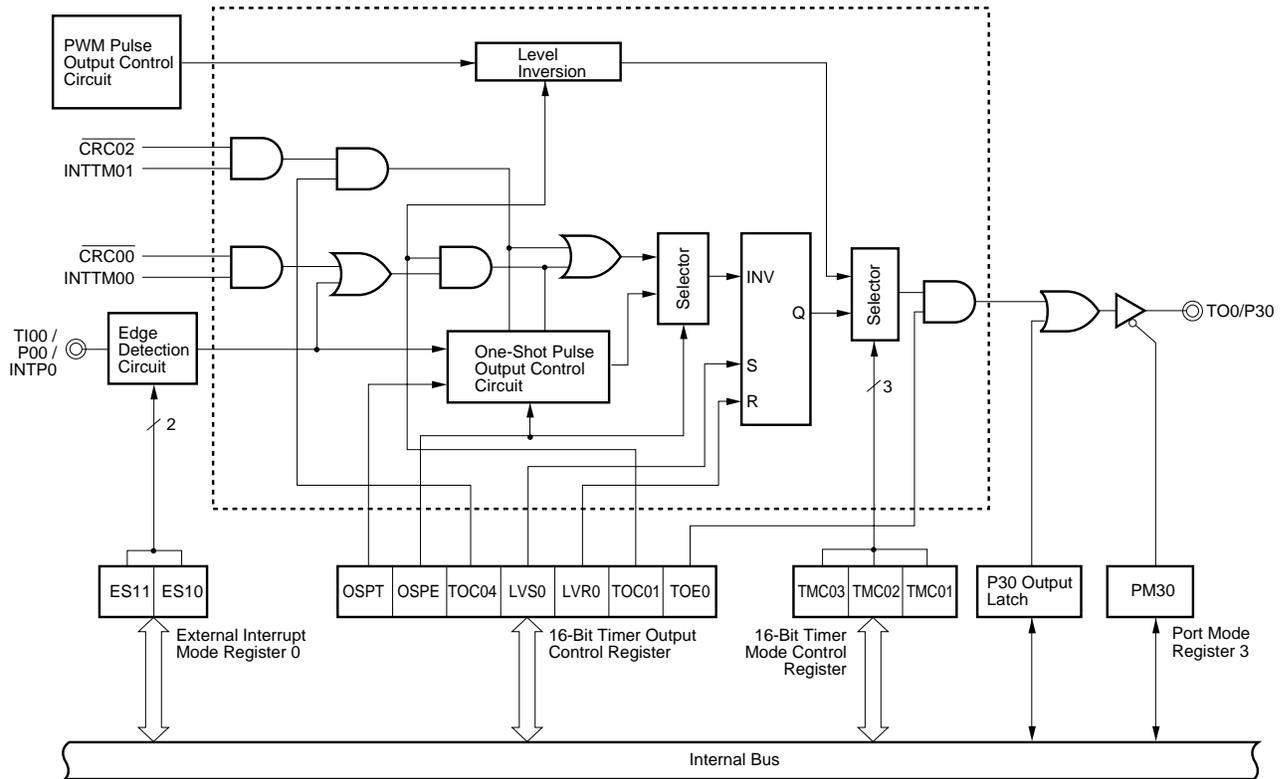
Note Refer to **Figure 22-1 Basic Configuration of Interrupt Function**.

Figure 8-1. 16-Bit Timer/Event Counter Block Diagram



- Notes**
1. Edge detection circuit
 2. The configuration of the 16-bit timer/event counter output control circuit is shown in Figure 8-2.

Figure 8-2. 16-Bit Timer/Event Counter Output Control Circuit Block Diagram



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

(1) Capture/compare register 00 (CR00)

CR00 is a 16-bit register which has the functions of both a capture register and a compare register. Whether it is used as a capture register or as a compare register is set by bit 0 (CRC00) of capture/compare control register 0 (CRC0).

When CR00 is used as a compare register, the value set in the CR00 is constantly compared with the 16-bit timer register (TM0) count value, and an interrupt request (INTTM00) is generated if they match. CR00 is used as a register to hold the interval time when TM0 is set to interval timer operation, and it is used as a register to set the pulse width when TM0 is set to the PWM output operation.

When CR00 is used as a capture register, it is possible to select the valid edge of the INTP0/TI00 pin or the INTP1/TI01 pin as the capture trigger. The INTP0/TI00 or INTP1/TI01 valid edge is set by means of external interrupt mode register 0 (INTM0).

If CR00 is specified as a capture register and capture trigger is specified to be the valid edge of the INTP0/TI00 pin, the situation is as shown in the following table.

Table 8-5. INTP0/TI00 Pin Valid Edge and CR00 Capture Trigger Valid Edge

ES11	ES10	INTP0/TI00 Pin Valid Edge	CR00 Capture Trigger Valid Edge
0	0	Falling edge	Rising edge
0	1	Rising edge	Falling edge
1	0	Setting prohibited	
1	1	Both rising and falling edges	No capture operation

CR00 is set by a 16-bit memory manipulation instruction.

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

- Cautions**
- 1. Set the PWM data (14 bits) to the higher 14 bits of CR00 and set 00 to the lower 2 bits.**
 - 2. Set values other than 0000H to CR00. Therefore, when used as an event counter, 1-pulse count operation cannot be executed.**
 - 3. When the value after CR00 is changed is smaller than the 16-bit timer register (TM0) value, TM0 continues to count, overflows, and resumes counting from zero. Therefore, when the value after CR00 is changed (M) is smaller than the value before CR00 is changed (N), it is necessary to restart the timer after changing CR00.**

(2) Capture/compare register 01 (CR01)

CR01 is a 16-bit register which has the functions of both a capture register and a compare register. Whether it is used as a capture register or a compare register is set by bit 2 (CRC02) of capture/compare control register 0.

When CR01 is used as a compare register, the value set in the CR01 is constantly compared with the 16-bit timer register (TM0) count value, and an interrupt request (INTTM01) is generated if they match.

When CR01 is used as a capture register, it is possible to select the valid edge of the INTP0/TI00 pin as the capture trigger. Setting of the INTP0/TI00 valid edge is set by means of external interrupt mode register 0 (INTM0).

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

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

Caution If a valid edge of TI0/P00 pin is input while reading out CR01, CR01 does not carry out capture operation and retains the data. However, the interrupt request flag (PIF0) is set by the detection of a valid edge.

(3) 16-bit timer register (TM0)

TM0 is a 16-bit register which counts the count pulses.

TM0 is read by a 16-bit memory manipulation instruction. When TM0 is read, capture/compare register (CR01) should first be set as a capture register.

$\overline{\text{RESET}}$ input sets TM0 to 0000H.

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

8.4 16-Bit Timer/Event Counter Control Registers

The following seven types of registers are used to control the 16-bit timer/event counter.

- Timer clock select register 0 (TCL0)
- 16-bit timer mode control register (TMC0)
- Capture/compare control register 0 (CRC0)
- 16-bit timer output control register (TOC0)
- Port mode register 3 (PM3)
- External interrupt mode register 0 (INTM0)
- Sampling clock select register (SCS)

(1) Timer clock select register 0 (TCL0)

This register is used to set the count clock of the 16-bit timer register.

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

$\overline{\text{RESET}}$ input sets TCL0 value to 00H.

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

Figure 8-3. Timer Clock Selection Register 0 Format

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

				PCL Output Clock Selection		
TCL03	TCL02	TCL01	TCL00	MCS = 1		MCS = 0
0	0	0	0	f _{XT} (32.768 kHz)		
0	1	0	1	f _{xx}	f _x (5.0 MHz)	f _x /2 (2.5 MHz)
0	1	1	0	f _{xx} /2	f _x /2 (2.5 MHz)	f _x /2 ² (1.25 MHz)
0	1	1	1	f _{xx} /2 ²	f _x /2 ² (1.25 MHz)	f _x /2 ³ (625 kHz)
1	0	0	0	f _{xx} /2 ³	f _x /2 ³ (625 kHz)	f _x /2 ⁴ (313 kHz)
1	0	0	1	f _{xx} /2 ⁴	f _x /2 ⁴ (313 kHz)	f _x /2 ⁵ (156 kHz)
1	0	1	0	f _{xx} /2 ⁵	f _x /2 ⁵ (156 kHz)	f _x /2 ⁶ (78.1 kHz)
1	0	1	1	f _{xx} /2 ⁶	f _x /2 ⁶ (78.1 kHz)	f _x /2 ⁷ (39.1 kHz)
1	1	0	0	f _{xx} /2 ⁷	f _x /2 ⁷ (39.1 kHz)	f _x /2 ⁸ (19.5 kHz)
Other than above				Setting prohibited		

			16-Bit Timer Register Count Clock Selection		
TCL06	TCL05	TCL04	MCS = 1		MCS = 0
0	0	0	TI00 (Valid edge specifiable)		
0	0	1	2f _{xx}	Setting prohibited	f _x (5.0 MHz)
0	1	0	f _{xx}	f _x (5.0 MHz)	f _x /2 (2.5 MHz)
0	1	1	f _{xx} /2	f _x /2 (2.5 MHz)	f _x /2 ² (1.25 MHz)
1	0	0	f _{xx} /2 ²	f _x /2 ² (1.25 MHz)	f _x /2 ³ (625 kHz)
1	1	1	Watch timer output (INTTM3)		
Other than above			Setting prohibited		

CLOE	PCL Output Control
0	Output disabled
1	Output enabled

- Cautions**
- External interrupt mode register 0 (INTM0) sets the TI00/INTP0 pin valid edge, and the sampling clock selection register (SCS) selects the sampling clock.
 - When enabling PCL output, set TCL00 to TCL03, then set 1 in CLOE with a 1-bit memory manipulation instruction.
 - To read the count value when TI00 has been specified as the TM0 count clock, the value should be read from TM0, not from capture/compare register 01 (CR01).
 - When rewriting TCL0 to other data, stop the timer operation beforehand.

- Remarks**
1. f_{xx} : Main system clock frequency (f_x or $f_x/2$)
 2. f_x : Main system clock oscillation frequency
 3. f_{XT} : Subsystem clock oscillation frequency
 4. TI00 : 16-bit timer/event counter input pin
 5. TM0 : 16-bit timer register
 6. MCS : Bit 0 of oscillation mode selection register (OSMS)
 7. Figures in parentheses apply to operation with $f_x = 5.0$ MHz or $f_{XT} = 32.768$ kHz.

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

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

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

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

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

Figure 8-4. 16-Bit Timer Mode Control Register Format

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

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

TMC03	TMC02	TMC01	Operating Mode and Clear Mode	TO0 Output Timing	Interrupt Generation
0	0	0	Operation stop (TM0 cleared to 0)	No change	Not Generated
0	0	1	PWM mode (free running)	PWM pulse output	Generated when TM0 and CR00 match, or TM0 and CR01 match.
0	1	0	Free running mode	TM0 and CR00 match, or TM0 and CR01 match.	
0	1	1		TM0 and CR00 match, TM0 and CR01 match, or on TI00 valid edge.	
1	0	0	Clear & start on TI00 valid edge	TM0 and CR00 match, or TM0 and CR01 match.	
1	0	1		TM0 and CR00 match, TM0 and CR01 match, or on TI00 valid edge	
1	1	0	Clear & start when TM0 and CR00 match	TM0 and CR00 match, or TM0 and CR01 match.	
1	1	1		TM0 and CR00 match, TM0 and CR01 match, or on TI00 valid edge	

★

- Cautions**
1. Switch the clear mode and the T00 output timing after stopping the timer operation (by setting TMC01 to TMC03 to 0, 0, 0).
 2. Set the valid edge of the TI00/INTP0 pin with an external interrupt mode register 0 (INTM0) and select the sampling clock frequency with a sampling clock select register (SCS).
 3. When using the PWM mode, set the PWM and then set data to CR00.
 4. If the mode is set so that the timer is cleared and starts when TM0 and CR00 match, the OVF0 flag is set to 1 when the TM0 value changes from 0FFFFH to 0000H with CR00 set to FFFFH.

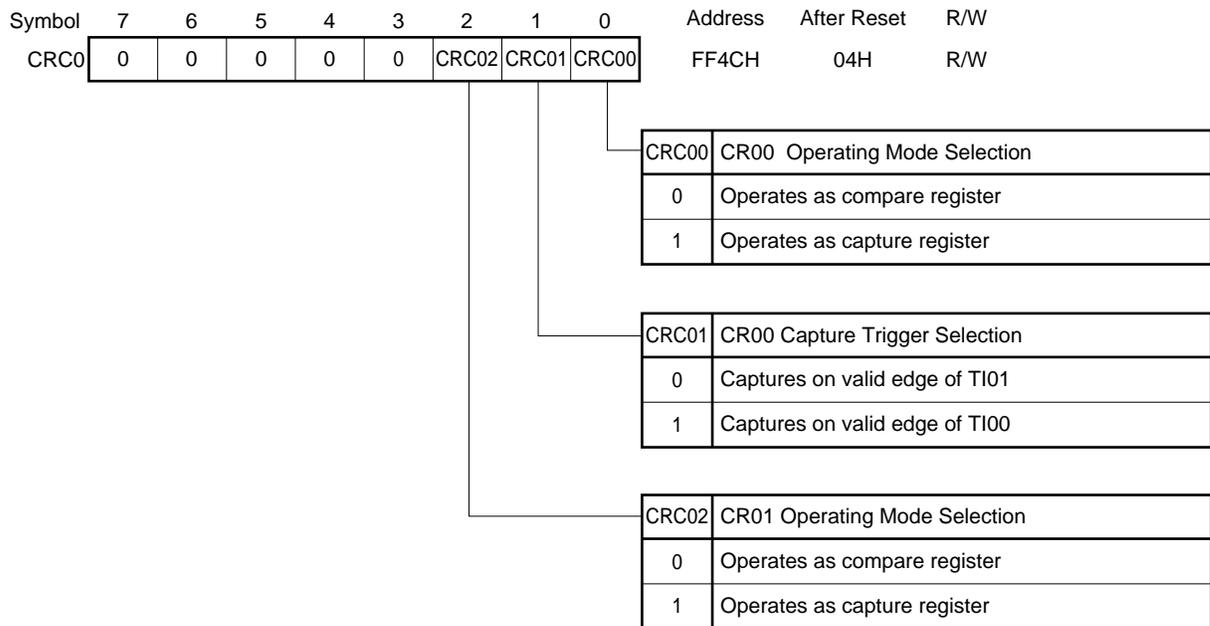
Remark

TO0 : 16-bit timer/event counter output pin
 TI00 : 16-bit timer/event counter input pin
 TM0 : 16-bit timer register
 CR00 : Compare register 00
 CR01 : Compare register 01

(3) Capture/compare control register 0 (CRC0)

This register controls the operation of the capture/compare registers (CR00, CR01).
 CRC0 is set with a 1-bit or 8-bit memory manipulation instruction.
 RESET input sets CRC0 value to 04H.

Figure 8-5. Capture/Compare Control Register 0 Format



- Cautions**
1. Timer operation must be stopped before setting CRC0.
 2. When clear & start mode on a match between TM0 and CR00 is selected with the 16-bit timer mode control register, CR00 should not be specified as a capture register.

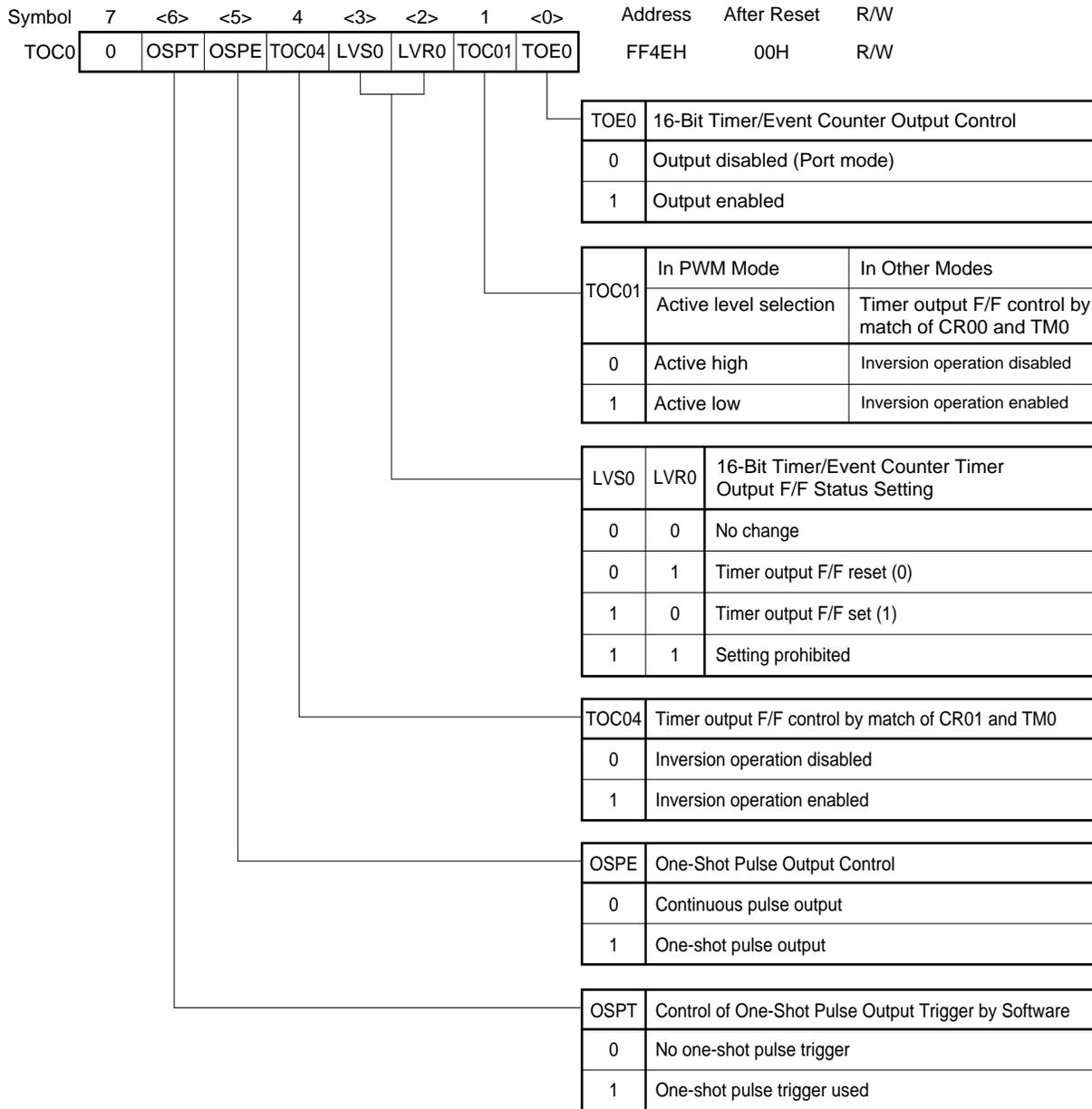
(4) 16-bit timer output control register (TOC0)

This register controls the operation of the 16-bit timer/event counter output control circuit. It sets R-S type flip-flop (LV0) setting/resetting, the active level in PWM mode, inversion enabling/disabling in modes other than PWM mode, 16-bit timer/event counter timer output enabling/disabling, one-shot pulse output operation enabling/disabling, and output trigger for a one-shop pulse by software.

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

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

Figure 8-6. 16-Bit Timer Output Control Register Format



- Cautions**
1. Timer operation must be stopped before setting TOC0 (except OSPT).
 2. If LVS0 and LVR0 are read after data is set, they will be 0.
 3. OSPT is cleared automatically after data setting, and will therefore be 0 if read.

(5) Port mode register 3 (PM3)

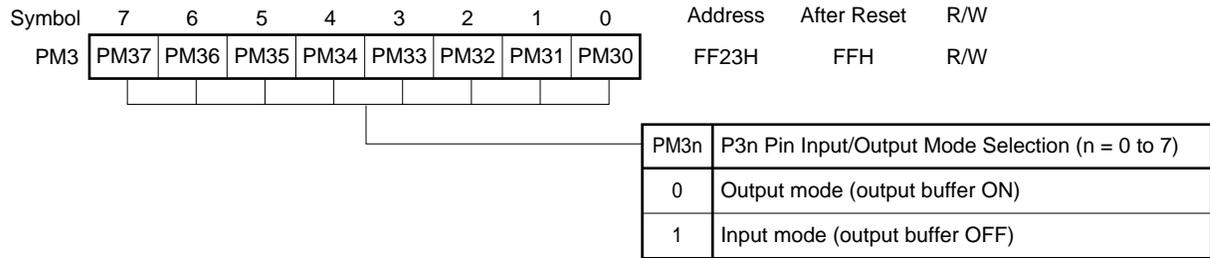
This register sets port 3 input/output in 1-bit units.

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

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

RESET input sets PM3 value to FFH.

Figure 8-7. Port Mode Register 3 Format



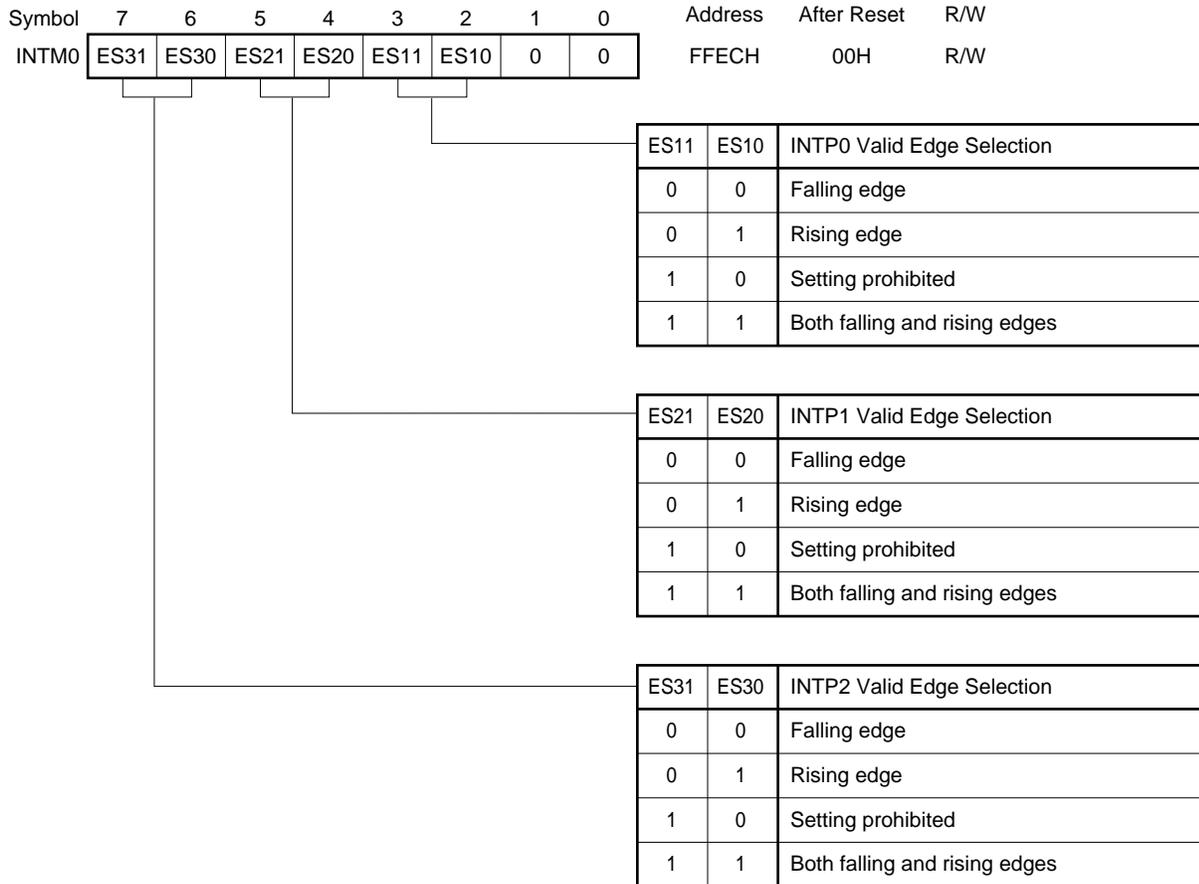
(6) External interrupt mode register 0 (INTM0)

This register is used to set INTP0 to INTP2 valid edges.

INTM0 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets INTM0 value to 00H.

Figure 8-8. External Interrupt Mode Register 0 Format



Caution Set 0, 0, 0 to bits 1 through 3 (TMC01 through TMC03) of the 16-bit timer mode control register (TMC0) and stop the timer operation before setting the valid edges of INTP0/TI00/P00 pins.

(7) Sampling clock select registers (SCS)

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

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

RESET input sets SCS value to 00H.

Figure 8-9. Sampling Clock Select Register Format



Caution $f_{xx}/2^N$ is the clock supplied to the CPU, and $f_{xx}/2^5$, $f_{xx}/2^6$, and $f_{xx}/2^7$ are clocks supplied to peripheral hardware. $f_{xx}/2^N$ is stopped in HALT mode.

- Remarks**
1. N : Value set in bits 0 to 2 (PCC0 to PCC2) of the processor clock control register (PCC) (N = 0 to 4)
 2. f_{xx} : Main system clock frequency (fx or fx/2)
 3. f_x : Main system clock oscillation frequency
 4. MCS : Bit 0 of oscillation mode selection register (OSMS)
 5. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

8.5 16-Bit Timer/Event Counter Operations

8.5.1 Interval timer operations

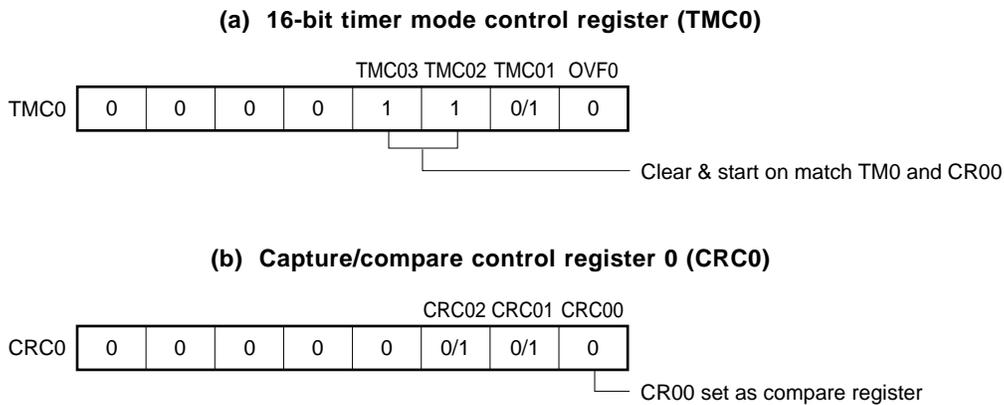
Setting the 16-bit timer mode control register (TMC0) and capture/compare control register 0 (CRC0) as shown in Figure 8-10 allows operation as an interval timer. Interrupt requests are generated repeatedly using the count value set in 16-bit capture/compare register 00 (CR00) beforehand is used as the interval.

When the count value of the 16-bit timer register (TM0) matches the value set to CR00, counting continues with the TM0 value cleared to 0 and the interrupt request signal (INTTM00) is generated.

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

For the operation in the case the value of the compare register is changed during the timer count operation, refer to **8.6 16-Bit Timer/Event Counter Precautions (3)**.

Figure 8-10. Control Register Settings for Interval Timer Operation



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the interval timer. See the description of the respective control registers for details.

Figure 8-11. Interval Timer Configuration Diagram

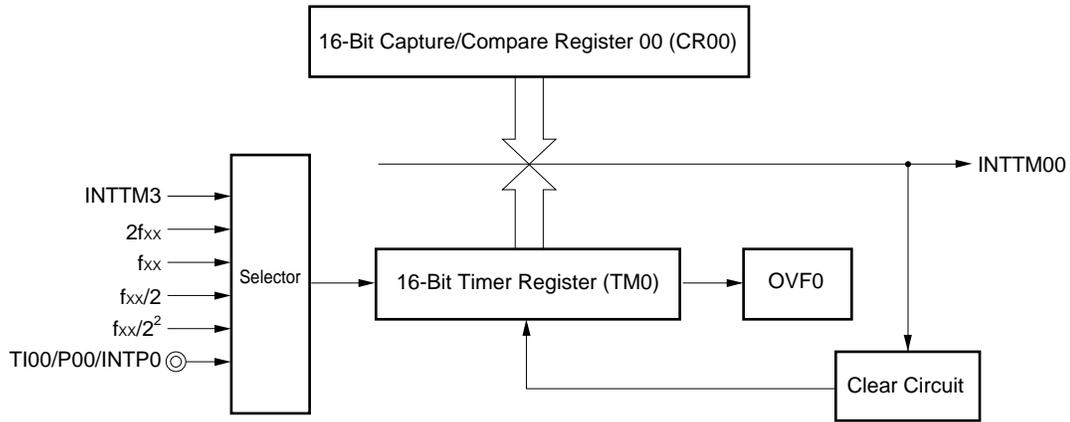
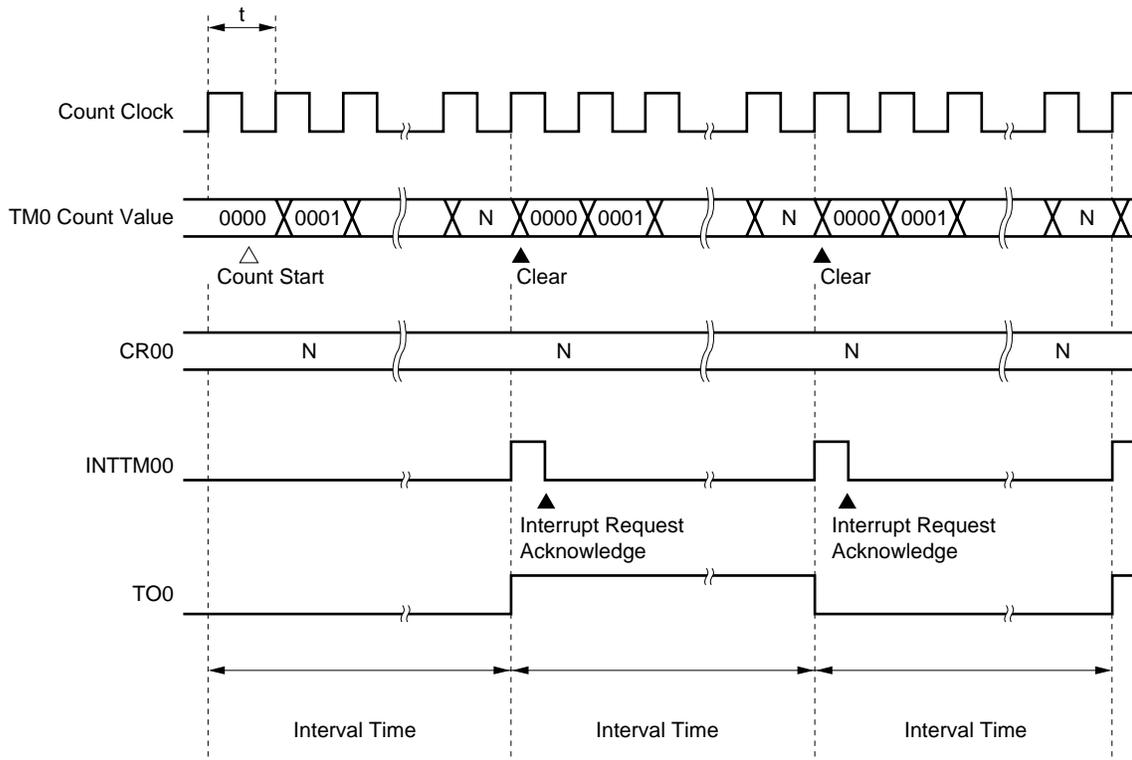


Figure 8-12. Interval Timer Operation Timings



Remark Interval time = $(N + 1) \times t$: $N = 0001H$ to $FFFFH$.

Table 8-6. 16-Bit Timer/Event Counter Interval Times

TCL06	TCL05	TCL4	Minimum Interval Time		Maximum Interval Time		Resolution	
			MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
0	0	0	2 x TI00 input cycle		2 ¹⁶ x TI00 input cycle		TI00 input edge cycle	
0	0	1	Setting prohibited	2 x 1/f _x (400 ns)	Setting prohibited	2 ¹⁶ x 1/f _x (13.1 ms)	Setting prohibited	1/f _x (200 ns)
0	1	0	2 x 1/f _x (400 ns)	2 ² x 1/f _x (800 ns)	2 ¹⁶ x 1/f _x (13.1 ms)	2 ¹⁷ x 1/f _x (26.2 ms)	1/f _x (200 ns)	2 x 1/f _x (400 ns)
0	1	1	2 ² x 1/f _x (800 ns)	2 ³ x 1/f _x (1.6 μs)	2 ¹⁷ x 1/f _x (26.2 ms)	2 ¹⁸ x 1/f _x (52.4 ms)	2 x 1/f _x (400 ns)	2 ² x 1/f _x (800 ns)
1	0	0	2 ³ x 1/f _x (1.6 μs)	2 ⁴ x 1/f _x (3.2 μs)	2 ¹⁸ x 1/f _x (52.4 ms)	2 ¹⁹ x 1/f _x (104.9 ms)	2 ² x 1/f _x (800 ns)	2 ³ x 1/f _x (1.6 μs)
1	1	1	2 x watch timer output cycle		2 ¹⁶ x watch timer output cycle		Watch timer output edge cycle	
Other than above			Setting prohibited					

- Remarks**
1. f_x : Main system clock oscillation frequency
 2. MCS : Bit 0 of oscillation mode selection register (OSMS)
 3. TCL04 to TCL06 : Bits 4 to 6 of timer clock select register 0 (TCL0)
 4. Figures in parentheses apply to operation with f_x = 5.0 MHz.

8.5.2 PWM output operations

Setting the 16-bit timer mode control register (TMC0), capture/compare control register 0 (CRC0), and the 16-bit timer output control register (TOC0) as shown in Figure 8-13 allows operation as PWM output. Pulses with the duty rate determined by the value set in 16-bit capture/compare register 00 (CR00) beforehand are output from the TO0/P30 pin.

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

This PWM pulse has a 14-bit resolution. The pulse can be converted to an analog voltage by integrating it with an external low-pass filter (LPF). The PWM pulse has a combination of the basic cycle determined by 2⁸/Φ and the sub-cycle determined by 2¹⁴/Φ so that the time constant of the external LPF can be shortened. Count clock Φ can be selected with bits 4 to 6 (TCL04 to TCL06) of the timer clock select register (TCL0).

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

- Cautions**
1. PWM operation mode should be selected before setting CR00.
 2. Write 0 to bits 0 and 1 of CR00.
 3. Do not select PWM operation mode for external clock input from the TI00/P00/INTP0 pin.

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

The analog output voltage (V_{AN}) used for D/A conversion with the configuration shown in Figure 8-14 is as follows.

$$V_{AN} = V_{REF} \times \frac{\text{capture/compare register 00 (CR00) value}}{2^{16}}$$

V_{REF} : External switching circuit reference voltage

Figure 8-14. Example of D/A Converter Configuration with PWM Output

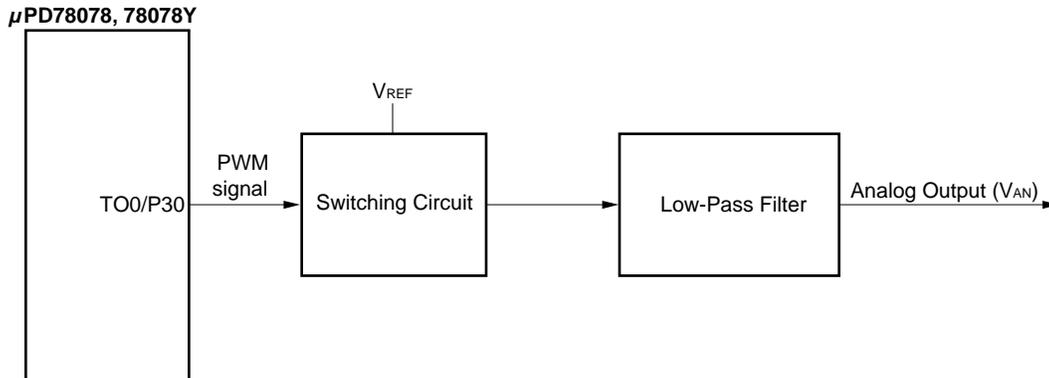
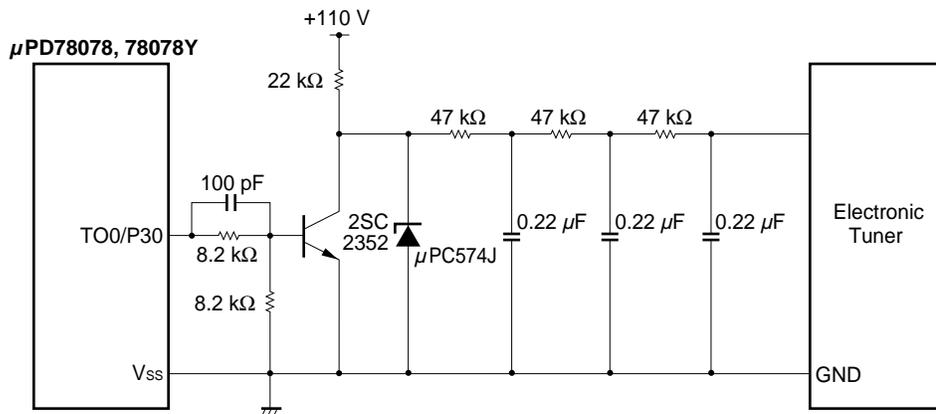


Figure 8-15 shows an example in which PWM output is converted to an analog voltage and used in a voltage synthesizer type TV tuner.

Figure 8-15. TV Tuner Application Circuit Example

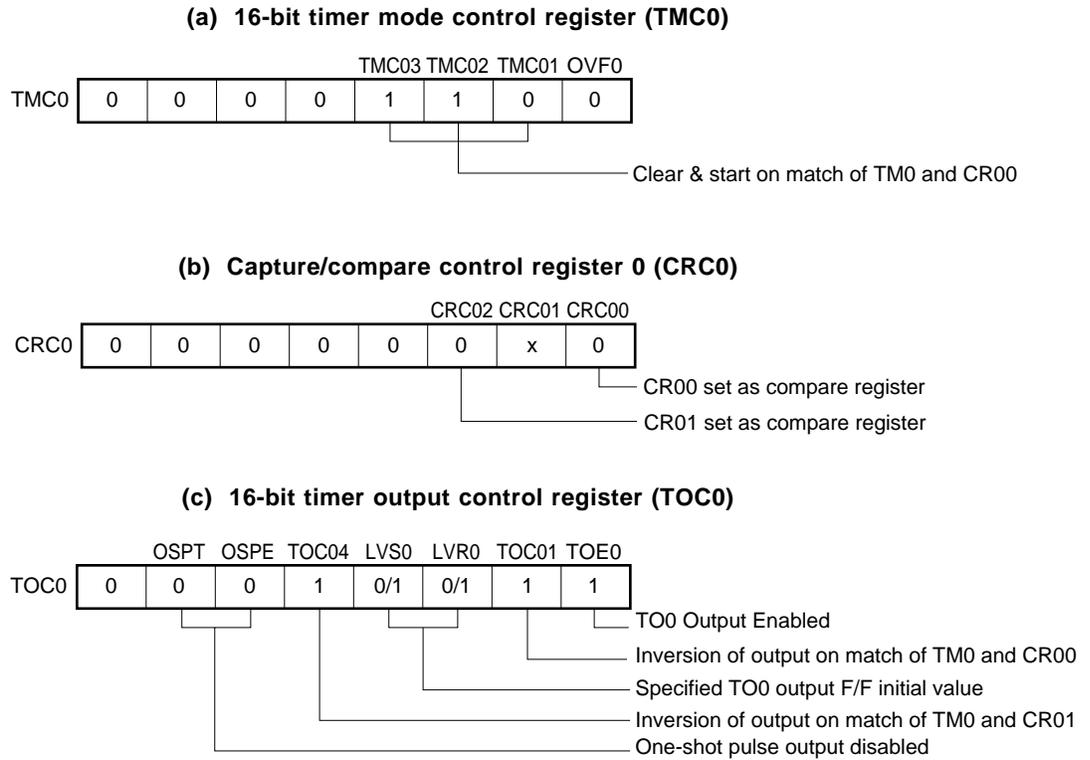


8.5.3 PPG output operations

Setting the 16-bit timer mode control register (TMC0) and capture/compare control register 0 (CRC0) as shown in Figure 8-16 allows operation as PPG (Programmable Pulse Generator) output.

In the PPG output operation, square waves are output from the TO0/P30 pin with the pulse width and the cycle that correspond to the count values set beforehand in 16-bit capture/compare register 01 (CR01) and in 16-bit capture/compare register 00 (CR00), respectively.

Figure 8-16. Control Register Settings for PPG Output Operation



Caution Values in the following range should be set in CR00 and CR01.

$$0000H \leq CR01 < CR00 \leq FFFFH$$

Remark x : Don't care

8.5.4 Pulse width measurement operations

It is possible to measure the pulse width of the signals input to the TI00/P00 pin and TI01/P01 pin using the bit timer register (TM0).

There are two measurement methods: measuring with TM0 used in free-running mode, and measuring by restarting the timer in synchronization with the edge of the signal input to the TI00/P00 pin.

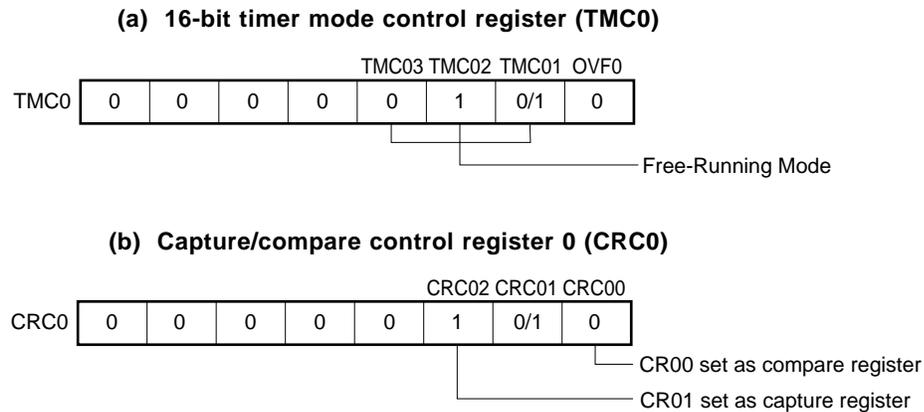
(1) Pulse width measurement with free-running counter and one capture register

When the 16-bit timer register (TM0) is operated in free-running mode (see register settings in Figure 8-17), and the edge specified by external interrupt mode register 0 (INTM0) is input to the TI00/P00 pin, the value of TM0 is taken into 16-bit capture/compare register 01 (CR01) and an external interrupt request signal (INTP0) is set.

Any of three edge specifications can be selected—rising, falling, or both edges—by means of bits 2 and 3 (EX10 and ES11) of INTM0.

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

Figure 8-17. Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See the description of the respective control registers for details.

Figure 8-18. Configuration Diagram for Pulse Width Measurement by Free-Running Counter

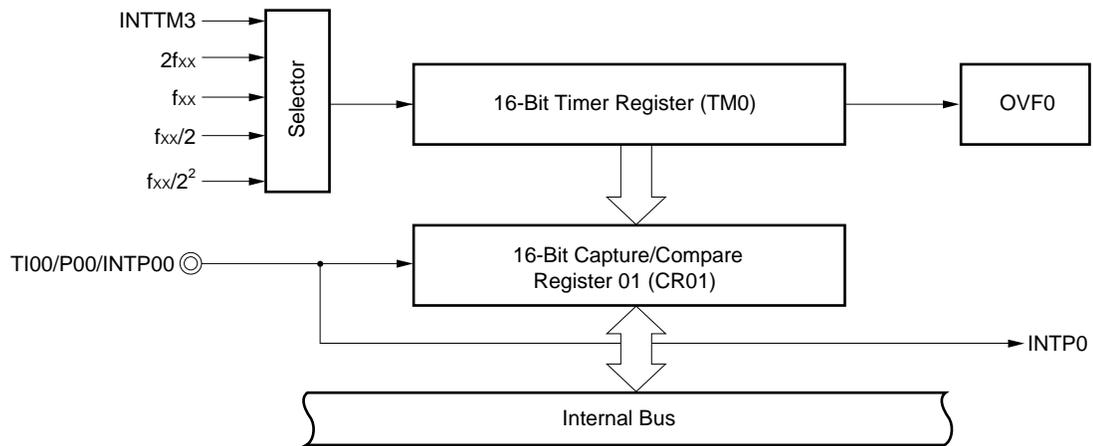
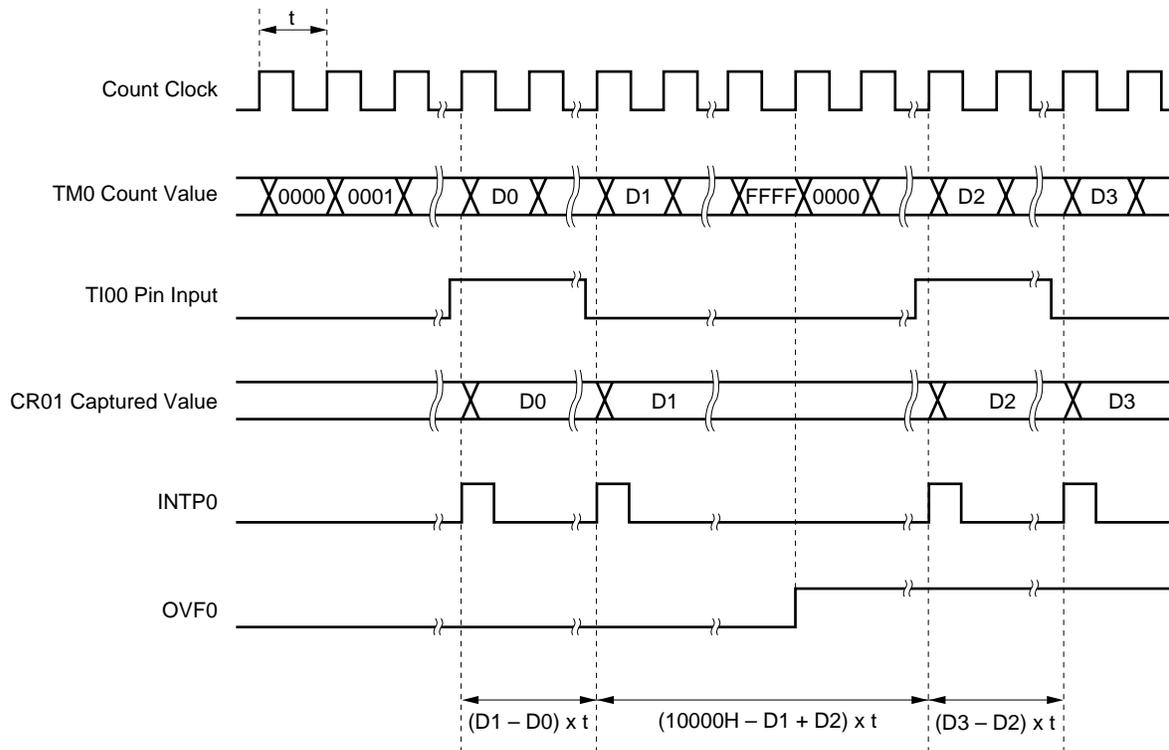


Figure 8-19. Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)



(2) Two pulse width measurements with free-running counter

When the 16-bit timer register (TM0) is operated in free-running mode (see register settings in Figure 8-20), it is possible to simultaneously measure the pulse widths of the two signals input to the TI00/P00 pin and the TI01/P01 pin.

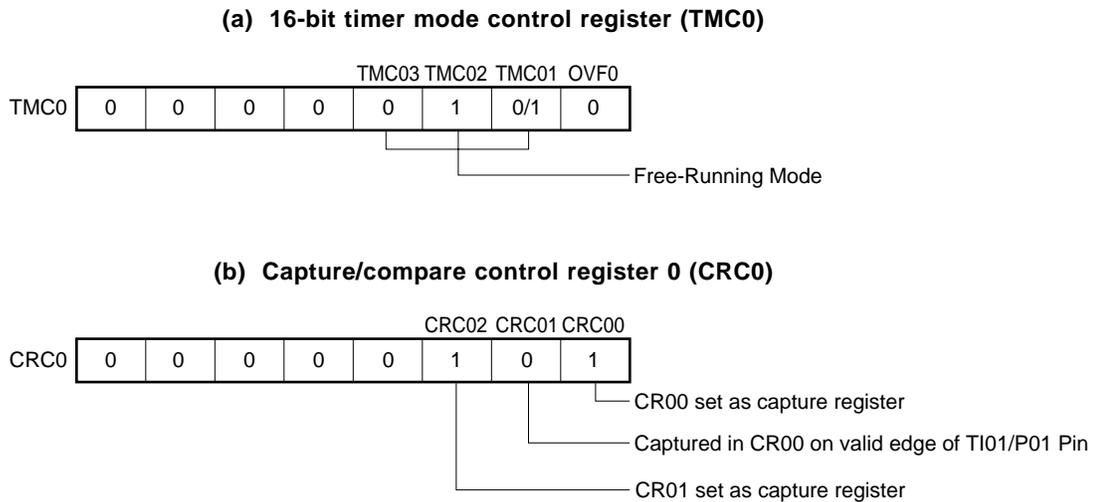
When the edge specified by bits 2 and 3 (ES10 and ES11) of external interrupt mode register 0 (INTM0) is input to the TI00/P00 pin, the value of TM0 is taken into 16-bit capture/compare register 01 (CR01) and an external interrupt request signal (INTP0) is set.

Also, when the edge specified by bits 4 and 5 (ES20 and ES21) of INTM0 is input to the TI01/P01 pin, the value of TM0 is taken into 16-bit capture/compare register 00 (CR00) and an external interrupt request signal (INTP1) is set.

Any of three edge specifications can be selected-rising, falling, or both edges-as the valid edges for the TI00/P00 pin and the TI01/P01 pin by means of bits 2 and 3 (ES10 and ES11) and bits 4 and 5 (ES20 and ES21) of INTM0, respectively.

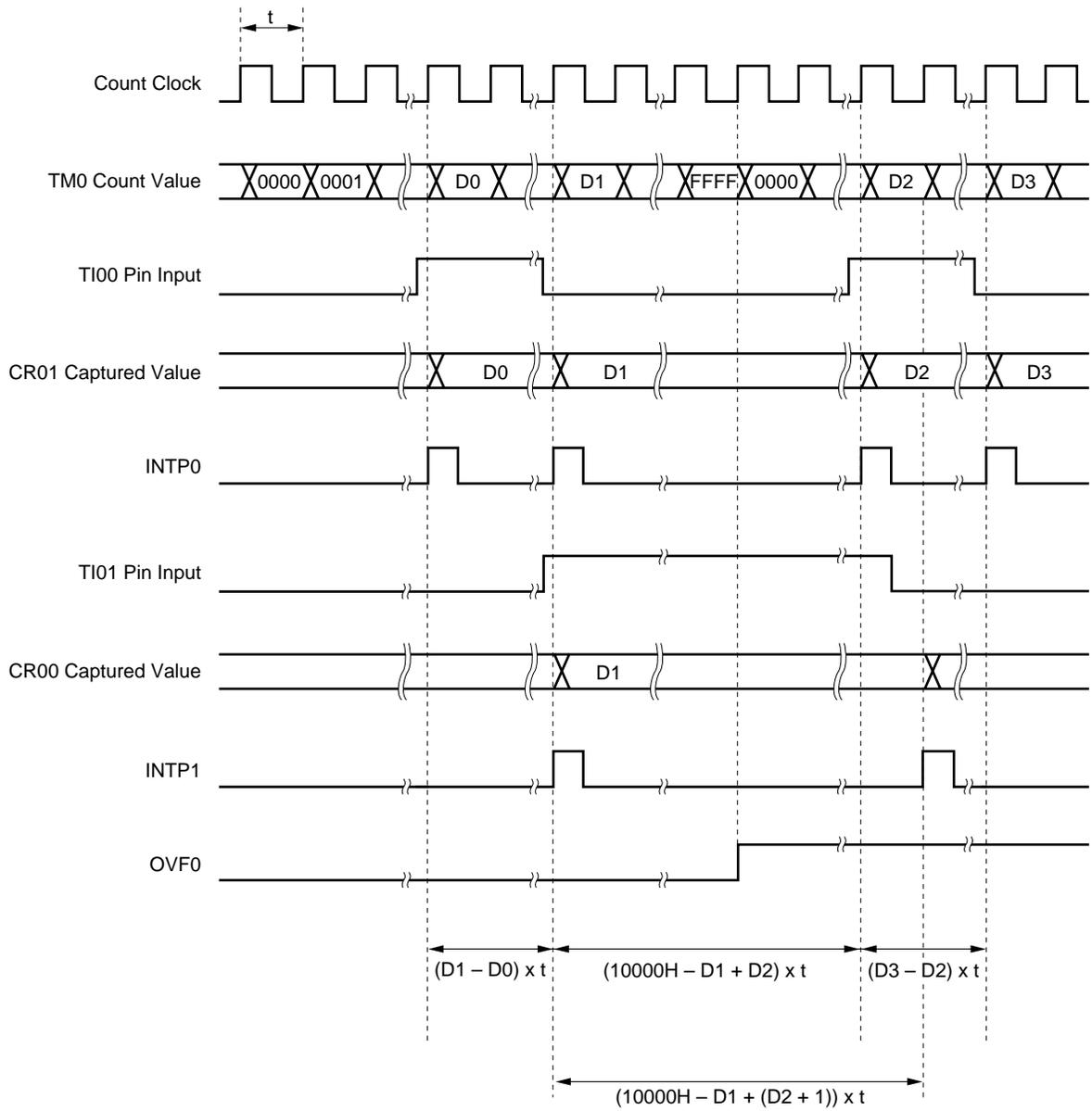
For TI00/P00 pin valid edge detection, sampling is performed at the interval selected by means of the sampling clock selection register (SCS), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Figure 8-20. Control Register Settings for Two Pulse Width Measurements with Free-Running Counter



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See the description of the respective control registers for details.

Figure 8-21. Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified)



(3) Pulse width measurement with free-running counter and two capture registers

When the 16-bit timer register (TM0) is operated in free-running mode (see register settings in Figure 8-22), it is possible to measure the pulse width of the signal input to the TI00/P00 pin.

When the edge specified by bits 2 and 3 (ES10 and ES11) of external interrupt mode register 0 (INTM0) is input to the TI00/P00 pin, the value of TM0 is taken into 16-bit capture/compare register 01 (CR01) and an external interrupt request signal (INTP0) is set.

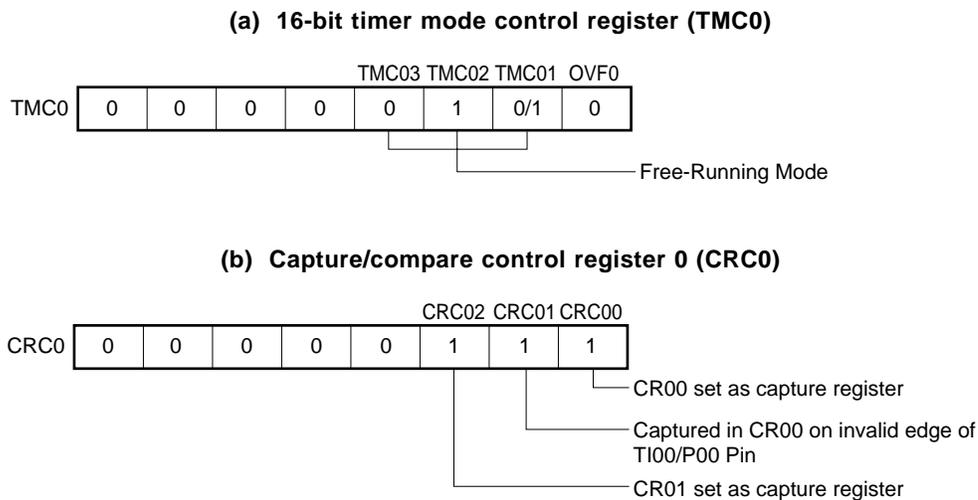
Also, on the inverse edge input of that of the capture operation into CR01, the value of TM0 is taken into 16-bit capture/compare register 00 (CR00).

Either of two edge specifications can be selected—rising or falling—as the valid edges for the TI00/P00 pin by means of bits 2 and 3 (ES10 and ES11) of INTM0.

For TI00/P00 pin valid edge detection, sampling is performed at the interval selected by means of the sampling clock selection register (SCS), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

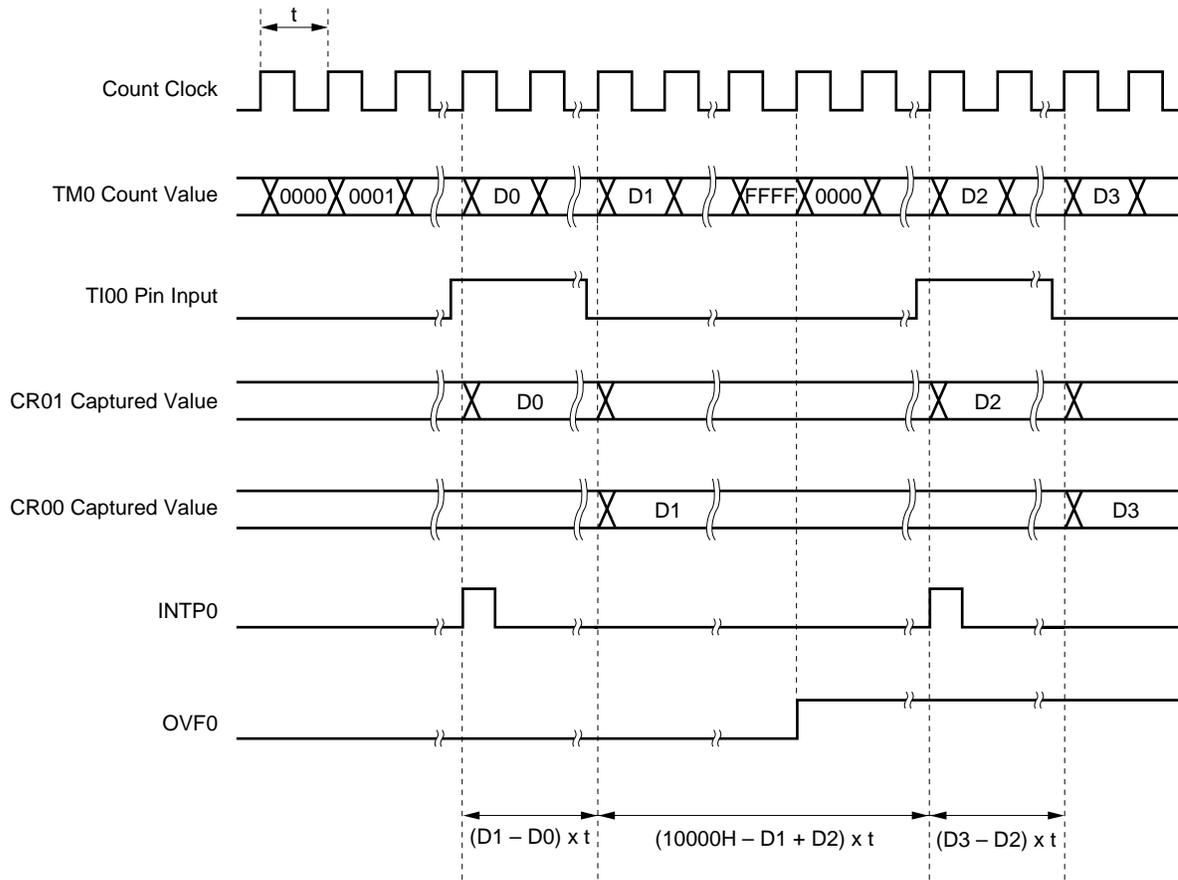
Caution If the valid edge of TI00/P00 is specified to be both rising and falling edge, capture/compare register 00 (CR00) cannot perform the capture operation.

Figure 8-22. Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See the description of the respective control registers for details.

Figure 8-23. Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified)



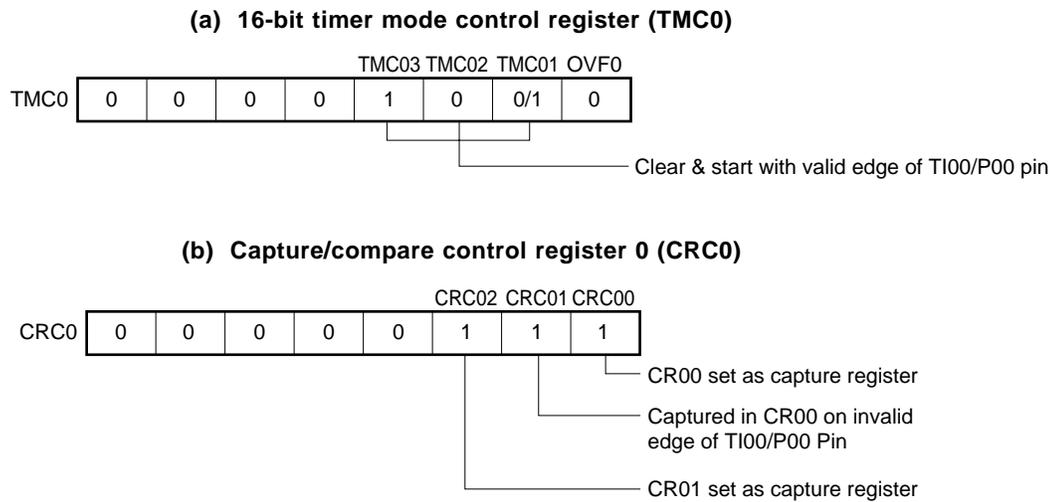
(4) Pulse width measurement by means of restart

When input of a valid edge to the TI00/P00 pin is detected, the count value of the 16-bit timer register (TM0) is taken into 16-bit capture/compare register 01 (CR01), and then the pulse width of the signal input to the TI00/P00 pin is measured by clearing TM0 and restarting the count (see register settings in Figure 8-24). The edge specification can be selected from two types, rising and falling edges, by bits 2 and 3 (ES10 and ES11) of the external interrupt mode register 0 (INTM0).

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

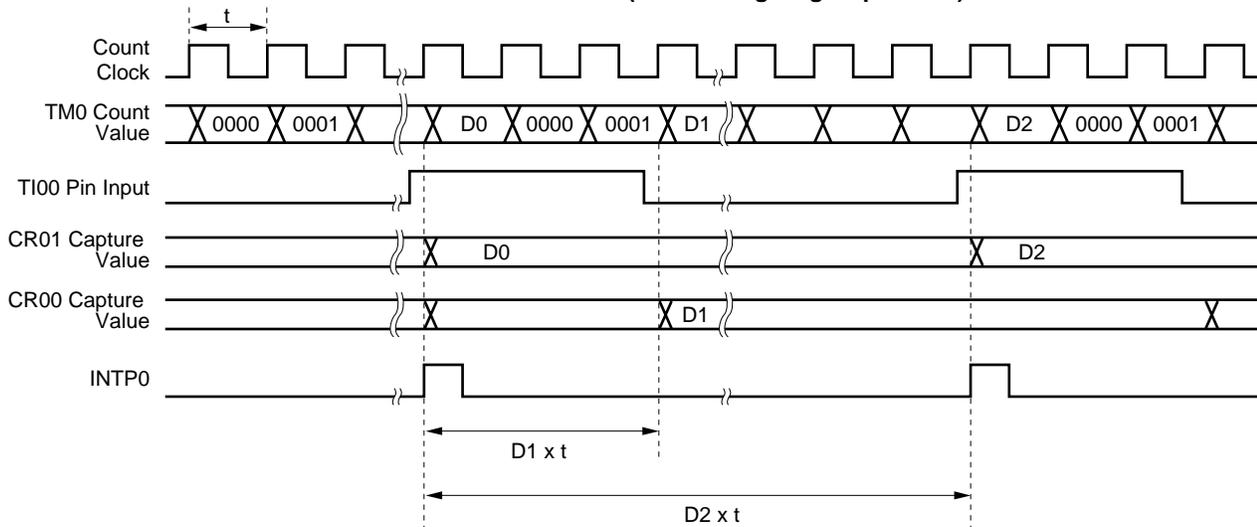
Caution If the valid edge of TI00/P00 is specified to be both rising and falling edge, capture/compare register 00 (CR00) cannot perform the capture operation.

Figure 8-24. Control Register Settings for Pulse Width Measurement by Means of Restart



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See the description of the respective control registers for details.

Figure 8-25. Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)



8.5.5 External event counter operation

The external event counter counts the number of external clock pulses to be input to the T100/P00 pin with the 16-bit timer register (TM0).

TM0 is incremented each time the valid edge specified with the external interrupt mode register 0 (INTM0) is input.

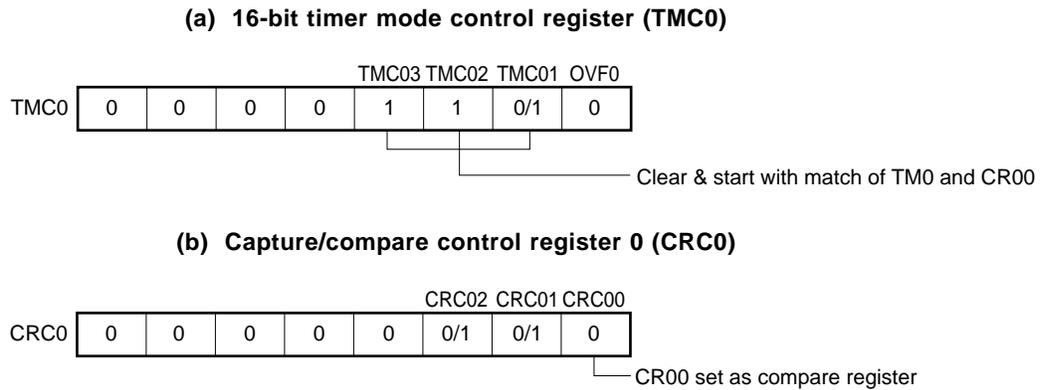
When the TM0 counted value matches the 16-bit capture/compare register 00 (CR00) value, TM0 is cleared to 0 and the interrupt request signal (INTTM00) is generated.

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

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

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

Figure 8-26. Control Register Settings in External Event Counter Mode



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the external event counter. See the description of the respective control registers for details.

Figure 8-27. External Event Counter Configuration Diagram

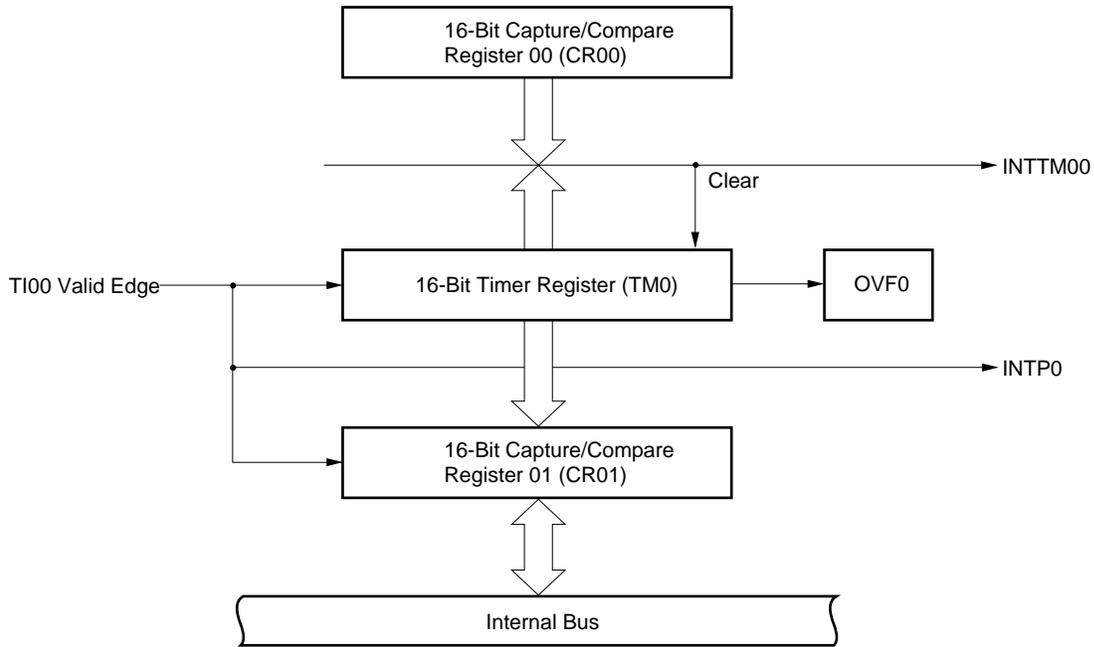
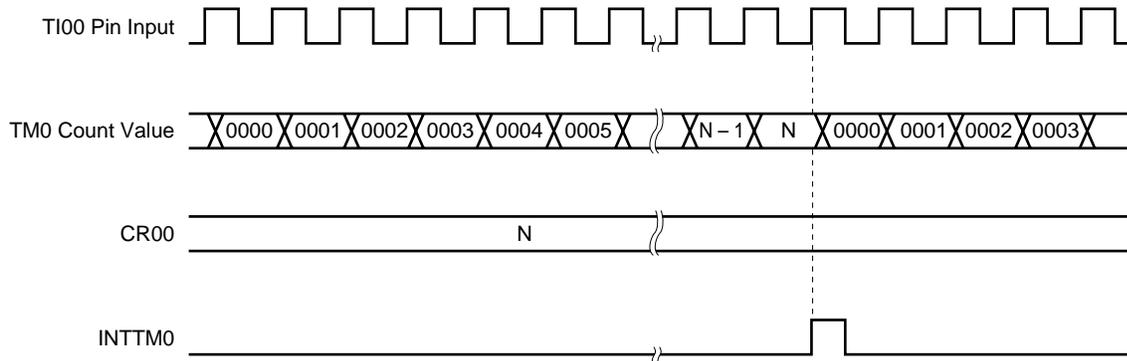


Figure 8-28. External Event Counter Operation Timings (with Rising Edge Specified)



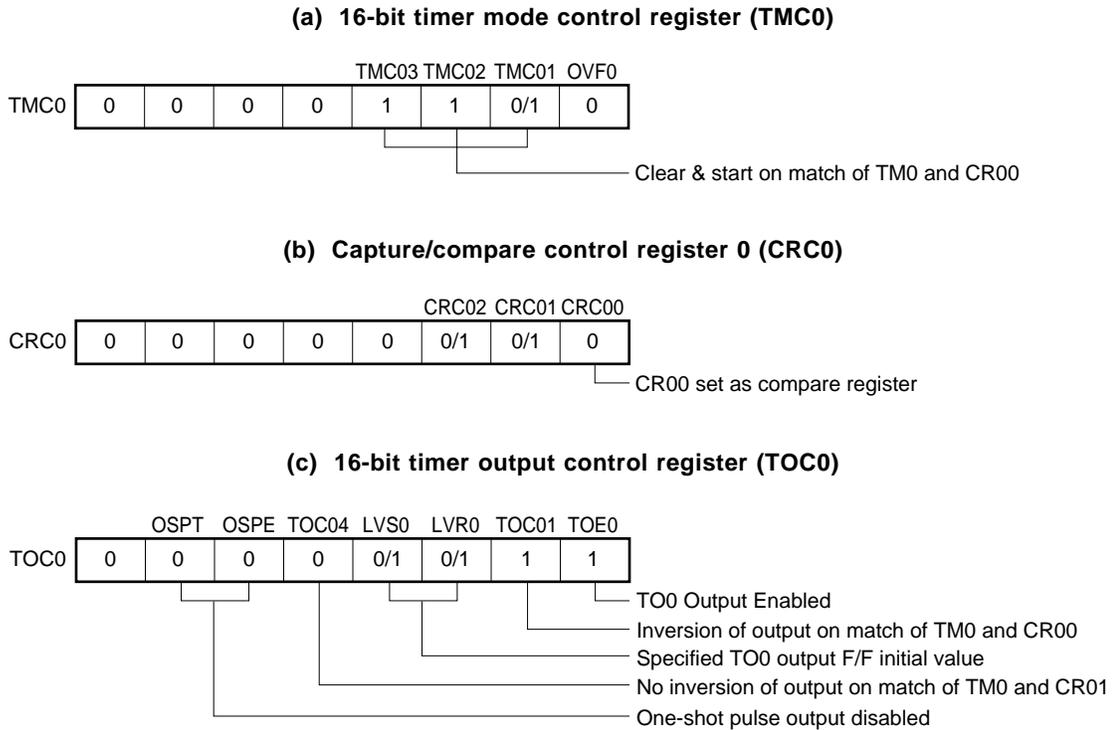
Caution When reading the external event counter count value, TM0 should be read.

8.5.6 Square-wave output operation

16-bit timer/event counter operates as a square wave output with any selected frequency at intervals of the count value preset to the 16-bit capture/compare register 00 (CR00).

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

Figure 8-29. Control Register Settings in Square-Wave Output Mode



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with square-wave output. See the description of the respective control registers for details.

Figure 8-30. Square-Wave Output Operation Timing

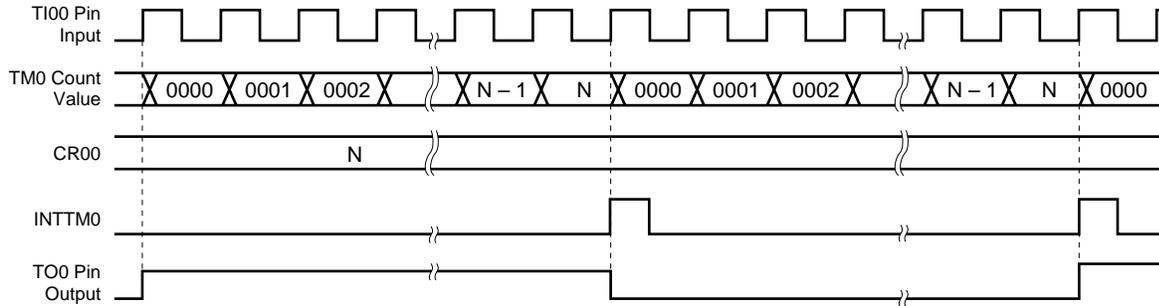


Table 8-7. 16-Bit Timer/Event Count Square-Wave Output Ranges

Minimum Pulse Width		Maximum Pulse Width		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
2 x TI00 input cycle		2 ¹⁶ x TI00 input cycle		TI00 input edge cycle	
—	2 x 1/f _x (400 ns)	—	2 ¹⁶ x 1/f _x (13.1 ms)	—	1/f _x (200 ns)
2 x 1/f _x (400 ns)	2 ² x 1/f _x (800 ns)	2 ¹⁶ x 1/f _x (13.1 ms)	2 ¹⁷ x 1/f _x (26.2 ms)	1/f _x (200 ns)	2 x 1/f _x (400 ns)
2 ² x 1/f _x (800 ns)	2 ³ x 1/f _x (1.6 μs)	2 ¹⁷ x 1/f _x (26.2 ms)	2 ¹⁸ x 1/f _x (52.4 ms)	2 x 1/f _x (400 ns)	2 ² x 1/f _x (800 ns)
2 ³ x 1/f _x (1.6 μs)	2 ⁴ x 1/f _x (3.2 μs)	2 ¹⁸ x 1/f _x (52.4 ms)	2 ¹⁹ x 1/f _x (104.9 ms)	2 ² x 1/f _x (800 ns)	2 ³ x 1/f _x (1.6 μs)
2 x watch timer output cycle		2 ¹⁶ x watch timer output cycle		Watch timer output edge cycle	

- Remarks**
1. f_x : Main system clock oscillation frequency
 2. MCS : Bit 0 of oscillation mode selection register (OSMS)
 3. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

8.5.7 One-shot pulse output operation

It is possible to output one-shot pulses synchronized with a software trigger or an external trigger (TI00/P00 pin input).

(1) One-shot pulse output using software trigger

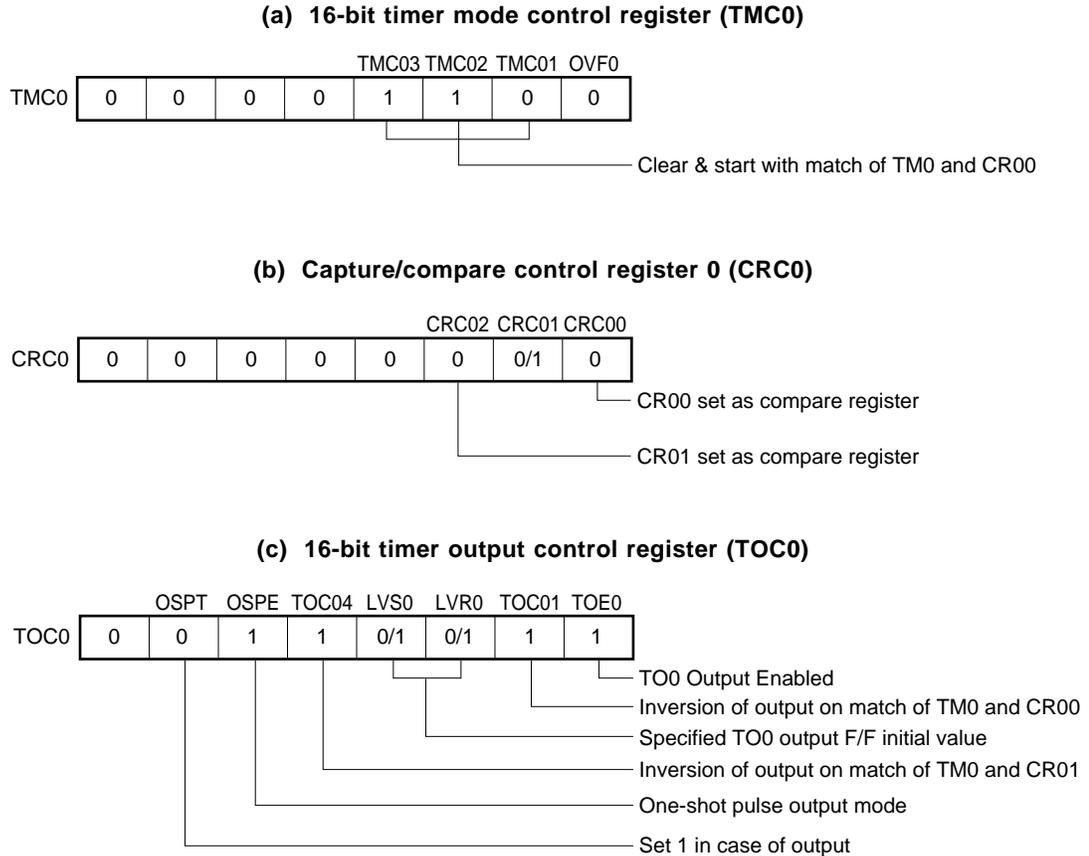
If the 16-bit timer mode control register (TMC0), capture/compare control register 0 (CRC0), and the 16-bit timer output control register (TOC0) are set as shown in Figure 8-31, and 1 is set in bit 6 (OSPT) of TOC0 by software, a one-shot pulse is output from the TO0/P30 pin.

By setting 1 in OSPT, the 16-bit timer/event counter is cleared and started, and output is activated by the count value set beforehand in 16-bit capture/compare register 01 (CR01). Thereafter, output is inactivated by the count value set beforehand in 16-bit capture/compare register 00 (CR00).

TM0 continues to operate after one-shot pulse is output. To stop TM0, 00H must be set to TMC0.

Caution When outputting one-shot pulse, do not set 1 in OSPT. When outputting one-shot pulse again, execute after the INTTM00, or interrupt match signal with CR00, is generated.

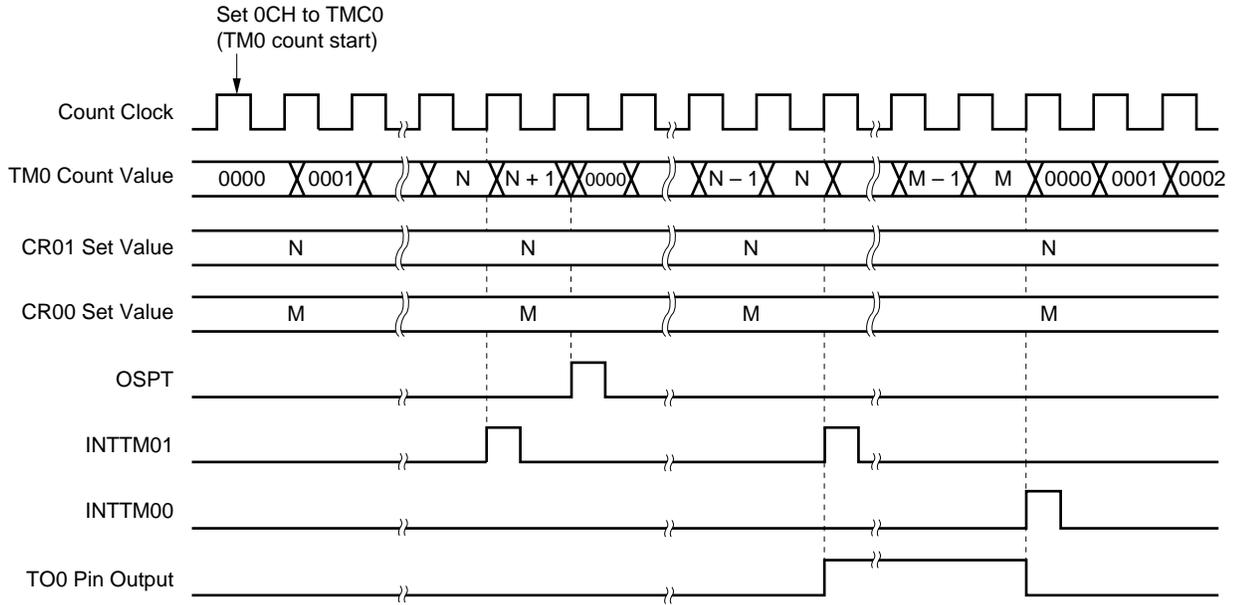
Figure 8-31. Control Register Settings for One-Shot Pulse Output Operation Using Software Trigger



Caution Values in the following range should be set in CR00 and CR01.
 $0000H \leq CR01 < CR00 \leq FFFFH$

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with one-shot pulse output. See the description of the respective control registers for details.

Figure 8-32. Timing of One-Shot Pulse Output Operation Using Software Trigger



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

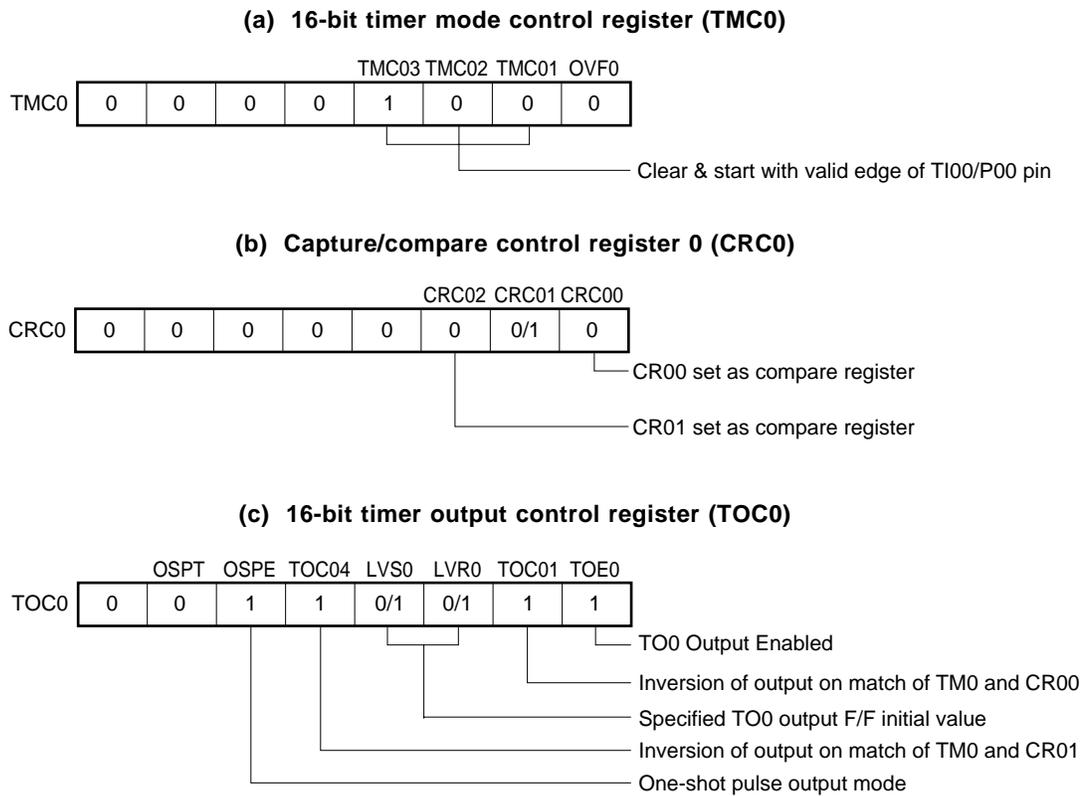
(2) One-shot pulse output using external trigger

If the 16-bit timer mode control register (TMC0), capture/compare control register 0 (CRC0), and the 16-bit timer output control register (TOC0) are set as shown in Figure 8-33, a one-shot pulse is output from the TO0/P30 pin with a TI00/P00 valid edge as an external trigger.

Any of three edge specifications can be selected-rising, falling, or both edges - as the valid edges for the TI00/P00 pin by means of bits 2 and 3 (ES10 and ES11) of external interrupt mode register 0 (INTM0). When a valid edge is input to the TI00/P00 pin, the 16-bit timer/event counter is cleared and started, and output is activated by the count values set beforehand in 16-bit capture/compare register 01 (CR01). Thereafter, output is inactivated by the count value set beforehand in 16-bit capture/compare register 00 (CR00).

Caution When outputting one-shot pulses, external trigger is ignored if generated again.

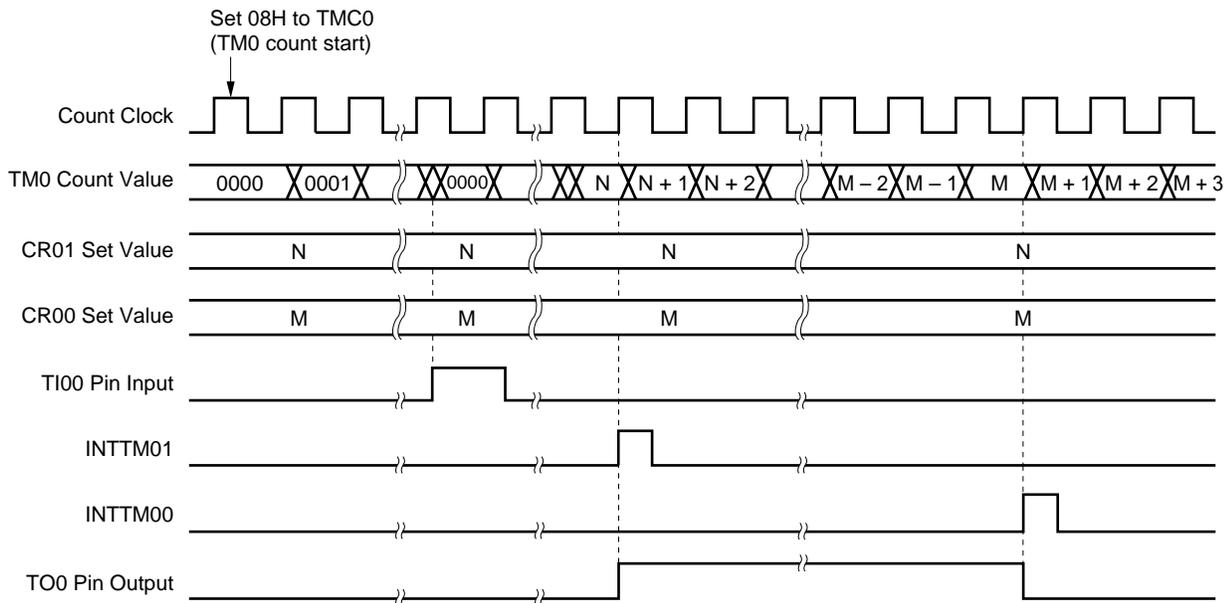
Figure 8-33. Control Register Settings for One-Shot Pulse Output Operation Using External Trigger



Caution Values in the following range should be set in CR00 and CR01.
 $0000H \leq CR01 < CR00 \leq FFFFH$

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with one-shot pulse output. See the description of the respective control registers for details.

Figure 8-34. Timing of One-Shot Pulse Output Operation Using External Trigger (With Rising Edge Specified)

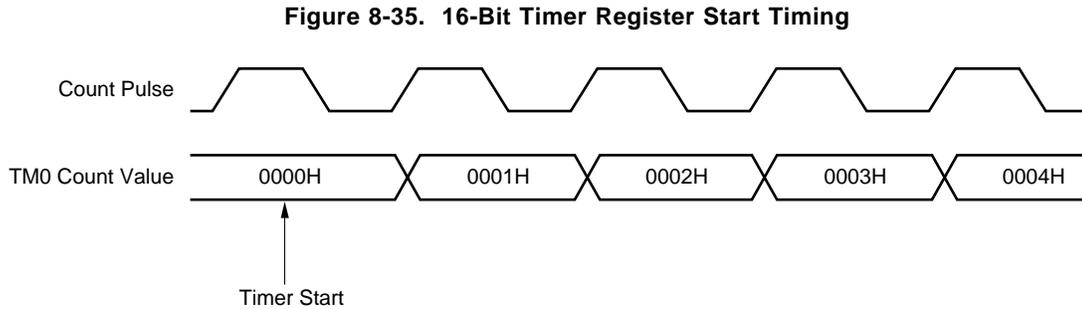


Caution The 16-bit timer register starts operation at the moment a value other than 0, 0, 0 (operation stop mode) is set to TMC01 to TMC03, respectively.

8.6 16-Bit Timer/Event Counter Operating Precautions

(1) Timer start errors

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



(2) 16-bit compare register setting

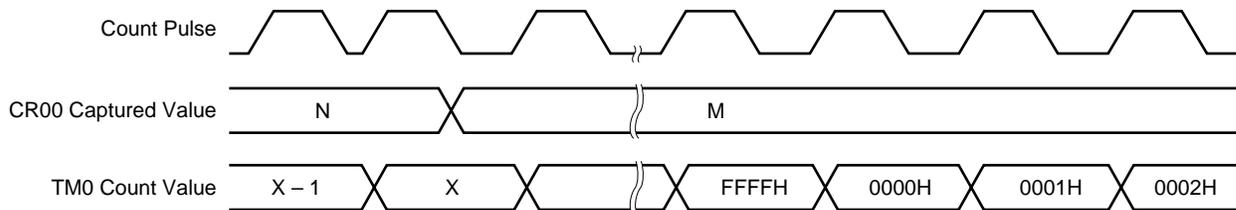
Set a value other than 0000H to the 16-bit capture/compare register 00 (CR00).

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

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

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

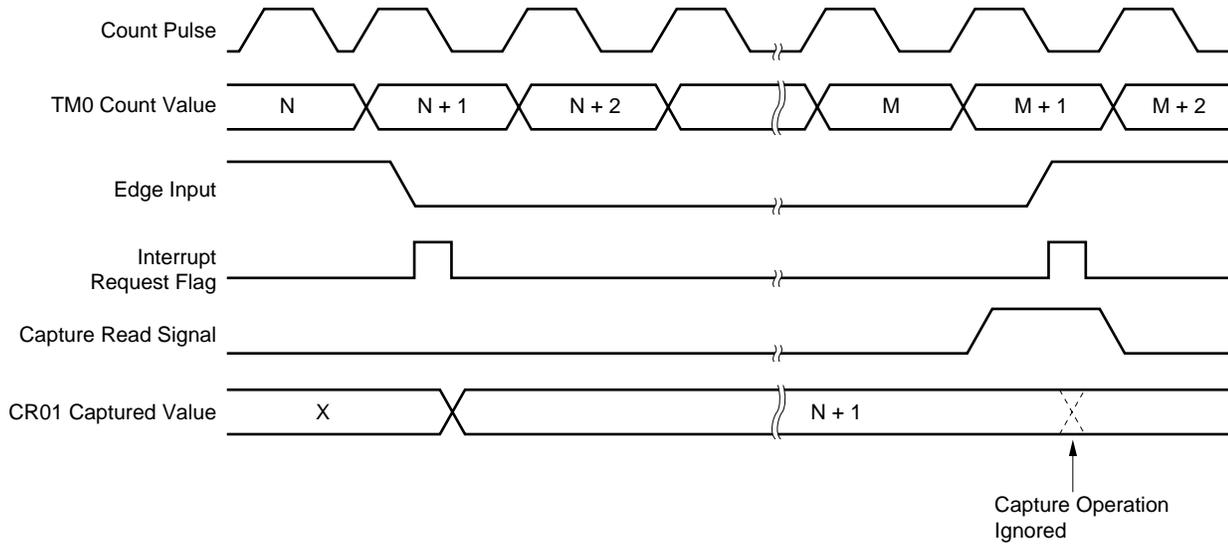
Figure 8-36. Timings After Change of Compare Register During Timer Count Operation



Remark $N > X > M$

(4) Capture register data retention timings

If the valid edge of the TI00/P00 pin is input during 16-bit capture/compare register 01 (CR01) read, CR01 holds data without carrying out capture operation. However, the interrupt request flag (PIF0) is set upon detection of the valid edge.

Figure 8-37. Capture Register Data Retention Timing**(5) Valid edge setting**

Set the valid edge of the TI00/P00/INTP0 pin after setting bits 1 to 3 (TMC01 to TMC03) of the 16-bit timer mode control register (TMC0) to 0, 0 and 0, respectively, and then stopping timer operation. Valid edge setting is carried out with bits 2 and 3 (ES10 and ES11) of the external interrupt mode register 0 (INTM0).

(6) Re-trigger of one-shot pulse**(a) One-shot pulse output using software**

When outputting one-shot pulse, do not set 1 in OSPT. When outputting one-shot pulse again, execute it after the INTTM00, which is the match interrupt request with CR00, is generated.

(b) One-shot pulse output using external trigger

When outputting one-shot pulses, external trigger is ignored if generated again.

(7) Operation of OVF0 flag

OVF0 flag is set to 1 in the following case.

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

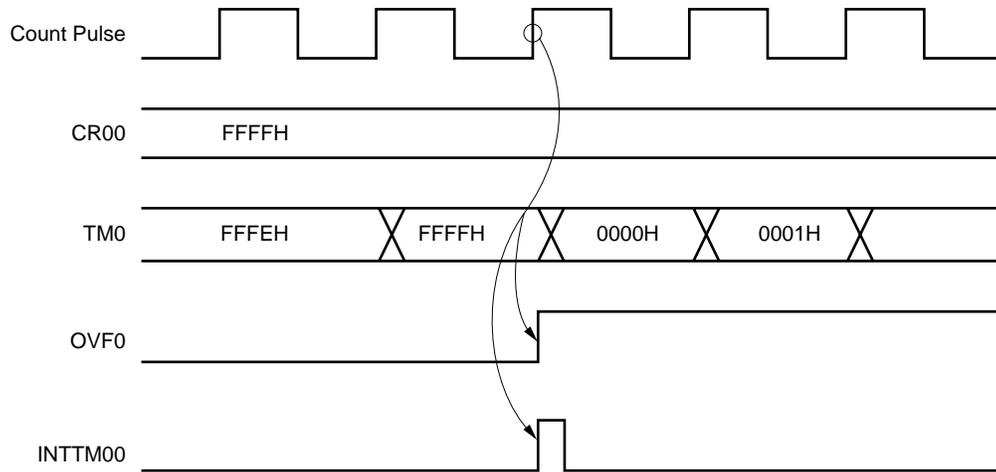


CR00 is set to FFFFH.



When TM0 is counted up from FFFFH to 0000H.

Figure 8-38. Operation Timing of OVF0 Flag



[MEMO]

CHAPTER 9 8-BIT TIMER/EVENT COUNTERS 1 AND 2

9.1 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).

9.1.1 8-bit timer/event counter mode

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

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

(1) 8-bit interval timer

Interrupt requests are generated at the preset time intervals.

Table 9-1. 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
$2 \times 1/f_x$ (400 ns)	$2^2 \times 1/f_x$ (800 ns)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)	$2 \times 1/f_x$ (400 ns)	$2^2 \times 1/f_x$ (800 ns)
$2^2 \times 1/f_x$ (800 ns)	$2^3 \times 1/f_x$ (1.6 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)	$2^{11} \times 1/f_x$ (409.6 μ s)	$2^2 \times 1/f_x$ (800 ns)	$2^3 \times 1/f_x$ (1.6 μ s)
$2^3 \times 1/f_x$ (1.6 μ s)	$2^4 \times 1/f_x$ (3.2 μ s)	$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)	$2^3 \times 1/f_x$ (1.6 μ s)	$2^4 \times 1/f_x$ (3.2 μ s)
$2^4 \times 1/f_x$ (3.2 μ s)	$2^5 \times 1/f_x$ (6.4 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)	$2^{13} \times 1/f_x$ (1.64 ms)	$2^4 \times 1/f_x$ (3.2 μ s)	$2^5 \times 1/f_x$ (6.4 μ s)
$2^5 \times 1/f_x$ (6.4 μ s)	$2^6 \times 1/f_x$ (12.8 μ s)	$2^{13} \times 1/f_x$ (1.64 ms)	$2^{14} \times 1/f_x$ (3.28 ms)	$2^5 \times 1/f_x$ (6.4 μ s)	$2^6 \times 1/f_x$ (12.8 μ s)
$2^6 \times 1/f_x$ (12.8 μ s)	$2^7 \times 1/f_x$ (25.6 μ s)	$2^{14} \times 1/f_x$ (3.28 ms)	$2^{15} \times 1/f_x$ (6.55 ms)	$2^6 \times 1/f_x$ (12.8 μ s)	$2^7 \times 1/f_x$ (25.6 μ s)
$2^7 \times 1/f_x$ (25.6 μ s)	$2^8 \times 1/f_x$ (51.2 μ s)	$2^{15} \times 1/f_x$ (6.55 ms)	$2^{16} \times 1/f_x$ (13.1 ms)	$2^7 \times 1/f_x$ (25.6 μ s)	$2^8 \times 1/f_x$ (51.2 μ s)
$2^8 \times 1/f_x$ (51.2 μ s)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{16} \times 1/f_x$ (13.1 ms)	$2^{17} \times 1/f_x$ (26.2 ms)	$2^8 \times 1/f_x$ (51.2 μ s)	$2^9 \times 1/f_x$ (102.4 μ s)
$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)	$2^{17} \times 1/f_x$ (26.2 ms)	$2^{18} \times 1/f_x$ (52.4 ms)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)
$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)	$2^{19} \times 1/f_x$ (104.9 ms)	$2^{20} \times 1/f_x$ (209.7 ms)	$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)

- Remarks**
1. f_x : Main system clock oscillation frequency
 2. MCS : Bit 0 of oscillation mode selection register (OSMS)
 3. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

(2) External event counter

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

(3) Square-wave output

A square wave with any selected frequency can be output.

Table 9-2. 8-Bit Timer/Event Counters 1 and 2 Square-Wave Output Ranges

Minimum Pulse Width		Maximum Pulse Width		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
$2 \times 1/f_x$ (400 ns)	$2^2 \times 1/f_x$ (800 ns)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)	$2 \times 1/f_x$ (400 ns)	$2^2 \times 1/f_x$ (800 ns)
$2^2 \times 1/f_x$ (800 ns)	$2^3 \times 1/f_x$ (1.6 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)	$2^{11} \times 1/f_x$ (409.6 μ s)	$2^2 \times 1/f_x$ (800 ns)	$2^3 \times 1/f_x$ (1.6 μ s)
$2^3 \times 1/f_x$ (1.6 μ s)	$2^4 \times 1/f_x$ (3.2 μ s)	$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)	$2^3 \times 1/f_x$ (1.6 μ s)	$2^4 \times 1/f_x$ (3.2 μ s)
$2^4 \times 1/f_x$ (3.2 μ s)	$2^5 \times 1/f_x$ (6.4 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)	$2^{13} \times 1/f_x$ (1.64 ms)	$2^4 \times 1/f_x$ (3.2 μ s)	$2^5 \times 1/f_x$ (6.4 μ s)
$2^5 \times 1/f_x$ (6.4 μ s)	$2^6 \times 1/f_x$ (12.8 μ s)	$2^{13} \times 1/f_x$ (1.64 ms)	$2^{14} \times 1/f_x$ (3.28 ms)	$2^5 \times 1/f_x$ (6.4 μ s)	$2^6 \times 1/f_x$ (12.8 μ s)
$2^6 \times 1/f_x$ (12.8 μ s)	$2^7 \times 1/f_x$ (25.6 μ s)	$2^{14} \times 1/f_x$ (3.28 ms)	$2^{15} \times 1/f_x$ (6.55 ms)	$2^6 \times 1/f_x$ (12.8 μ s)	$2^7 \times 1/f_x$ (25.6 μ s)
$2^7 \times 1/f_x$ (25.6 μ s)	$2^8 \times 1/f_x$ (51.2 μ s)	$2^{15} \times 1/f_x$ (6.55 ms)	$2^{16} \times 1/f_x$ (13.1 ms)	$2^7 \times 1/f_x$ (25.6 μ s)	$2^8 \times 1/f_x$ (51.2 μ s)
$2^8 \times 1/f_x$ (51.2 μ s)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{16} \times 1/f_x$ (13.1 ms)	$2^{17} \times 1/f_x$ (26.2 ms)	$2^8 \times 1/f_x$ (51.2 μ s)	$2^9 \times 1/f_x$ (102.4 μ s)
$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)	$2^{17} \times 1/f_x$ (26.2 ms)	$2^{18} \times 1/f_x$ (52.4 ms)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)
$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)	$2^{19} \times 1/f_x$ (104.9 ms)	$2^{20} \times 1/f_x$ (209.7 ms)	$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)

- Remarks**
1. f_x : Main system clock oscillation frequency
 2. MCS : Bit 0 of oscillation mode selection register (OSMS)
 3. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

9.1.2 16-bit timer/event counter mode

(1) 16-bit interval timer

Interrupt requests can be generated at the preset time intervals.

Table 9-3. 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
$2 \times 1/f_x$ (400 ns)	$2^2 \times 1/f_x$ (800 ns)	$2^{17} \times 1/f_x$ (26.2 ms)	$2^{18} \times 1/f_x$ (52.4 ms)	$2 \times 1/f_x$ (400 ns)	$2^2 \times 1/f_x$ (800 ns)
$2^2 \times 1/f_x$ (800 ns)	$2^3 \times 1/f_x$ (1.6 μ s)	$2^{18} \times 1/f_x$ (52.4 ms)	$2^{19} \times 1/f_x$ (104.9 ms)	$2^2 \times 1/f_x$ (800 ns)	$2^3 \times 1/f_x$ (1.6 μ s)
$2^3 \times 1/f_x$ (1.6 μ s)	$2^4 \times 1/f_x$ (3.2 μ s)	$2^{19} \times 1/f_x$ (104.9 ms)	$2^{20} \times 1/f_x$ (209.7 ms)	$2^3 \times 1/f_x$ (1.6 μ s)	$2^4 \times 1/f_x$ (3.2 μ s)
$2^4 \times 1/f_x$ (3.2 μ s)	$2^5 \times 1/f_x$ (6.4 μ s)	$2^{20} \times 1/f_x$ (209.7 ms)	$2^{21} \times 1/f_x$ (419.4 ms)	$2^4 \times 1/f_x$ (3.2 μ s)	$2^5 \times 1/f_x$ (6.4 μ s)
$2^5 \times 1/f_x$ (6.4 μ s)	$2^6 \times 1/f_x$ (12.8 μ s)	$2^{21} \times 1/f_x$ (419.4 ms)	$2^{22} \times 1/f_x$ (838.9 ms)	$2^5 \times 1/f_x$ (6.4 μ s)	$2^6 \times 1/f_x$ (12.8 μ s)
$2^6 \times 1/f_x$ (12.8 μ s)	$2^7 \times 1/f_x$ (25.6 μ s)	$2^{22} \times 1/f_x$ (838.9 ms)	$2^{23} \times 1/f_x$ (1.7 s)	$2^6 \times 1/f_x$ (12.8 μ s)	$2^7 \times 1/f_x$ (25.6 μ s)
$2^7 \times 1/f_x$ (25.6 μ s)	$2^8 \times 1/f_x$ (51.2 μ s)	$2^{23} \times 1/f_x$ (1.7 s)	$2^{24} \times 1/f_x$ (3.4 s)	$2^7 \times 1/f_x$ (25.6 μ s)	$2^8 \times 1/f_x$ (51.2 μ s)
$2^8 \times 1/f_x$ (51.2 μ s)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{24} \times 1/f_x$ (3.4 s)	$2^{25} \times 1/f_x$ (6.7 s)	$2^8 \times 1/f_x$ (51.2 μ s)	$2^9 \times 1/f_x$ (102.4 μ s)
$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)	$2^{25} \times 1/f_x$ (6.7 s)	$2^{26} \times 1/f_x$ (13.4 s)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)
$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)	$2^{27} \times 1/f_x$ (26.8 s)	$2^{28} \times 1/f_x$ (53.7 s)	$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)

- Remarks**
1. f_x : Main system clock oscillation frequency
 2. MCS : Bit 0 of oscillation mode selection register (OSMS)
 3. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

(2) External event counter

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

(3) Square-wave output

A square wave with any selected frequency can be output.

Table 9-4. Square-Wave Output Ranges when 8-Bit Timer/Event Counters 1 and 2 are Used as 16-Bit Timer/Event Counters

Minimum Pulse Width		Maximum Pulse Width		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
$2 \times 1/f_x$ (400 ns)	$2^2 \times 1/f_x$ (800 ns)	$2^{17} \times 1/f_x$ (26.2 ms)	$2^{18} \times 1/f_x$ (52.4 ms)	$2 \times 1/f_x$ (400 ns)	$2^2 \times 1/f_x$ (800 ns)
$2^2 \times 1/f_x$ (800 ns)	$2^3 \times 1/f_x$ (1.6 μ s)	$2^{18} \times 1/f_x$ (52.4 ms)	$2^{19} \times 1/f_x$ (104.9 ms)	$2^2 \times 1/f_x$ (800 ns)	$2^3 \times 1/f_x$ (1.6 μ s)
$2^3 \times 1/f_x$ (1.6 μ s)	$2^4 \times 1/f_x$ (3.2 μ s)	$2^{19} \times 1/f_x$ (104.9 ms)	$2^{20} \times 1/f_x$ (209.7 ms)	$2^3 \times 1/f_x$ (1.6 μ s)	$2^4 \times 1/f_x$ (3.2 μ s)
$2^4 \times 1/f_x$ (3.2 μ s)	$2^5 \times 1/f_x$ (6.4 μ s)	$2^{20} \times 1/f_x$ (209.7 ms)	$2^{21} \times 1/f_x$ (419.4 ms)	$2^4 \times 1/f_x$ (3.2 μ s)	$2^5 \times 1/f_x$ (6.4 μ s)
$2^5 \times 1/f_x$ (6.4 μ s)	$2^6 \times 1/f_x$ (12.8 μ s)	$2^{21} \times 1/f_x$ (419.4 ms)	$2^{22} \times 1/f_x$ (838.9 ms)	$2^5 \times 1/f_x$ (6.4 μ s)	$2^6 \times 1/f_x$ (12.8 μ s)
$2^6 \times 1/f_x$ (12.8 μ s)	$2^7 \times 1/f_x$ (25.6 μ s)	$2^{22} \times 1/f_x$ (838.9 ms)	$2^{23} \times 1/f_x$ (1.7 s)	$2^6 \times 1/f_x$ (12.8 μ s)	$2^7 \times 1/f_x$ (25.6 μ s)
$2^7 \times 1/f_x$ (25.6 μ s)	$2^8 \times 1/f_x$ (51.2 μ s)	$2^{23} \times 1/f_x$ (1.7 s)	$2^{24} \times 1/f_x$ (3.4 s)	$2^7 \times 1/f_x$ (25.6 μ s)	$2^8 \times 1/f_x$ (51.2 μ s)
$2^8 \times 1/f_x$ (51.2 μ s)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{24} \times 1/f_x$ (3.4 s)	$2^{25} \times 1/f_x$ (6.7 s)	$2^8 \times 1/f_x$ (51.2 μ s)	$2^9 \times 1/f_x$ (102.4 μ s)
$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)	$2^{25} \times 1/f_x$ (6.7 s)	$2^{26} \times 1/f_x$ (13.4 s)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)
$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)	$2^{27} \times 1/f_x$ (26.8 s)	$2^{28} \times 1/f_x$ (53.7 s)	$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)

Remarks 1. f_x : Main system clock oscillation frequency

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

3. Figures in parentheses apply to operation with at $f_x = 5.0$ MHz.

9.2 8-Bit Timer/Event Counters 1 and 2 Configurations

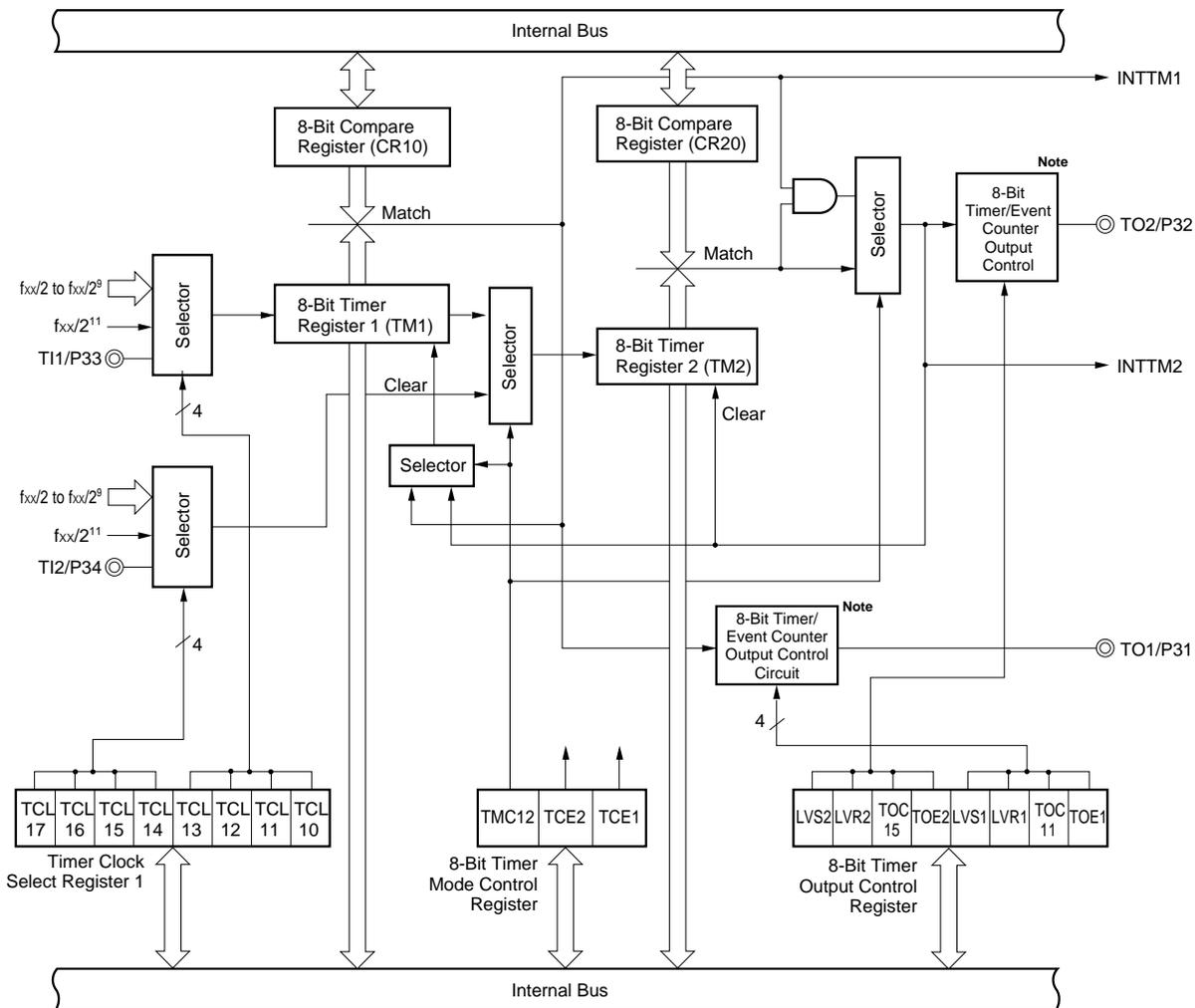
The 8-bit timer/event counters 1 and 2 consist of the following hardware.

Table 9-5. 8-Bit Timer/Event Counters 1 and 2 Configurations

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

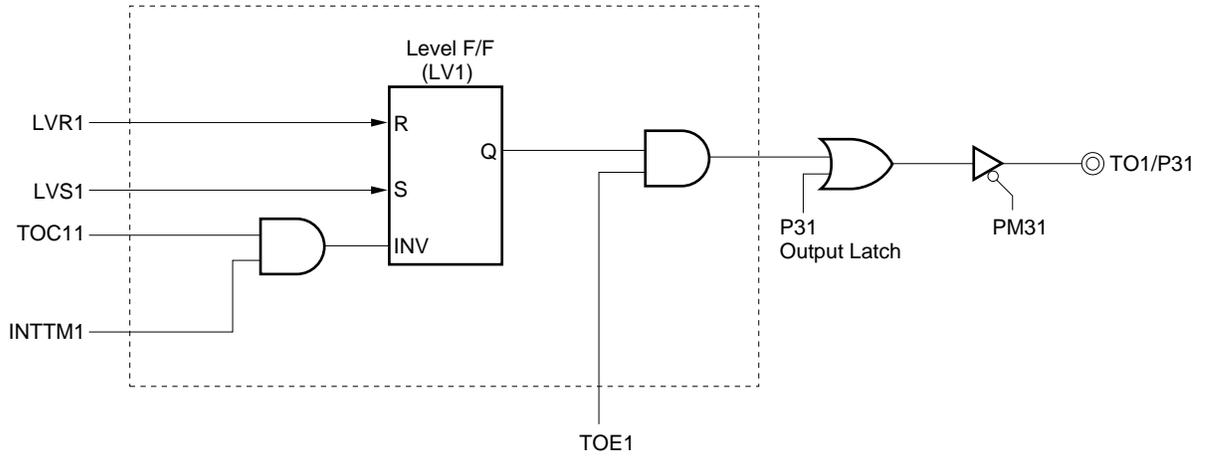
Note Refer to **Figure 6-9 Block Diagram of P30 to P37**.

Figure 9-1. 8-Bit Timer/Event Counters 1 and 2 Block Diagram



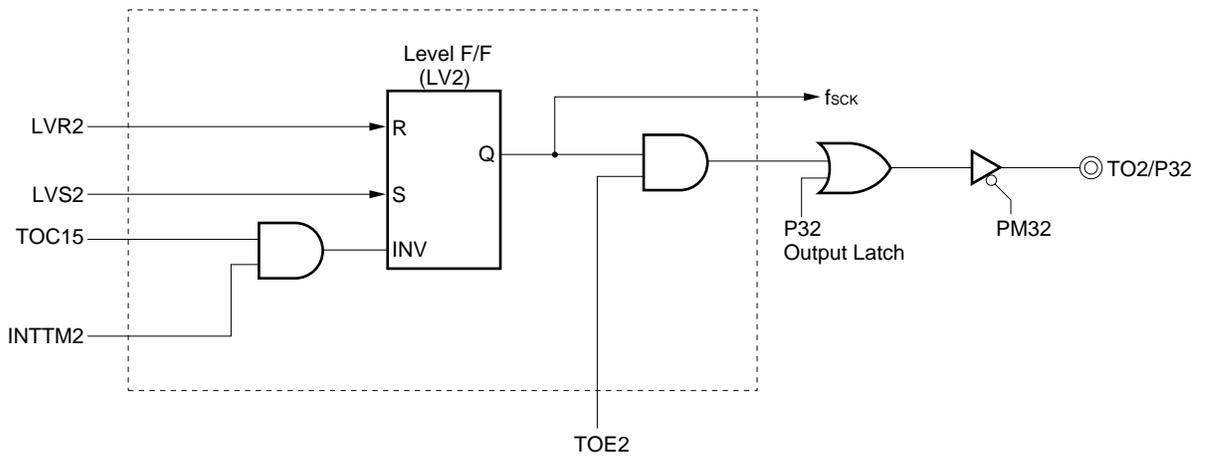
Note Refer to **Figures 9-2 and 9-3** for details of 8-bit timer/event counter output control circuits 1 and 2, respectively.

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



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

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



Remarks 1. The section in the broken line is an output control circuit.
 2. f_{sck} : Serial clock frequency

(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.

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

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

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

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

9.3 8-Bit Timer/Event Counters 1 and 2 Control Registers

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

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

(1) Timer clock select register 1 (TCL1)

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

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

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

Figure 9-4. Timer Clock Select Register 1 Format

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

TCL13				TCL12				TCL11				TCL10				8-Bit Timer Register 1 Count Clock Selection							
																MCS = 1				MCS = 0			
0	0	0	0	TI1 falling edge																			
0	0	0	1	TI1 rising edge																			
0	1	1	0	$f_{xx}/2$	$f_x/2$ (2.5 MHz)				$f_x/2^2$ (1.25 MHz)														
0	1	1	1	$f_{xx}/2^2$	$f_x/2^2$ (1.25 MHz)				$f_x/2^3$ (625 kHz)														
1	0	0	0	$f_{xx}/2^3$	$f_x/2^3$ (625 kHz)				$f_x/2^4$ (313 kHz)														
1	0	0	1	$f_{xx}/2^4$	$f_x/2^4$ (313 kHz)				$f_x/2^5$ (156 kHz)														
1	0	1	0	$f_{xx}/2^5$	$f_x/2^5$ (156 kHz)				$f_x/2^6$ (78.1 kHz)														
1	0	1	1	$f_{xx}/2^6$	$f_x/2^6$ (78.1 kHz)				$f_x/2^7$ (39.1 kHz)														
1	1	0	0	$f_{xx}/2^7$	$f_x/2^7$ (39.1 kHz)				$f_x/2^8$ (19.5 kHz)														
1	1	0	1	$f_{xx}/2^8$	$f_x/2^8$ (19.5 kHz)				$f_x/2^9$ (9.8 kHz)														
1	1	1	0	$f_{xx}/2^9$	$f_x/2^9$ (9.8 kHz)				$f_x/2^{10}$ (4.9 kHz)														
1	1	1	1	$f_{xx}/2^{11}$	$f_x/2^{11}$ (2.4 kHz)				$f_x/2^{12}$ (1.2 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	$f_{xx}/2$	$f_x/2$ (2.5 MHz)				$f_x/2^2$ (1.25 MHz)														
0	1	1	1	$f_{xx}/2^2$	$f_x/2^2$ (1.25 MHz)				$f_x/2^3$ (625 kHz)														
1	0	0	0	$f_{xx}/2^3$	$f_x/2^3$ (625 kHz)				$f_x/2^4$ (313 kHz)														
1	0	0	1	$f_{xx}/2^4$	$f_x/2^4$ (313 kHz)				$f_x/2^5$ (156 kHz)														
1	0	1	0	$f_{xx}/2^5$	$f_x/2^5$ (156 kHz)				$f_x/2^6$ (78.1 kHz)														
1	0	1	1	$f_{xx}/2^6$	$f_x/2^6$ (78.1 kHz)				$f_x/2^7$ (39.1 kHz)														
1	1	0	0	$f_{xx}/2^7$	$f_x/2^7$ (39.1 kHz)				$f_x/2^8$ (19.5 kHz)														
1	1	0	1	$f_{xx}/2^8$	$f_x/2^8$ (19.5 kHz)				$f_x/2^9$ (9.8 kHz)														
1	1	1	0	$f_{xx}/2^9$	$f_x/2^9$ (9.8 kHz)				$f_x/2^{10}$ (4.9 kHz)														
1	1	1	1	$f_{xx}/2^{11}$	$f_x/2^{11}$ (2.4 kHz)				$f_x/2^{12}$ (1.2 kHz)														
Other than above				Setting prohibited																			

Caution Stop the timer operation before rewriting TCL1 to other data.

- Remarks**
1. f_{xx} : Main system clock frequency (f_x or $f_x/2$)
 2. f_x : Main system clock oscillation frequency
 3. TI1 : 8-bit timer register 1 input pin
 4. TI2 : 8-bit timer register 2 input pin
 5. MCS : Bit 0 of oscillation mode selection register (OSMS)
 6. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

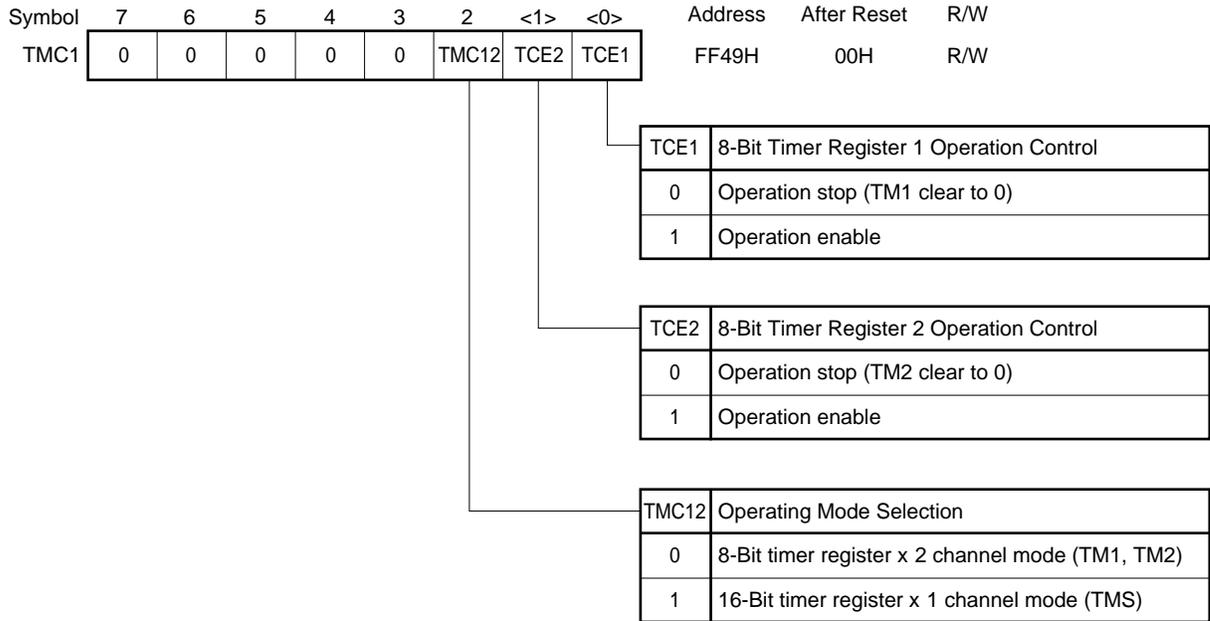
(2) 8-bit timer mode control register 1 (TMC1)

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 9-5. 8-Bit Timer Mode Control Register 1 Format



- Cautions**
1. Stop the timer operation before switching the operating mode.
 2. When used as 16-bit timer register, TCE1 should be used for operation enable/stop.

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

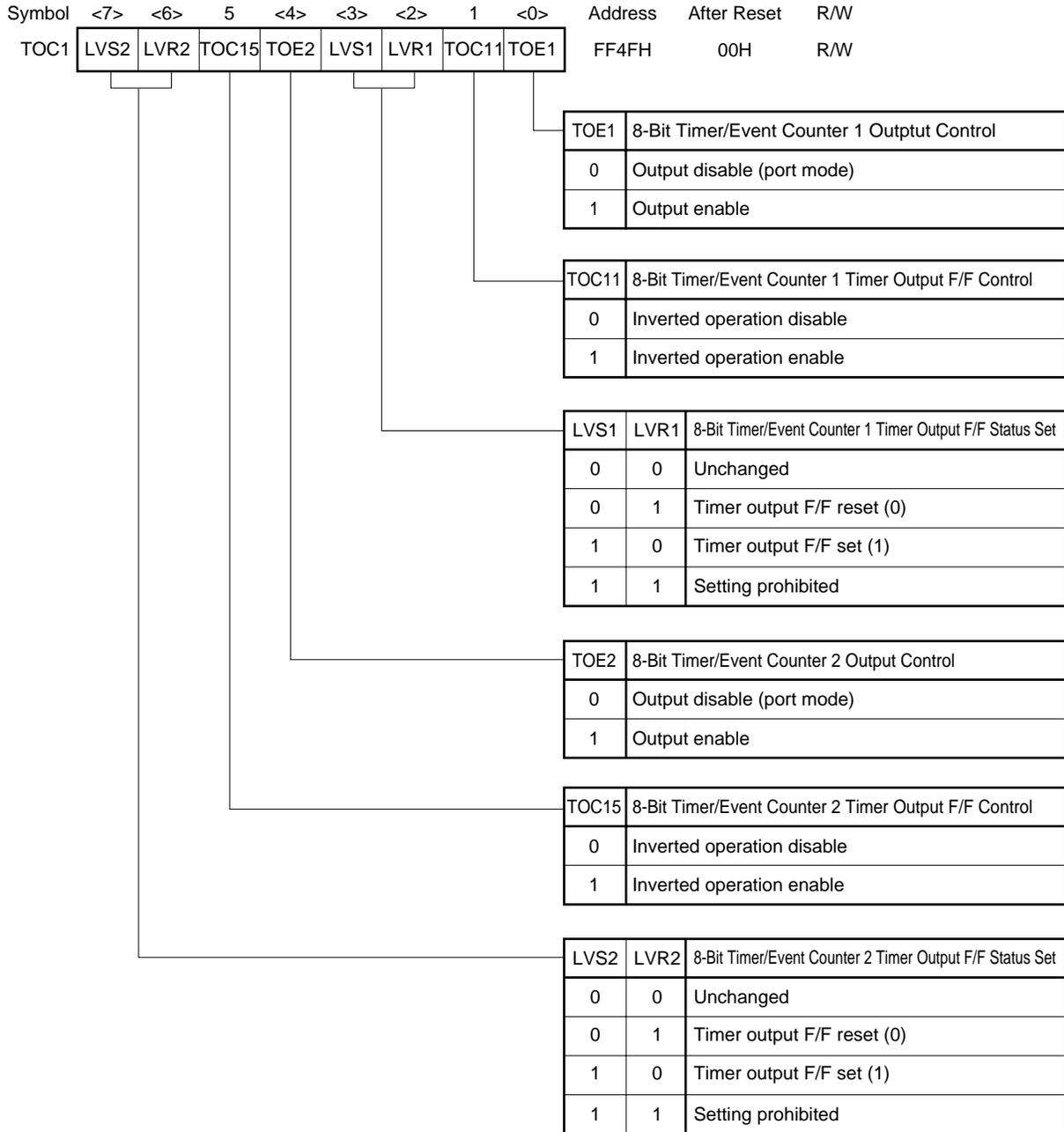
This register controls operation of 8-bit timer/event counter output control circuits 1 and 2.

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

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

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

Figure 9-6. 8-Bit Timer Output Control Register Format



Cautions 1. Stop the timer operation before setting TOC1.

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

(4) Port mode register 3 (PM3)

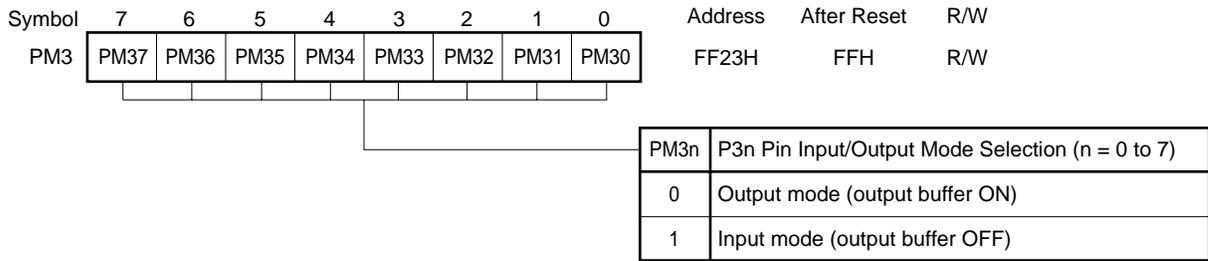
This register sets port 3 input/output in 1-bit units.

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

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

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

Figure 9-7. Port Mode Register 3 Format



9.4 8-Bit Timer/Event Counters 1 and 2 Operations

9.4.1 8-bit timer/event counter mode

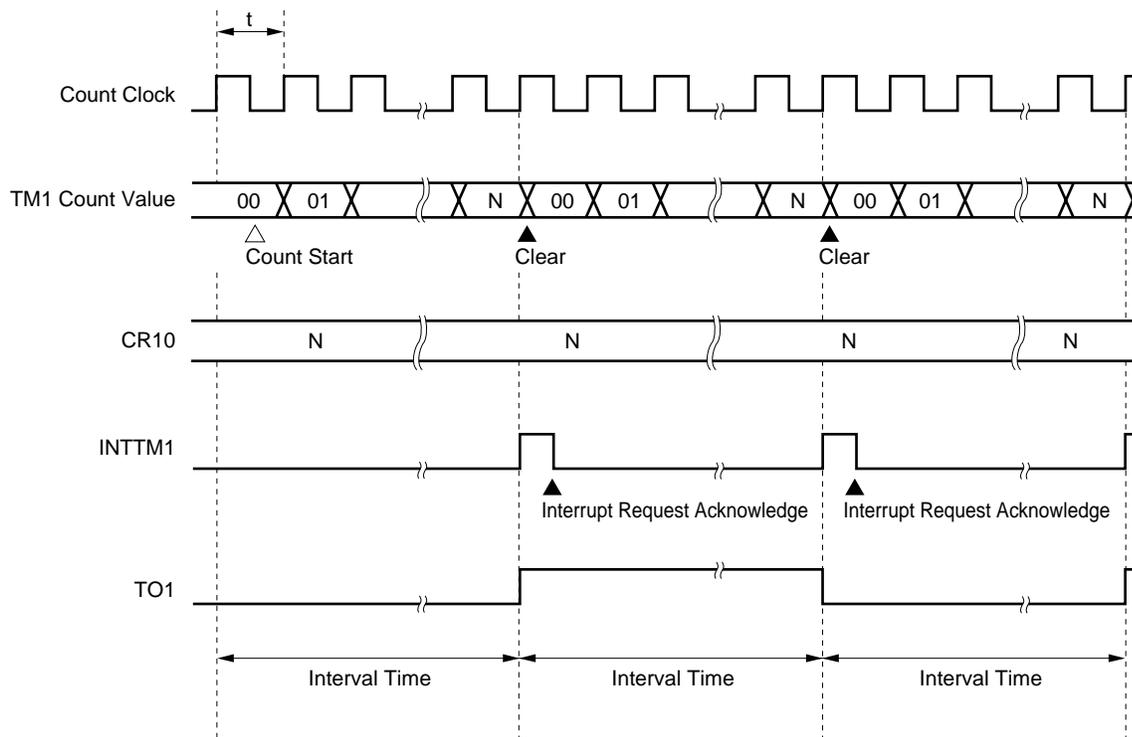
(1) Interval timer operations

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 TM1 can be selected with bits 0 to 3 (TCL10 to TCL13) of the timer clock select register 1 (TCL1). Count clock of the TM2 can be selected with bits 4 to 7 (TCL14 to TCL17) of the timer clock select register 1 (TCL1).

For the operation in the case the value of the compare register is changed during the timer count operation, refer to **9.5 8-Bit Timer/Event Counters 1 and 2 Precautions (3)**.

Figure 9-8. Interval Timer Operation Timing



Remark Interval time = $(N + 1) \times t$: $N = 00H$ to FFH

Table 9-6. 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	T11 input cycle		$2^8 \times$ T11 input cycle		T11 input edge cycle	
0	0	0	1	T11 input cycle		$2^8 \times$ T11 input cycle		T11 input edge cycle	
0	1	1	0	$2 \times 1/f_x$ (400 ns)	$2^2 \times 1/f_x$ (800 ns)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)	$2 \times 1/f_x$ (400 ns)	$2^2 \times 1/f_x$ (800 ns)
0	1	1	1	$2^2 \times 1/f_x$ (800 ns)	$2^3 \times 1/f_x$ (1.6 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)	$2^{11} \times 1/f_x$ (409.6 μ s)	$2^2 \times 1/f_x$ (800 ns)	$2^3 \times 1/f_x$ (1.6 μ s)
1	0	0	0	$2^3 \times 1/f_x$ (1.6 μ s)	$2^4 \times 1/f_x$ (3.2 μ s)	$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)	$2^3 \times 1/f_x$ (1.6 μ s)	$2^4 \times 1/f_x$ (3.2 μ s)
1	0	0	1	$2^4 \times 1/f_x$ (3.2 μ s)	$2^5 \times 1/f_x$ (6.4 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)	$2^{13} \times 1/f_x$ (1.64 ms)	$2^4 \times 1/f_x$ (3.2 μ s)	$2^5 \times 1/f_x$ (6.4 μ s)
1	0	1	0	$2^5 \times 1/f_x$ (6.4 μ s)	$2^6 \times 1/f_x$ (12.8 μ s)	$2^{13} \times 1/f_x$ (1.64 ms)	$2^{14} \times 1/f_x$ (3.28 ms)	$2^5 \times 1/f_x$ (6.4 μ s)	$2^6 \times 1/f_x$ (12.8 μ s)
1	0	1	1	$2^6 \times 1/f_x$ (12.8 μ s)	$2^7 \times 1/f_x$ (25.6 μ s)	$2^{14} \times 1/f_x$ (3.28 ms)	$2^{15} \times 1/f_x$ (6.55 ms)	$2^6 \times 1/f_x$ (12.8 μ s)	$2^7 \times 1/f_x$ (25.6 μ s)
1	1	0	0	$2^7 \times 1/f_x$ (25.6 μ s)	$2^8 \times 1/f_x$ (51.2 μ s)	$2^{15} \times 1/f_x$ (6.55 ms)	$2^{16} \times 1/f_x$ (13.1 ms)	$2^7 \times 1/f_x$ (25.6 μ s)	$2^8 \times 1/f_x$ (51.2 μ s)
1	1	0	1	$2^8 \times 1/f_x$ (51.2 μ s)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{16} \times 1/f_x$ (13.1 ms)	$2^{17} \times 1/f_x$ (26.2 ms)	$2^8 \times 1/f_x$ (51.2 μ s)	$2^9 \times 1/f_x$ (102.4 μ s)
1	1	1	0	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)	$2^{17} \times 1/f_x$ (26.2 ms)	$2^{18} \times 1/f_x$ (52.4 ms)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)
1	1	1	1	$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)	$2^{19} \times 1/f_x$ (104.9 ms)	$2^{20} \times 1/f_x$ (209.7 ms)	$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)
Other than above				Setting prohibited					

- Remarks**
1. f_x : Main system clock oscillation frequency
 2. MCS : Bit 0 of oscillation mode selection register (OSMS)
 3. TCL10 to TCL13 : Bits 0 to 3 of timer clock select register 1 (TCL1)
 4. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

Table 9-7. 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	Tl2 input cycle		2 ⁸ x Tl2 input cycle		Tl2 input edge cycle	
0	0	0	1	Tl2 input cycle		2 ⁸ x Tl2 input cycle		Tl2 input edge cycle	
0	1	1	0	2 x 1/f _x (400 ns)	2 ² x 1/f _x (800 ns)	2 ⁹ x 1/f _x (102.4 μs)	2 ¹⁰ x 1/f _x (204.8 μs)	2 x 1/f _x (400 ns)	2 ² x 1/f _x (800 ns)
0	1	1	1	2 ² x 1/f _x (800 ns)	2 ³ x 1/f _x (1.6 μs)	2 ¹⁰ x 1/f _x (204.8 μs)	2 ¹¹ x 1/f _x (409.6 μs)	2 ² x 1/f _x (800 ns)	2 ³ x 1/f _x (1.6 μs)
1	0	0	0	2 ³ x 1/f _x (1.6 μs)	2 ⁴ x 1/f _x (3.2 μs)	2 ¹¹ x 1/f _x (409.6 μs)	2 ¹² x 1/f _x (819.2 μs)	2 ³ x 1/f _x (1.6 μs)	2 ⁴ x 1/f _x (3.2 μs)
1	0	0	1	2 ⁴ x 1/f _x (3.2 μs)	2 ⁵ x 1/f _x (6.4 μs)	2 ¹² x 1/f _x (819.2 μs)	2 ¹³ x 1/f _x (1.64 ms)	2 ⁴ x 1/f _x (3.2 μs)	2 ⁵ x 1/f _x (6.4 μs)
1	0	1	0	2 ⁵ x 1/f _x (6.4 μs)	2 ⁶ x 1/f _x (12.8 μs)	2 ¹³ x 1/f _x (1.64 ms)	2 ¹⁴ x 1/f _x (3.28 ms)	2 ⁵ x 1/f _x (6.4 μs)	2 ⁶ x 1/f _x (12.8 μs)
1	0	1	1	2 ⁶ x 1/f _x (12.8 μs)	2 ⁷ x 1/f _x (25.6 μs)	2 ¹⁴ x 1/f _x (3.28 ms)	2 ¹⁵ x 1/f _x (6.55 ms)	2 ⁶ x 1/f _x (12.8 μs)	2 ⁷ x 1/f _x (25.6 μs)
1	1	0	0	2 ⁷ x 1/f _x (25.6 μs)	2 ⁸ x 1/f _x (51.2 μs)	2 ¹⁵ x 1/f _x (6.55 ms)	2 ¹⁶ x 1/f _x (13.1 ms)	2 ⁷ x 1/f _x (25.6 μs)	2 ⁸ x 1/f _x (51.2 μs)
1	1	0	1	2 ⁸ x 1/f _x (51.2 μs)	2 ⁹ x 1/f _x (102.4 μs)	2 ¹⁶ x 1/f _x (13.1 ms)	2 ¹⁷ x 1/f _x (26.2 ms)	2 ⁸ x 1/f _x (51.2 μs)	2 ⁹ x 1/f _x (102.4 μs)
1	1	1	0	2 ⁹ x 1/f _x (102.4 μs)	2 ¹⁰ x 1/f _x (204.8 μs)	2 ¹⁷ x 1/f _x (26.2 ms)	2 ¹⁸ x 1/f _x (52.4 ms)	2 ⁹ x 1/f _x (102.4 μs)	2 ¹⁰ x 1/f _x (204.8 μs)
1	1	1	1	2 ¹¹ x 1/f _x (409.6 μs)	2 ¹² x 1/f _x (819.2 μs)	2 ¹⁹ x 1/f _x (104.9 ms)	2 ²⁰ x 1/f _x (209.7 ms)	2 ¹¹ x 1/f _x (409.6 μs)	2 ¹² x 1/f _x (819.2 μs)
Other than above				Setting prohibited					

- Remarks**
1. f_x : Main system clock oscillation frequency
 2. MCS : Bit 0 of oscillation mode selection register (OSMS)
 3. TCL14 to TCL17 : Bits 4 to 7 timer clock select register 1 (TCL1)
 4. Figures in parentheses apply to operation with f_x = 5.0 MHz.

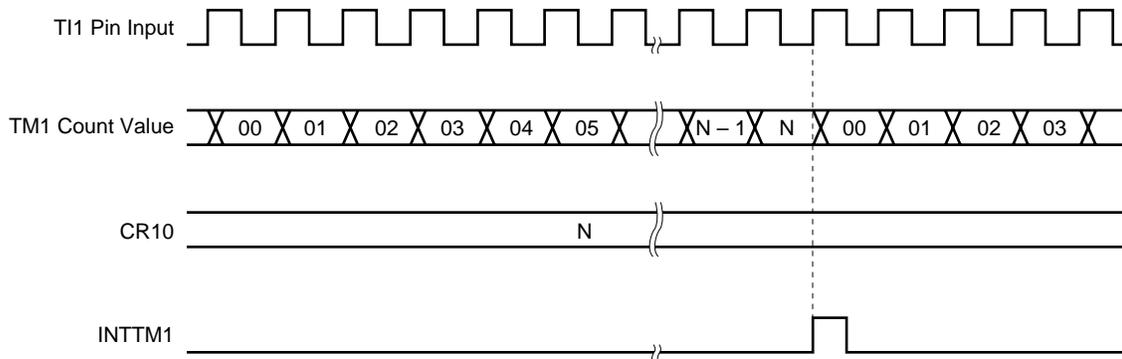
(2) External event counter operation

The external event counter counts the number of external clock pulses to be input to the T11/P33 and T12/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 9-9. External Event Counter Operation Timings (with Rising Edge Specified)



Remark N = 00H to FFH

(3) Square-wave output Operation

8-bit timer/event counters 1 and 2 operate as a square wave outputs with any selected frequency at intervals of the value preset to 8-bit compare registers 10 and 20 (CR10 and CR20).

The TO1/P31 or TO2/P32 pin output status is reversed at intervals of the count value preset to CR10 or CR20 by setting bit 0 (TOE1) or bit 4 (TOE2) of the 8-bit timer output control register (TOC1) to 1. This enables a square wave with any selected frequency to be output.

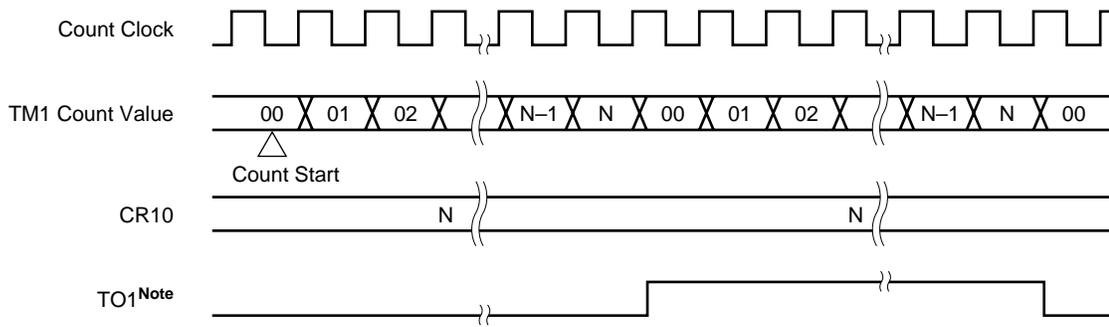
Table 9-8. 8-Bit Timer/Event Counters 1 and 2 Square-Wave Output Ranges

Minimum Pulse Width		Maximum Pulse Width		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
$2 \times 1/f_x$ (400 ns)	$2^2 \times 1/f_x$ (800 ns)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)	$2 \times 1/f_x$ (400 ns)	$2^2 \times 1/f_x$ (800 ns)
$2^2 \times 1/f_x$ (800 ns)	$2^3 \times 1/f_x$ (1.6 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)	$2^{11} \times 1/f_x$ (409.6 μ s)	$2^2 \times 1/f_x$ (800 ns)	$2^3 \times 1/f_x$ (1.6 μ s)
$2^3 \times 1/f_x$ (1.6 μ s)	$2^4 \times 1/f_x$ (3.2 μ s)	$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)	$2^3 \times 1/f_x$ (1.6 μ s)	$2^4 \times 1/f_x$ (3.2 μ s)
$2^4 \times 1/f_x$ (3.2 μ s)	$2^5 \times 1/f_x$ (6.4 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)	$2^{13} \times 1/f_x$ (1.64 ms)	$2^4 \times 1/f_x$ (3.2 μ s)	$2^5 \times 1/f_x$ (6.4 μ s)
$2^5 \times 1/f_x$ (6.4 μ s)	$2^6 \times 1/f_x$ (12.8 μ s)	$2^{13} \times 1/f_x$ (1.64 ms)	$2^{14} \times 1/f_x$ (3.28 ms)	$2^5 \times 1/f_x$ (6.4 μ s)	$2^6 \times 1/f_x$ (12.8 μ s)
$2^6 \times 1/f_x$ (12.8 μ s)	$2^7 \times 1/f_x$ (25.6 μ s)	$2^{14} \times 1/f_x$ (3.28 ms)	$2^{15} \times 1/f_x$ (6.55 ms)	$2^6 \times 1/f_x$ (12.8 μ s)	$2^7 \times 1/f_x$ (25.6 μ s)
$2^7 \times 1/f_x$ (25.6 μ s)	$2^8 \times 1/f_x$ (51.2 μ s)	$2^{15} \times 1/f_x$ (6.55 ms)	$2^{16} \times 1/f_x$ (13.1 ms)	$2^7 \times 1/f_x$ (25.6 μ s)	$2^8 \times 1/f_x$ (51.2 μ s)
$2^8 \times 1/f_x$ (51.2 μ s)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{16} \times 1/f_x$ (13.1 ms)	$2^{17} \times 1/f_x$ (26.2 ms)	$2^8 \times 1/f_x$ (51.2 μ s)	$2^9 \times 1/f_x$ (102.4 μ s)
$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)	$2^{17} \times 1/f_x$ (26.2 ms)	$2^{18} \times 1/f_x$ (52.4 ms)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)
$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)	$2^{19} \times 1/f_x$ (104.9 ms)	$2^{20} \times 1/f_x$ (209.7 ms)	$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)

- Remarks**
1. f_x : Main system clock oscillation frequency
 2. MCS : Bit 0 of oscillation mode selection register (OSMS)
 3. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

★

Figure 9-10. Timing of Square Wave Output Operation



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

9.4.2 16-bit timer/event counter mode

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

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

The counting operation is enabled or disabled in this mode by using bit 0 (TCE1) of TMC1.

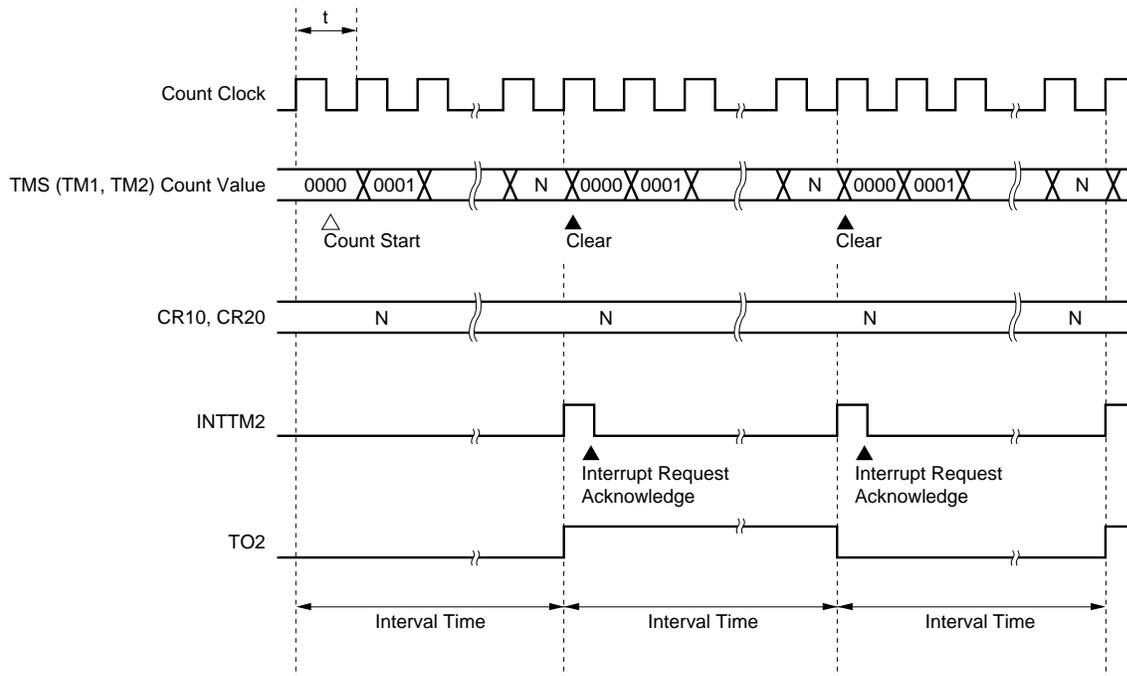
(1) Operation as interval timer

The 16-bit timer/event counter operates as an interval timer that repeatedly generates an interrupt request at intervals of the count values set in advance to the 2 channels of the 8-bit compare registers (CR10 and CR20). When setting a count value, assign the value of the high-order 8 bits to CR20 and the value of the low-order 8 bits to CR10. For the count values that can be set (interval time), refer to **Table 9-9**.

When the value of 8-bit timer register 1 (TM1) coincides with the value of CR10 and 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 at the same time, an interrupt request signal (INTTM2) is generated. For the operation timing of the interval timer, refer to **Figure 9-11**.

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

Figure 9-11. Interval Timer Operation Timing



Remark Interval time = $(N + 1) \times t$: $N = 0000H$ 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 and the F/F of 8-bit timer/event counter output control circuit 1 is inverted. Thus, when using 8-bit timer/event counter as 16-bit interval timer, set the INTTM1 mask flag TMMK1 to 1 to disable INTTM1 acknowledgment. When reading the 16-bit timer (TMS) count value, use the 16-bit memory manipulation instruction.

Table 9-9. Interval Times when 2-Channel 8-Bit Timer/Event Counters (TM1 and TM2) are Used as 16-Bit Timer/Event Counter

TCL13	TCL12	TCL11	TCL10	Minimum Interval Time		Maximum Interval Time		Resolution	
				MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
0	0	0	0	T11 input cycle		$2^8 \times$ T11 input cycle		T11 input edge cycle	
0	0	0	1	T11 input cycle		$2^8 \times$ T11 input cycle		T11 input edge cycle	
0	1	1	0	$2 \times 1/f_x$ (400 ns)	$2^2 \times 1/f_x$ (800 ns)	$2^{17} \times 1/f_x$ (26.2 ms)	$2^{18} \times 1/f_x$ (52.4 ms)	$2 \times 1/f_x$ (400 ns)	$2^2 \times 1/f_x$ (800 ns)
0	1	1	1	$2^2 \times 1/f_x$ (800 ns)	$2^3 \times 1/f_x$ (1.6 μ s)	$2^{18} \times 1/f_x$ (52.4 ms)	$2^{19} \times 1/f_x$ (104.9 ms)	$2^2 \times 1/f_x$ (800 ns)	$2^3 \times 1/f_x$ (1.6 μ s)
1	0	0	0	$2^3 \times 1/f_x$ (1.6 μ s)	$2^4 \times 1/f_x$ (3.2 μ s)	$2^{19} \times 1/f_x$ (104.9 ms)	$2^{20} \times 1/f_x$ (209.7 ms)	$2^3 \times 1/f_x$ (1.6 μ s)	$2^4 \times 1/f_x$ (3.2 μ s)
1	0	0	1	$2^4 \times 1/f_x$ (3.2 μ s)	$2^5 \times 1/f_x$ (6.4 μ s)	$2^{20} \times 1/f_x$ (209.7 ms)	$2^{21} \times 1/f_x$ (419.4 ms)	$2^4 \times 1/f_x$ (3.2 μ s)	$2^5 \times 1/f_x$ (6.4 μ s)
1	0	1	0	$2^5 \times 1/f_x$ (6.4 μ s)	$2^6 \times 1/f_x$ (12.8 μ s)	$2^{21} \times 1/f_x$ (419.4 ms)	$2^{22} \times 1/f_x$ (838.9 ms)	$2^5 \times 1/f_x$ (6.4 μ s)	$2^6 \times 1/f_x$ (12.8 μ s)
1	0	1	1	$2^6 \times 1/f_x$ (12.8 μ s)	$2^7 \times 1/f_x$ (25.6 μ s)	$2^{22} \times 1/f_x$ (838.9 ms)	$2^{23} \times 1/f_x$ (1.7 s)	$2^6 \times 1/f_x$ (12.8 μ s)	$2^7 \times 1/f_x$ (25.6 μ s)
1	1	0	0	$2^7 \times 1/f_x$ (25.6 μ s)	$2^8 \times 1/f_x$ (51.2 μ s)	$2^{23} \times 1/f_x$ (1.7 s)	$2^{24} \times 1/f_x$ (3.4 s)	$2^7 \times 1/f_x$ (25.6 μ s)	$2^8 \times 1/f_x$ (51.2 μ s)
1	1	0	1	$2^8 \times 1/f_x$ (51.2 μ s)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{24} \times 1/f_x$ (3.4 s)	$2^{25} \times 1/f_x$ (6.7 s)	$2^8 \times 1/f_x$ (51.2 μ s)	$2^9 \times 1/f_x$ (102.4 μ s)
1	1	1	0	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)	$2^{25} \times 1/f_x$ (6.7 s)	$2^{26} \times 1/f_x$ (13.4 s)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)
1	1	1	1	$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)	$2^{27} \times 1/f_x$ (26.8 s)	$2^{28} \times 1/f_x$ (53.7 s)	$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)
Other than above				Setting prohibited					

- Remarks**
1. f_x : Main system clock oscillation frequency
 2. MCS : Bit 0 of oscillation mode selection register (OSMS)
 3. TCL10 to TCL13 : Bits 0 to 3 of timer clock select register 1 (TCL1)
 4. Figures in parentheses apply to operation with at $f_x = 5.0$ MHz.

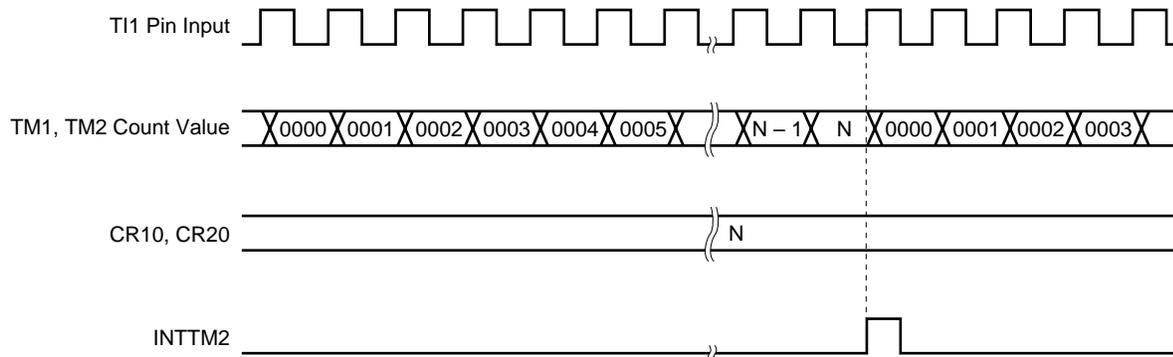
(2) External event counter operations

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

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

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

Figure 9-12. 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 and the F/F of 8-bit timer/event counter output control circuit 1 is inverted. Thus, when using 8-bit timer/event counter as 16-bit interval timer, set the INTTM1 mask flag TMMK1 to 1 to disable INTTM1 acknowledgment. When reading the 16-bit timer register (TMS) count value, use the 16-bit memory manipulation instruction.

(3) Square-wave output operation

The 8-bit timer/event counters 1 and 2 operate as square wave outputs with any selected frequency at intervals of the value preset to 8-bit compare registers (CR10 and CR20). To set the count value, set the values of the higher 8 bits to CR20 and set the values of the lower 8 bits to CR10.

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

Table 9-10. Square-Wave Output Ranges when 2-Channel 8-Bit Timer/Event Counters (TM1 and TM2) are Used as 16-Bit Timer/Event Counter

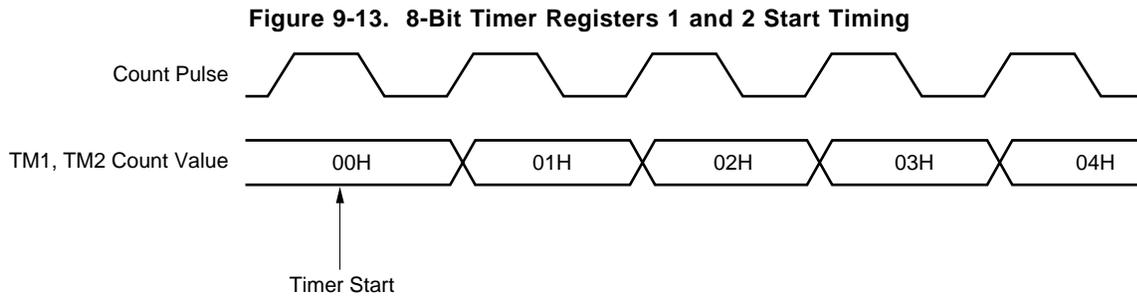
Minimum Pulse Width		Maximum Pulse Width		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
$2 \times 1/f_x$ (400 ns)	$2^2 \times 1/f_x$ (800 ns)	$2^{17} \times 1/f_x$ (26.2 ms)	$2^{18} \times 1/f_x$ (52.4 ms)	$2 \times 1/f_x$ (400 ns)	$2^2 \times 1/f_x$ (800 ns)
$2^2 \times 1/f_x$ (800 ns)	$2^3 \times 1/f_x$ (1.6 μ s)	$2^{18} \times 1/f_x$ (52.4 ms)	$2^{19} \times 1/f_x$ (104.9 ms)	$2^2 \times 1/f_x$ (800 ns)	$2^3 \times 1/f_x$ (1.6 μ s)
$2^3 \times 1/f_x$ (1.6 μ s)	$2^4 \times 1/f_x$ (3.2 μ s)	$2^{19} \times 1/f_x$ (104.9 ms)	$2^{20} \times 1/f_x$ (209.7 ms)	$2^3 \times 1/f_x$ (1.6 μ s)	$2^4 \times 1/f_x$ (3.2 μ s)
$2^4 \times 1/f_x$ (3.2 μ s)	$2^5 \times 1/f_x$ (6.4 μ s)	$2^{20} \times 1/f_x$ (209.7 ms)	$2^{21} \times 1/f_x$ (419.4 ms)	$2^4 \times 1/f_x$ (3.2 μ s)	$2^5 \times 1/f_x$ (6.4 μ s)
$2^5 \times 1/f_x$ (6.4 μ s)	$2^6 \times 1/f_x$ (12.8 μ s)	$2^{21} \times 1/f_x$ (419.4 ms)	$2^{22} \times 1/f_x$ (838.9 ms)	$2^5 \times 1/f_x$ (6.4 μ s)	$2^6 \times 1/f_x$ (12.8 μ s)
$2^6 \times 1/f_x$ (12.8 μ s)	$2^7 \times 1/f_x$ (25.6 μ s)	$2^{22} \times 1/f_x$ (838.9 ms)	$2^{23} \times 1/f_x$ (1.7 s)	$2^6 \times 1/f_x$ (12.8 μ s)	$2^7 \times 1/f_x$ (25.6 μ s)
$2^7 \times 1/f_x$ (25.6 μ s)	$2^8 \times 1/f_x$ (51.2 μ s)	$2^{23} \times 1/f_x$ (1.7 s)	$2^{24} \times 1/f_x$ (3.4 s)	$2^7 \times 1/f_x$ (25.6 μ s)	$2^8 \times 1/f_x$ (51.2 μ s)
$2^8 \times 1/f_x$ (51.2 μ s)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{24} \times 1/f_x$ (3.4 s)	$2^{25} \times 1/f_x$ (6.7 s)	$2^8 \times 1/f_x$ (51.2 μ s)	$2^9 \times 1/f_x$ (102.4 μ s)
$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)	$2^{25} \times 1/f_x$ (6.7 s)	$2^{26} \times 1/f_x$ (13.4 s)	$2^9 \times 1/f_x$ (102.4 μ s)	$2^{10} \times 1/f_x$ (204.8 μ s)
$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)	$2^{27} \times 1/f_x$ (26.8 s)	$2^{28} \times 1/f_x$ (53.7 s)	$2^{11} \times 1/f_x$ (409.6 μ s)	$2^{12} \times 1/f_x$ (819.2 μ s)

- Remarks**
1. f_x : Main system clock oscillation frequency
 2. MCS : Bit 0 of oscillation mode selection register (OSMS)
 3. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

9.5 8-Bit Timer/Event Counters 1 and 2 Precautions

(1) Timer start errors

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

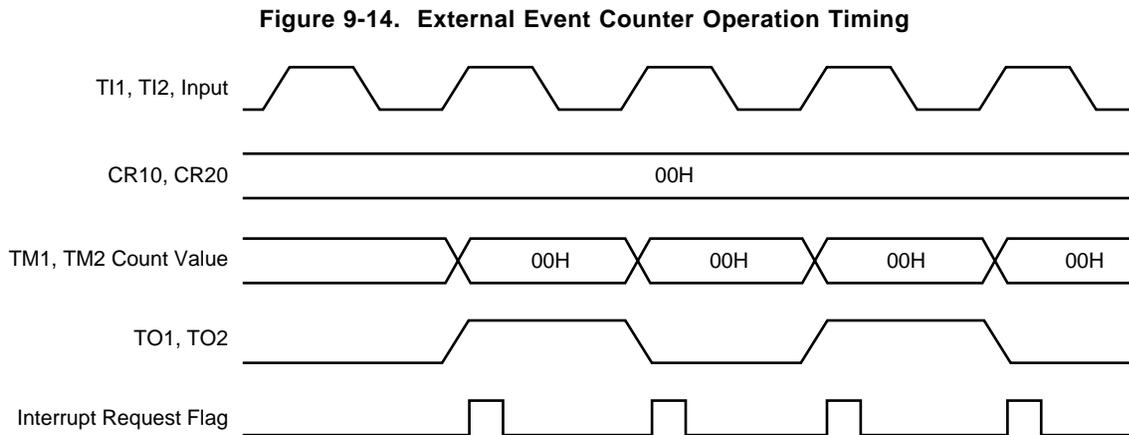


(2) 8-bit compare register 10 and 20 setting

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.

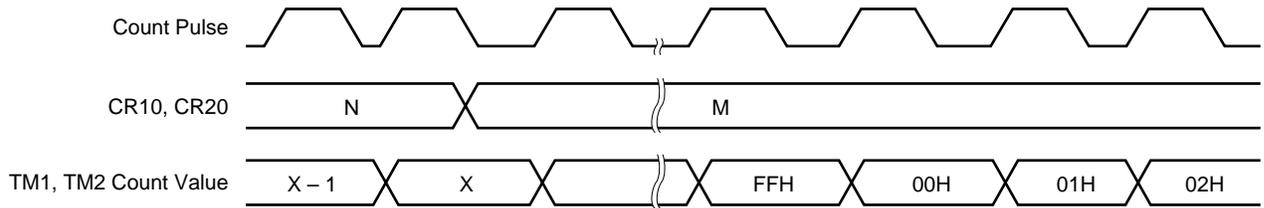
When the 8-bit compare register is used as 16-bit timer/event counter, write data to CR10 and CR20 after setting bit 0 (TCE1) of the 8-bit timer mode control register 1 to 0 and stopping timer operation.



(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 9-15. Timing after Compare Register Change during Timer Count Operation



Remark $N > X > M$

[MEMO]

10.1 8-Bit Timer/Event Counters 5 and 6 Functions

The 8-bit timer event counters 5 and 6 (TM5, TM6) have the following functions.

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

(1) 8-bit interval timer

Interrupt requests are generated at the preset time intervals.

Table 10-1. 8-Bit Timer/Event Counters 5 and 6 Interval Times

Minimum Interval Width		Maximum Interval Width		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
—	1/fx (200 ns)	—	2 ⁸ x 1/fx (51.2 μs)	—	1/fx (200 ns)
1/fx (200 ns)	2 x 1/fx (400 ns)	2 ⁸ x 1/fx (51.2 μs)	2 ⁹ x 1/fx (102.4 μs)	1/fx (200 ns)	2 x 1/fx (400 ns)
2 x 1/fx (400 ns)	2 ² x 1/fx (800 ns)	2 ⁹ x 1/fx (102.4 μs)	2 ¹⁰ x 1/fx (204.8 μs)	2 x 1/fx (400 ns)	2 ² x 1/fx (800 ns)
2 ² x 1/fx (800 ns)	2 ³ x 1/fx (1.6 μs)	2 ¹⁰ x 1/fx (204.8 μs)	2 ¹¹ x 1/fx (409.6 μs)	2 ² x 1/fx (800 ns)	2 ³ x 1/fx (1.6 μs)
2 ³ x 1/fx (1.6 μs)	2 ⁴ x 1/fx (3.2 μs)	2 ¹¹ x 1/fx (409.6 μs)	2 ¹² x 1/fx (819.2 μs)	2 ³ x 1/fx (1.6 μs)	2 ⁴ x 1/fx (3.2 μs)
2 ⁴ x 1/fx (3.2 μs)	2 ⁵ x 1/fx (6.4 μs)	2 ¹² x 1/fx (819.2 μs)	2 ¹³ x 1/fx (1.64 ms)	2 ⁴ x 1/fx (3.2 μs)	2 ⁵ x 1/fx (6.4 μs)
2 ⁵ x 1/fx (6.4 μs)	2 ⁶ x 1/fx (12.8 μs)	2 ¹³ x 1/fx (1.64 ms)	2 ¹⁴ x 1/fx (3.28 ms)	2 ⁵ x 1/fx (6.4 μs)	2 ⁶ x 1/fx (12.8 μs)
2 ⁶ x 1/fx (12.8 μs)	2 ⁷ x 1/fx (25.6 μs)	2 ¹⁴ x 1/fx (3.28 ms)	2 ¹⁵ x 1/fx (6.55 ms)	2 ⁶ x 1/fx (12.8 μs)	2 ⁷ x 1/fx (25.6 μs)
2 ⁷ x 1/fx (25.6 μs)	2 ⁸ x 1/fx (51.2 μs)	2 ¹⁵ x 1/fx (6.55 ms)	2 ¹⁶ x 1/fx (13.1 ms)	2 ⁷ x 1/fx (25.6 μs)	2 ⁸ x 1/fx (51.2 μs)
2 ⁸ x 1/fx (51.2 μs)	2 ⁹ x 1/fx (102.4 μs)	2 ¹⁶ x 1/fx (13.1 ms)	2 ¹⁷ x 1/fx (26.2 ms)	2 ⁸ x 1/fx (51.2 μs)	2 ⁹ x 1/fx (102.4 μs)
2 ⁹ x 1/fx (102.4 μs)	2 ¹⁰ x 1/fx (204.8 μs)	2 ¹⁷ x 1/fx (26.2 ms)	2 ¹⁸ x 1/fx (52.4 ms)	2 ⁹ x 1/fx (102.4 μs)	2 ¹⁰ x 1/fx (204.8 μs)
2 ¹¹ x 1/fx (409.6 μs)	2 ¹² x 1/fx (819.2 μs)	2 ¹⁹ x 1/fx (104.9 ms)	2 ²⁰ x 1/fx (209.7 ms)	2 ¹¹ x 1/fx (409.6 μs)	2 ¹² x 1/fx (819.2 μs)

- Remarks**
1. fx : Main system clock oscillation frequency
 2. MCS : Bit 0 of oscillation mode selection register (OSMS)
 3. Figures in parentheses apply to operation with fx = 5.0 MHz.

(2) External event counter

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

(3) Square-wave output

A square wave with any selected frequency can be output.

Table 10-2. 8-Bit Timer/Event Counters 5 and 6 Square-Wave Output Ranges

Minimum Pulse Width		Maximum Pulse Width		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
—	1/fx (200 ns)	—	2 ⁸ x 1/fx (51.2 μs)	—	1/fx (200 ns)
1/fx (200 ns)	2 x 1/fx (400 ns)	2 ⁸ x 1/fx (51.2 μs)	2 ⁹ x 1/fx (102.4 μs)	1/fx (200 ns)	2 x 1/fx (400 ns)
2 x 1/fx (400 ns)	2 ² x 1/fx (800 ns)	2 ⁹ x 1/fx (102.4 μs)	2 ¹⁰ x 1/fx (204.8 μs)	2 x 1/fx (400 ns)	2 ² x 1/fx (800 ns)
2 ² x 1/fx (800 ns)	2 ³ x 1/fx (1.6 μs)	2 ¹⁰ x 1/fx (204.8 μs)	2 ¹¹ x 1/fx (409.6 μs)	2 ² x 1/fx (800 ns)	2 ³ x 1/fx (1.6 μs)
2 ³ x 1/fx (1.6 μs)	2 ⁴ x 1/fx (3.2 μs)	2 ¹¹ x 1/fx (409.6 μs)	2 ¹² x 1/fx (819.2 μs)	2 ³ x 1/fx (1.6 μs)	2 ⁴ x 1/fx (3.2 μs)
2 ⁴ x 1/fx (3.2 μs)	2 ⁵ x 1/fx (6.4 μs)	2 ¹² x 1/fx (819.2 μs)	2 ¹³ x 1/fx (1.64 ms)	2 ⁴ x 1/fx (3.2 μs)	2 ⁵ x 1/fx (6.4 μs)
2 ⁵ x 1/fx (6.4 μs)	2 ⁶ x 1/fx (12.8 μs)	2 ¹³ x 1/fx (1.64 ms)	2 ¹⁴ x 1/fx (3.28 ms)	2 ⁵ x 1/fx (6.4 μs)	2 ⁶ x 1/fx (12.8 μs)
2 ⁶ x 1/fx (12.8 μs)	2 ⁷ x 1/fx (25.6 μs)	2 ¹⁴ x 1/fx (3.28 ms)	2 ¹⁵ x 1/fx (6.55 ms)	2 ⁶ x 1/fx (12.8 μs)	2 ⁷ x 1/fx (25.6 μs)
2 ⁷ x 1/fx (25.6 μs)	2 ⁸ x 1/fx (51.2 μs)	2 ¹⁵ x 1/fx (6.55 ms)	2 ¹⁶ x 1/fx (13.1 ms)	2 ⁷ x 1/fx (25.6 μs)	2 ⁸ x 1/fx (51.2 μs)
2 ⁸ x 1/fx (51.2 μs)	2 ⁹ x 1/fx (102.4 μs)	2 ¹⁶ x 1/fx (13.1 ms)	2 ¹⁷ x 1/fx (26.2 ms)	2 ⁸ x 1/fx (51.2 μs)	2 ⁹ x 1/fx (102.4 μs)
2 ⁹ x 1/fx (102.4 μs)	2 ¹⁰ x 1/fx (204.8 μs)	2 ¹⁷ x 1/fx (26.2 ms)	2 ¹⁸ x 1/fx (52.4 ms)	2 ⁹ x 1/fx (102.4 μs)	2 ¹⁰ x 1/fx (204.8 μs)
2 ¹¹ x 1/fx (409.6 μs)	2 ¹² x 1/fx (819.2 μs)	2 ¹⁹ x 1/fx (104.9 ms)	2 ²⁰ x 1/fx (209.7 ms)	2 ¹¹ x 1/fx (409.6 μs)	2 ¹² x 1/fx (819.2 μs)

- Remarks**
1. fx : Main system clock oscillation frequency
 2. MCS : Bit 0 of oscillation mode selection register (OSMS)
 3. Figures in parentheses apply to operation with fx = 5.0 MHz.

(4) PWM output

TM5 and TM6 can generate 8-bit resolution PWM output.

10.2 8-Bit Timer/Event Counters 5 and 6 Configurations

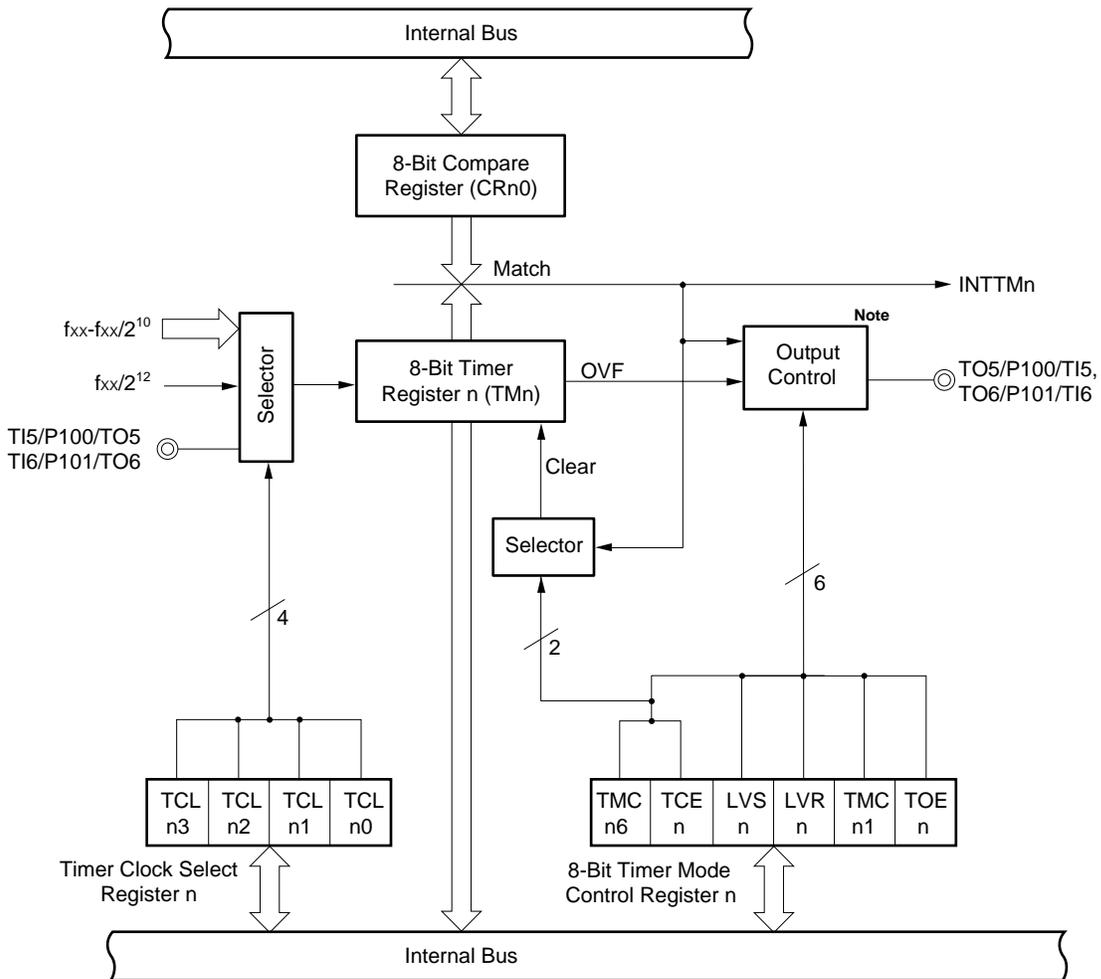
The 8-bit timer/event counters 5 and 6 consist of the following hardware.

Table 10-3. 8-Bit Timer/Event Counters 5 and 6 Configurations

Item	Configuration
Timer register	8 bits x 2 (TM5, TM6)
Register	Compare register: 8 bits x 2 (CR50, CR60)
Timer output	2 (TO5, TO6)
Control register	Timer clock select register 5 and 6 (TCL5, TCL6) 8-bit timer mode control registers 5 and 6 (TMC5, TMC6) Port mode register 10 (PM10) ^{Note}

Note Refer to **Figure 6-20 Block Diagram of P100 and P101**.

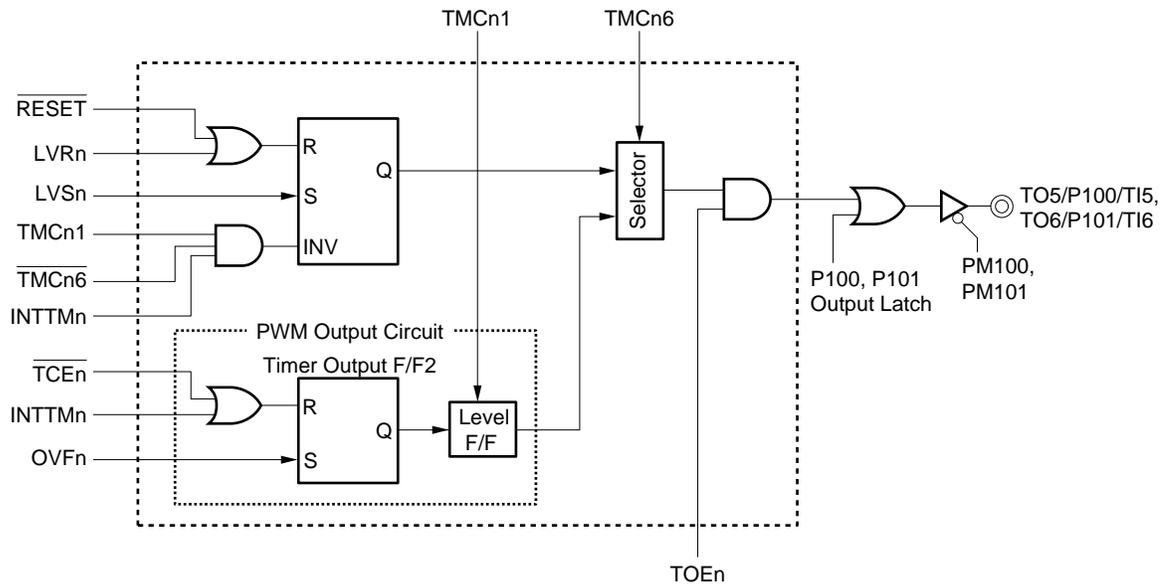
Figure 10-1. 8-Bit Timer/Event Counters 5 and 6 Block Diagram



Note Refer to **Figure 10-2** for details of configurations of 8-bit timer/event counters 5 and 6 output control circuits.

Remark n = 5, 6

Figure 10-2. Block Diagram of 8-Bit Timer/Event Counters 5 and 6 Output Control Circuit



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

2. $n = 5, 6$

(1) Compare register 50 and 60 (CR50, 60)

These 8-bit registers compare the value set to CR50 to 8-bit timer register 5 (TM5) count value, and the value set to CR60 to the 8-bit timer register 6 (TM6) count value, and, if they match, generate interrupts request (INTTM5 and INTTM6, respectively).

CR50 and CR60 are set with an 8-bit memory manipulation instruction. They cannot be set with a 16-bit memory manipulation instruction. The 00H to FFH values can be set.

$\overline{\text{RESET}}$ input sets CR50 and CR60 values to 00H.

Caution To use PWM mode, set CRn0 value before setting TMCn ($n = 5, 6$) to PWM mode.

(2) 8-bit timer registers 5 and 6 (TM5, TM6)

These 8-bit registers count count pulses.

TM5 and TM6 are read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TM5 and TM6 to 00H.

10.3 8-Bit Timer/Event Counters 5 and 6 Control Registers

The following three types of registers are used to control the 8-bit timer/event counters 5 and 6.

- Timer clock select register 5 and 6 (TCL5, TCL6)
- 8-bit timer mode control registers 5 and 6 (TMC5, TMC6)
- Port mode register 10 (PM10)

(1) Timer clock select register 5 (TCL5)

This register sets count clocks of 8-bit timer register 5.
 TCL5 is set with an 8-bit memory manipulation instruction.
 $\overline{\text{RESET}}$ input sets TCL5 to 00H.

Figure 10-3. Timer Clock Select Register 5 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL5	0	0	0	0	TCL53	TCL52	TCL51	TCL50	FF52H	00H	R/W

TCL53				TCL52				TCL51				TCL50				8-bit Timer Register 5 Count Clock Selection			
												MCS = 1				MCS = 0			
0	0	0	0	0	0	0	0	TI5 falling edge ^{Note}											
0	0	0	1	TI5 rising edge ^{Note}															
0	1	0	0	2f _{xx}	(Setting prohibited)				f _x	(5.0 MHz)									
0	1	0	1	f _{xx}	f _x (5.0 MHz)				f _x /2	(2.5 MHz)									
0	1	1	0	f _{xx} /2	f _x /2 (2.5 MHz)				f _x /2 ²	(1.25 MHz)									
0	1	1	1	f _{xx} /2 ²	f _x /2 ² (1.25 MHz)				f _x /2 ³	(625 kHz)									
1	0	0	0	f _{xx} /2 ³	f _x /2 ³ (625 kHz)				f _x /2 ⁴	(313 kHz)									
1	0	0	1	f _{xx} /2 ⁴	f _x /2 ⁴ (313 kHz)				f _x /2 ⁵	(156 kHz)									
1	0	1	0	f _{xx} /2 ⁵	f _x /2 ⁵ (156 kHz)				f _x /2 ⁶	(78.1 kHz)									
1	0	1	1	f _{xx} /2 ⁶	f _x /2 ⁶ (78.1 kHz)				f _x /2 ⁷	(39.1 kHz)									
1	1	0	0	f _{xx} /2 ⁷	f _x /2 ⁷ (39.1 kHz)				f _x /2 ⁸	(19.5 kHz)									
1	1	0	1	f _{xx} /2 ⁸	f _x /2 ⁸ (19.5 kHz)				f _x /2 ⁹	(9.8 kHz)									
1	1	1	0	f _{xx} /2 ⁹	f _x /2 ⁹ (9.8 kHz)				f _x /2 ¹⁰	(4.9 kHz)									
1	1	1	1	f _{xx} /2 ¹¹	f _x /2 ¹¹ (2.4 kHz)				f _x /2 ¹¹	(1.2 kHz)									
Other than above				Setting prohibited															

Note When clock is input from the external, timer output (PWM output) cannot be used.

Caution When rewriting TCL5 to other data, stop the timer operation beforehand.

- Remarks**
1. f_{xx} : Main system clock frequency (f_x or f_x/2)
 2. f_x : Main system clock oscillation frequency
 3. TI5 : 8-bit timer register 5 input pin
 4. MCS : Bit 0 of oscillation mode selection register (OSMS)
 5. Figures in parentheses apply to operation with f_x = 5.0 MHz.

(2) Timer clock select register 6 (TCL6)

This register sets count clocks of 8-bit timer register 6.

TCL6 is set with an 8-bit memory manipulation instruction.

RESET input sets TCL6 to 00H.

Figure 10-4. Timer Clock Select Register 6 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL6	0	0	0	0	TCL63	TCL62	TCL61	TCL60	FF56H	00H	R/W

TCL63	TCL62	TCL61	TCL60	8-bit Timer Register 5 Count Clock Selection			
				MCS = 1		MCS = 0	
0	0	0	0	TI6 falling edge ^{Note}			
0	0	0	1	TI6 rising edge ^{Note}			
0	1	0	0	2f _{xx}	(Setting prohibited)		f _x (5.0 MHz)
0	1	0	1	f _{xx}	f _x (5.0 MHz)	f _x /2 (2.5 MHz)	
0	1	1	0	f _{xx} /2	f _x /2 (2.5 MHz)	f _x /2 ² (1.25 MHz)	
0	1	1	1	f _{xx} /2 ²	f _x /2 ² (1.25 MHz)	f _x /2 ³ (625 kHz)	
1	0	0	0	f _{xx} /2 ³	f _x /2 ³ (625 kHz)	f _x /2 ⁴ (313 kHz)	
1	0	0	1	f _{xx} /2 ⁴	f _x /2 ⁴ (313 kHz)	f _x /2 ⁵ (156 kHz)	
1	0	1	0	f _{xx} /2 ⁵	f _x /2 ⁵ (156 kHz)	f _x /2 ⁶ (78.1 kHz)	
1	0	1	1	f _{xx} /2 ⁶	f _x /2 ⁶ (78.1 kHz)	f _x /2 ⁷ (39.1 kHz)	
1	1	0	0	f _{xx} /2 ⁷	f _x /2 ⁷ (39.1 kHz)	f _x /2 ⁸ (19.5 kHz)	
1	1	0	1	f _{xx} /2 ⁸	f _x /2 ⁸ (19.5 kHz)	f _x /2 ⁹ (9.8 kHz)	
1	1	1	0	f _{xx} /2 ⁹	f _x /2 ⁹ (9.8 kHz)	f _x /2 ¹⁰ (4.9 kHz)	
1	1	1	1	f _{xx} /2 ¹¹	f _x /2 ¹¹ (2.4 kHz)	f _x /2 ¹¹ (1.2 kHz)	
Other than above				Setting prohibited			

Note When clock is input from the external, timer output (PWM output) cannot be used.

Caution When rewriting TCL6 to other data, stop the timer operation beforehand.

- Remarks**
1. f_{xx} : Main system clock frequency (f_x or f_x/2)
 2. f_x : Main system clock oscillation frequency
 3. TI6 : 8-bit timer register 6 input pin
 4. MCS : Bit 0 of oscillation mode selection register (OSMS)
 5. Figures in parentheses apply to operation with f_x = 5.0 MHz.

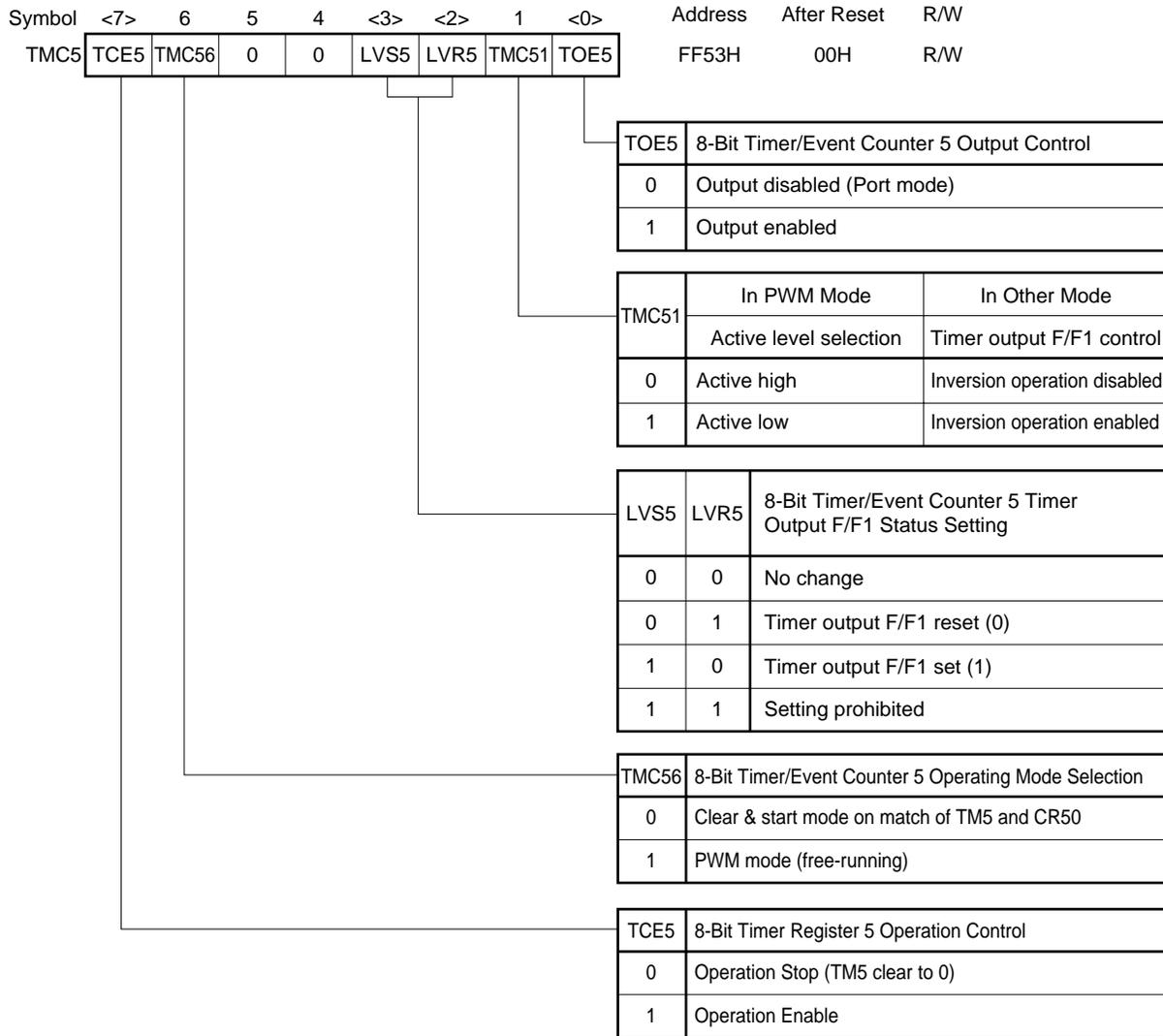
(3) 8-bit timer mode control register 5 (TMC5)

This register enables/stops operation of 8-bit timer register 5, sets the operating mode of 8-bit timer register 5 and controls operation of 8-bit timer/event counter 5 output control circuit.

It sets R-S flip-flop (timer output F/F 1,2) setting/resetting, the active level in PWM mode, inversion enabling/disabling in modes other than PWM mode and 8-bit timer/event counter 5 timer output enabling/disabling. TMC5 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMC5 to 00H.

Figure 10-5. 8-Bit Timer Output Control Register Format



- Cautions**
1. Timer operation must be stopped before setting TMC5.
 2. If LVS5 and LVR5 are read after data are set, they will be 0.
 3. Set bits 4 and 5 to 0.

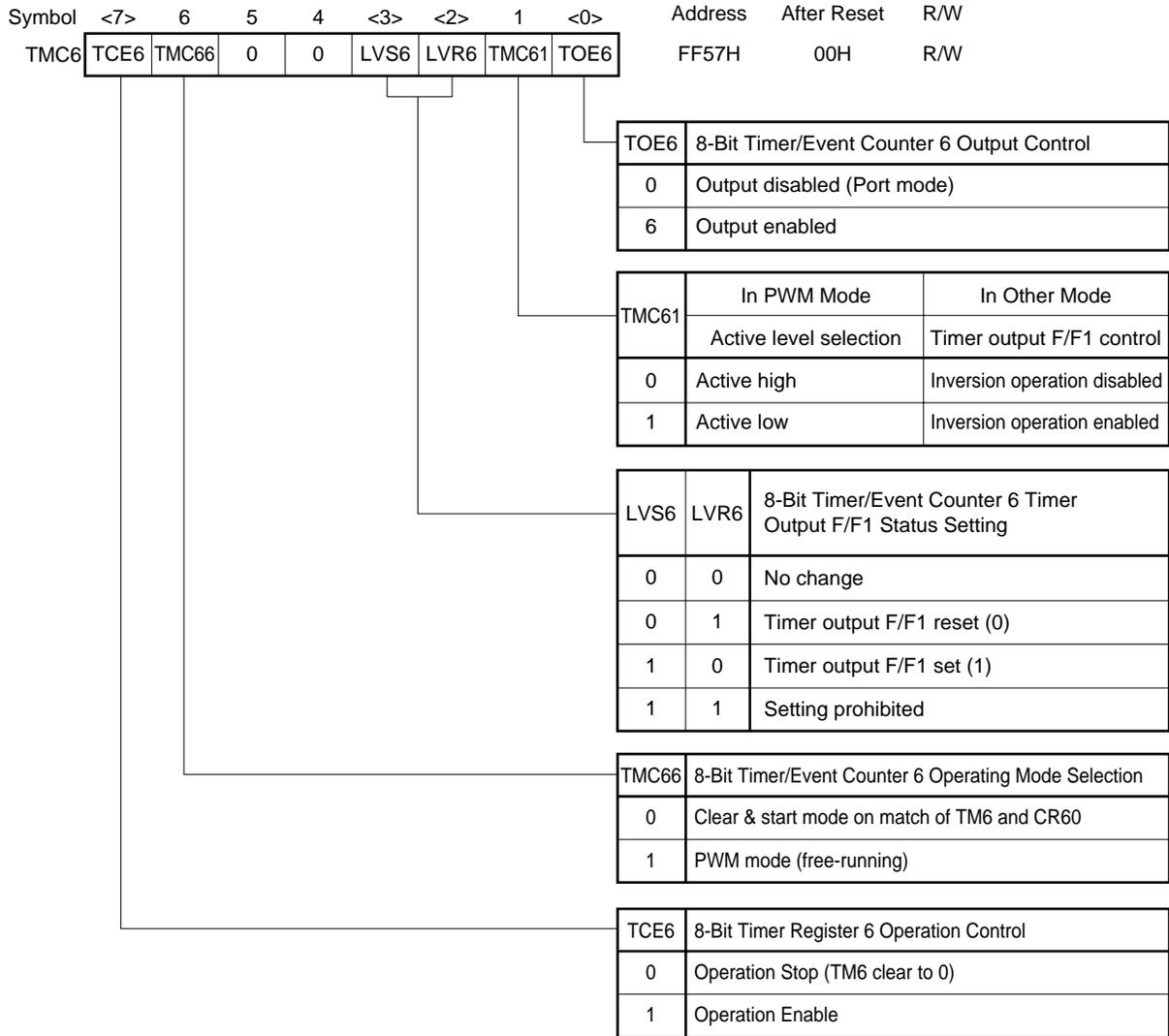
(4) 8-bit timer mode control register 6 (TMC6)

This register enables/stops operation of 8-bit timer register 6, sets the operating mode of 8-bit timer register 6 and controls operation of 8-bit timer/event counter 6 output control circuit.

It sets R-S flip-flop (timer output F/F 1,2) setting/resetting, active level in PWM mode, inversion enabling/disabling in modes other than PWM mode and 8-bit timer/event counter 6 timer output enabling/disabling. TMC6 is set with a 1-bit or 8-bit memory manipulation instruction.

RESET input sets TMC6 to 00H.

Figure 10-6. 8-Bit Timer Output Control Register 6 Format



- Cautions**
1. Timer operation must be stopped before setting TMC6.
 2. If LVS6 and LVR6 are read after data are set, they will be 0.
 3. Set bits 4 and 5 to 0.

(5) Port mode register 10 (PM10)

This register sets port 10 input/output in 1-bit units.

When using the P100/TI5/TO5 and P101/TI6/TO6 pins for timer output, set PM100, PM101 and output latches of P100 and P101 to 0.

PM10 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM10 to FFH.

Figure 10-7. Port Mode Register 10 Format



10.4 8-Bit Timer/Event Counters 5 and 6 Operations

10.4.1 Interval timer operations

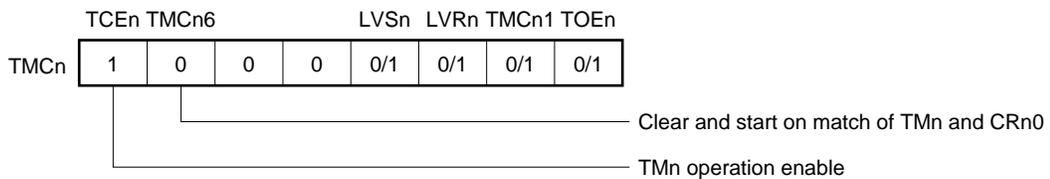
Setting the 8-bit timer mode control registers (TMC5 and TMC6) as shown in Figure 10-8 allows operation as an interval timer. Interrupt requests are generated repeatedly using the count value preset in 8-bit compare registers (CR50 and CR60) as the interval.

When the count value of the 8-bit timer register 5 or 6 (TM5, TM6) matches the value set to CR50 or CR60, counting continues with the TM5 or TM6 value cleared to 0 and the interrupt request signal (INTTM5, INTTM6) is generated.

Count clock of the 8-bit timer register 5 (TM5) can be selected with the timer clock select register 5 (TCL5) and count clock of the 8 bit timer register 6 (TM6) can be selected with the timer clock select register 6 (TCL6).

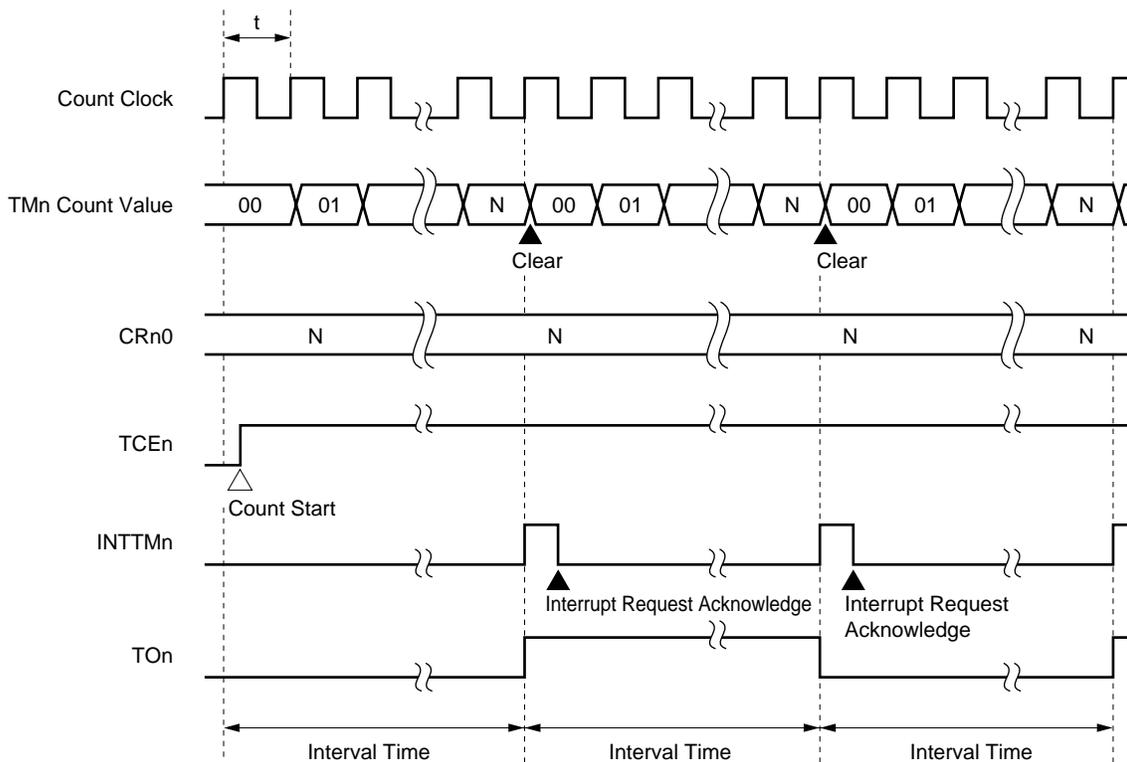
For the operation in the case the value of the compare register is changed during the timer count operation, refer to **10.5 8-Bit Timer/Event Counters 5 and 6 Precautions (3)**.

Figure 10-8. 8-Bit Timer Mode Control Register Settings for Interval Timer Operation



- Remarks 1.** 0/1 : Setting 0 or 1 allows another function to be used simultaneously with the interval timer. See **10.3 (3), (4)** for details.
- 2.** n = 5, 6

Figure 10-9. Interval Timer Operation Timings



- Remarks 1.** Interval time = $(N + 1) \times t$: N = 00H to FFH
- 2.** n = 5, 6

Table 10-4. 8-Bit Timer/Event Counters 5 and 6 Interval Times

TCLn3	TCLn2	TCLn1	TCLn0	Minimum Interval Time		Maximum Interval Time		Resolution	
				MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
0	0	0	0	TIn input cycle		2 ⁸ x TIn input cycle		TIn input edge input cycle	
0	0	0	1	TIn input cycle		2 ⁸ x TIn input cycle		TIn input edge input cycle	
0	1	0	0	(Setting prohibited)	1/fx (200 ns)	(Setting prohibited)	2 ⁸ x 1/fx (51.2 μs)	(Setting prohibited)	2 x 1/fx (200 ns)
0	1	0	1	1/fx (200 ns)	2 x 1/fx (400 ns)	2 ⁸ x 1/fx (51.2 μs)	2 ⁹ x 1/fx (102.4 μs)	1/fx (200 ns)	2 x 1/fx (400 ns)
0	1	1	0	2 x 1/fx (400 ns)	2 ² x 1/fx (800 ns)	2 ⁹ x 1/fx (102.4 μs)	2 ¹⁰ x 1/fx (204.8 μs)	2 x 1/fx (400 ns)	2 ² x 1/fx (800 ns)
0	1	1	1	2 ² x 1/fx (800 ns)	2 ³ x 1/fx (1.6 μs)	2 ¹⁰ x 1/fx (204.8 μs)	2 ¹¹ x 1/fx (409.6 μs)	2 ² x 1/fx (800 ns)	2 ³ x 1/fx (1.6 μs)
1	0	0	0	2 ³ x 1/fx (1.6 μs)	2 ⁴ x 1/fx (3.2 μs)	2 ¹¹ x 1/fx (409.6 μs)	2 ¹² x 1/fx (819.2 μs)	2 ³ x 1/fx (1.6 μs)	2 ⁴ x 1/fx (3.2 μs)
1	0	0	1	2 ⁴ x 1/fx (3.2 μs)	2 ⁵ x 1/fx (6.4 μs)	2 ¹² x 1/fx (819.2 μs)	2 ¹³ x 1/fx (1.64 ms)	2 ⁴ x 1/fx (3.2 μs)	2 ⁵ x 1/fx (6.4 μs)
1	0	1	0	2 ⁵ x 1/fx (6.4 μs)	2 ⁶ x 1/fx (12.8 μs)	2 ¹³ x 1/fx (1.64 ms)	2 ¹⁴ x 1/fx (3.28 ms)	2 ⁵ x 1/fx (6.4 μs)	2 ⁶ x 1/fx (12.8 μs)
1	0	1	1	2 ⁶ x 1/fx (12.8 μs)	2 ⁷ x 1/fx (25.6 μs)	2 ¹⁴ x 1/fx (3.28 ms)	2 ¹⁵ x 1/fx (6.55 ms)	2 ⁶ x 1/fx (12.8 μs)	2 ⁷ x 1/fx (25.6 μs)
1	1	0	0	2 ⁷ x 1/fx (25.6 μs)	2 ⁸ x 1/fx (51.2 μs)	2 ¹⁵ x 1/fx (6.55 ms)	2 ¹⁶ x 1/fx (13.1 ms)	2 ⁷ x 1/fx (25.6 μs)	2 ⁸ x 1/fx (51.2 μs)
1	1	0	1	2 ⁸ x 1/fx (51.2 μs)	2 ⁹ x 1/fx (102.4 μs)	2 ¹⁶ x 1/fx (13.1 ms)	2 ¹⁷ x 1/fx (26.2 ms)	2 ⁸ x 1/fx (51.2 μs)	2 ⁹ x 1/fx (102.4 μs)
1	1	1	0	2 ⁹ x 1/fx (102.4 μs)	2 ¹⁰ x 1/fx (204.8 μs)	2 ¹⁷ x 1/fx (26.2 ms)	2 ¹⁸ x 1/fx (52.4 ms)	2 ⁹ x 1/fx (102.4 μs)	2 ¹⁰ x 1/fx (204.8 μs)
1	1	1	1	2 ¹¹ x 1/fx (409.6 μs)	2 ¹² x 1/fx (819.2 μs)	2 ¹⁹ x 1/fx (104.9 ms)	2 ²⁰ x 1/fx (209.7 ms)	2 ¹¹ x 1/fx (409.6 μs)	2 ¹² x 1/fx (819.2 μs)
Other than above				Setting prohibited					

- Remarks**
1. fx : Main system clock oscillation frequency
 2. MCS : Bit 0 of oscillation mode selection register (OSMS)
 3. TCLn0 to TCLn3 : Bits 0 to 3 of timer clock select register n (TCLn)
 4. Figures in parentheses apply to operation with fx = 5.0 MHz.
 5. n = 5, 6

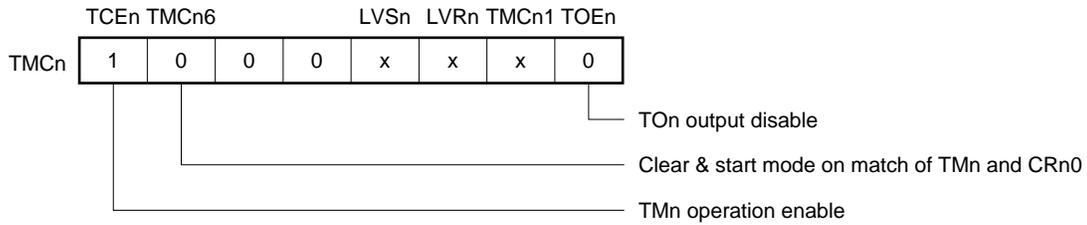
10.4.2 External event counter operation

The external event counter counts the number of external clock pulses to be input to the TI5/P100/TO5 and TI6/P101/TO6 pins with 8-bit timer registers 5 and 6 (TM5 and TM6).

TM5 and TM6 are incremented each time the valid edge specified with timer clock select registers 5 and 6 (TCL5 and TCL6) is input. Either rising or falling edge can be selected.

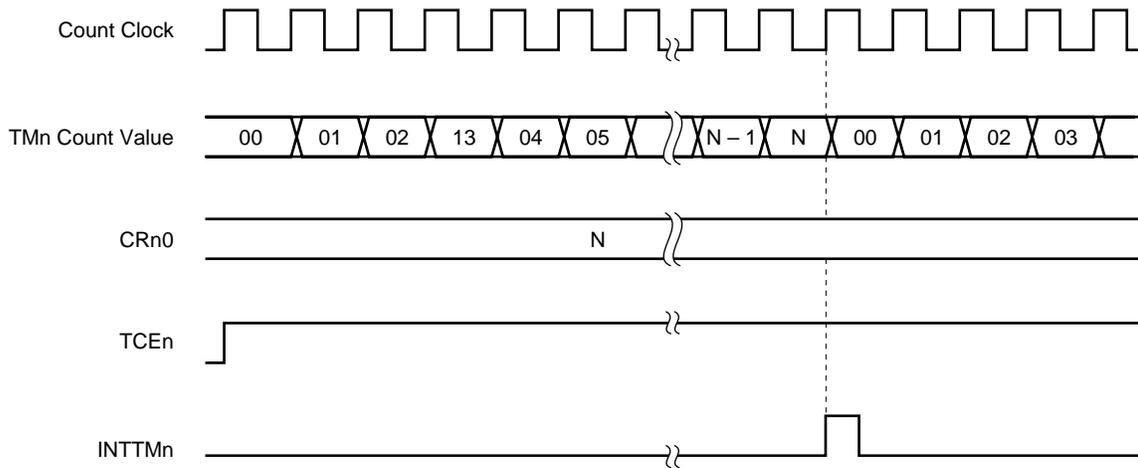
When the TM5 and TM6 counted values match the values of 8-bit compare registers (CR50 and CR60), TM5 and TM6 are cleared to 0 and the interrupt request signals (INTTM5 and INTTM6) are generated.

Figure 10-10. 8-Bit Timer Mode Control Register Setting for External Event Counter Operation



- Remarks 1.** n = 5, 6
- 2.** x : Don't care

Figure 10-11. External Event Counter Operation Timings (with Rising Edge Specified)



- Remarks 1.** N = 00H to FFH
- 2.** n = 5, 6

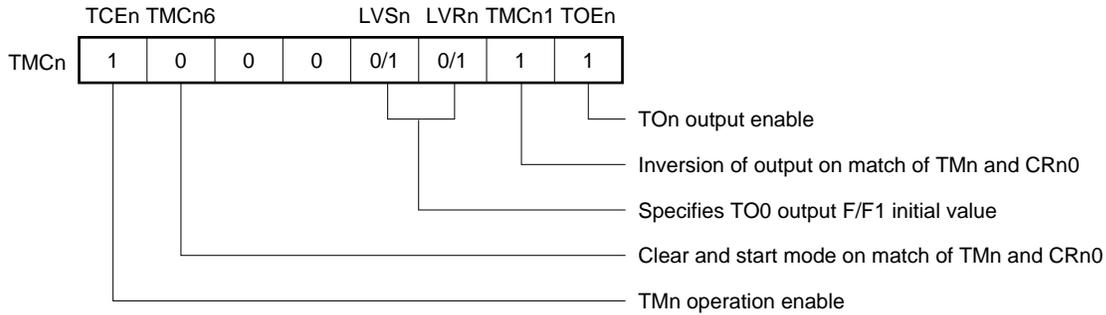
10.4.3 Square-wave output

A square wave with any selected frequency is output at intervals of the value preset to 8-bit compare registers (CR50 and CR60).

The TO5/P100/TI5 or TO6/P101/TI6 pin output status is reversed at intervals of the count value preset to CR50 or CR60 by setting bit 1 (TMC51) and bit 0 (TOE5) of the 8-bit timer output control register 5 (TMC5), or bit 1 (TMC61) and bit 0 (TOE1) of the 8-bit timer mode control register 6 (TMC6) to 1.

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

Figure 10-12. 8-Bit Timer Mode Control Register Settings for Square-Wave Output Operation

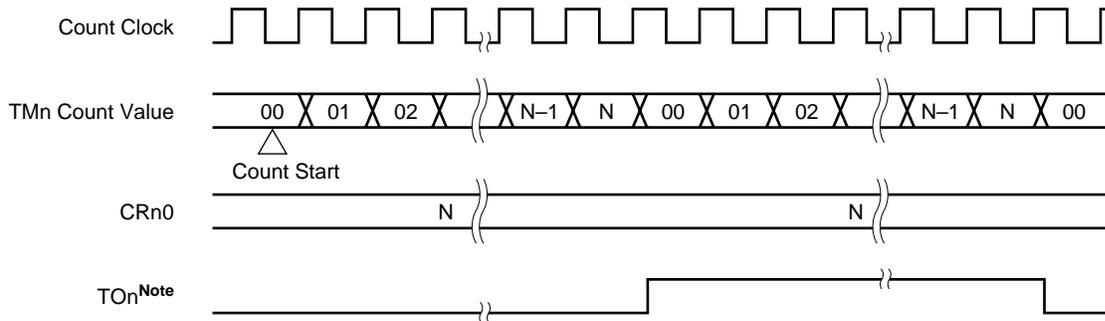


Caution When TI5/P100/TO5 or TI6/P101/TO6 pin is used as the timer output, set port mode register (PM100 or PM101) and output latch (P100 or P101) to 0.

Remark n = 5, 6

★

Figure 10-13. Timing of Square Wave Output Operation



Note The initial value of the TOn output can be set by bits 2 and 3 (LVSn and LVRn) of the 8-bit timer output control register n (TOCn).

Remark n = 5, 6

Table 10-5. 8-Bit Timer/Event Counters 5 and 6 Square-Wave Output Ranges

Minimum Pulse Time		Maximum Pulse Time		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
—	1/fx (200 ns)	—	2 ⁸ x 1/fx (51.2 μs)	—	1/fx (200 ns)
1/fx (200 ns)	2 x 1/fx (400 ns)	2 ⁸ x 1/fx (51.2 μs)	2 ⁹ x 1/fx (102.4 μs)	1/fx (200 ns)	2 x 1/fx (400 ns)
2 x 1/fx (400 ns)	2 ² x 1/fx (800 ns)	2 ⁹ x 1/fx (102.4 μs)	2 ¹⁰ x 1/fx (204.8 μs)	2 x 1/fx (400 ns)	2 ² x 1/fx (800 ns)
2 ² x 1/fx (800 ns)	2 ³ x 1/fx (1.6 μs)	2 ¹⁰ x 1/fx (204.8 μs)	2 ¹¹ x 1/fx (409.6 μs)	2 ² x 1/fx (800 ns)	2 ³ x 1/fx (1.6 μs)
2 ³ x 1/fx (1.6 μs)	2 ⁴ x 1/fx (3.2 μs)	2 ¹¹ x 1/fx (409.6 μs)	2 ¹² x 1/fx (819.2 μs)	2 ³ x 1/fx (1.6 μs)	2 ⁴ x 1/fx (3.2 μs)
2 ⁴ x 1/fx (3.2 μs)	2 ⁵ x 1/fx (6.4 μs)	2 ¹² x 1/fx (819.2 μs)	2 ¹³ x 1/fx (1.64 ms)	2 ⁴ x 1/fx (3.2 μs)	2 ⁵ x 1/fx (6.4 μs)
2 ⁵ x 1/fx (6.4 μs)	2 ⁶ x 1/fx (12.8 μs)	2 ¹³ x 1/fx (1.64 ms)	2 ¹⁴ x 1/fx (3.28 ms)	2 ⁵ x 1/fx (6.4 μs)	2 ⁶ x 1/fx (12.8 μs)
2 ⁶ x 1/fx (12.8 μs)	2 ⁷ x 1/fx (25.6 μs)	2 ¹⁴ x 1/fx (3.28 ms)	2 ¹⁵ x 1/fx (6.55 ms)	2 ⁶ x 1/fx (12.8 μs)	2 ⁷ x 1/fx (25.6 μs)
2 ⁷ x 1/fx (25.6 μs)	2 ⁸ x 1/fx (51.2 μs)	2 ¹⁵ x 1/fx (6.55 ms)	2 ¹⁶ x 1/fx (13.1 ms)	2 ⁷ x 1/fx (25.6 μs)	2 ⁸ x 1/fx (51.2 μs)
2 ⁸ x 1/fx (51.2 μs)	2 ⁹ x 1/fx (102.4 μs)	2 ¹⁶ x 1/fx (13.1 ms)	2 ¹⁷ x 1/fx (26.2 ms)	2 ⁸ x 1/fx (51.2 μs)	2 ⁹ x 1/fx (102.4 μs)
2 ⁹ x 1/fx (102.4 μs)	2 ¹⁰ x 1/fx (204.8 μs)	2 ¹⁷ x 1/fx (26.2 ms)	2 ¹⁸ x 1/fx (52.4 ms)	2 ⁹ x 1/fx (102.4 μs)	2 ¹⁰ x 1/fx (204.8 μs)
2 ¹¹ x 1/fx (409.6 μs)	2 ¹² x 1/fx (819.2 μs)	2 ¹⁹ x 1/fx (104.9 ms)	2 ²⁰ x 1/fx (209.7 ms)	2 ¹¹ x 1/fx (409.6 μs)	2 ¹² x 1/fx (819.2 μs)

- Remarks**
1. f_x : Main system clock oscillation frequency
 2. MCS : Bit 0 of oscillation mode selection register (OSMS)
 3. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.
 4. $n = 5, 6$

10.4.4 PWM output operations

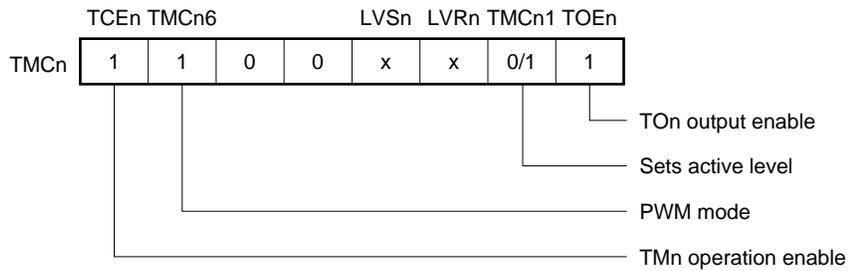
Setting the 8-bit timer mode control registers (TMC5 and TMC6) as shown in Figure 10-14 allows operation as PWM output. Pulses with the duty rate determined by the values preset in 8-bit compare registers (CR50 and CR60) output from the TO5/P100/TI5 or TO6/P101/TI6 pin.

Select the active level of PWM pulse with bit 1 of the 8-bit timer mode control register 5 (TMC5) or bit 1 of the 8-bit timer mode control register 6 (TMC6).

This PWM pulse has an 8-bit resolution. The pulse can be converted into an analog voltage by integrating it with an external low-pass filter (LPF). Count clock of the 8-bit timer register 5 (TM5) can be selected with the timer clock select register 5 (TCL5) and count clock of the 8-bit timer register 6 (TM6) can be selected with the timer clock select register 6 (TCL6).

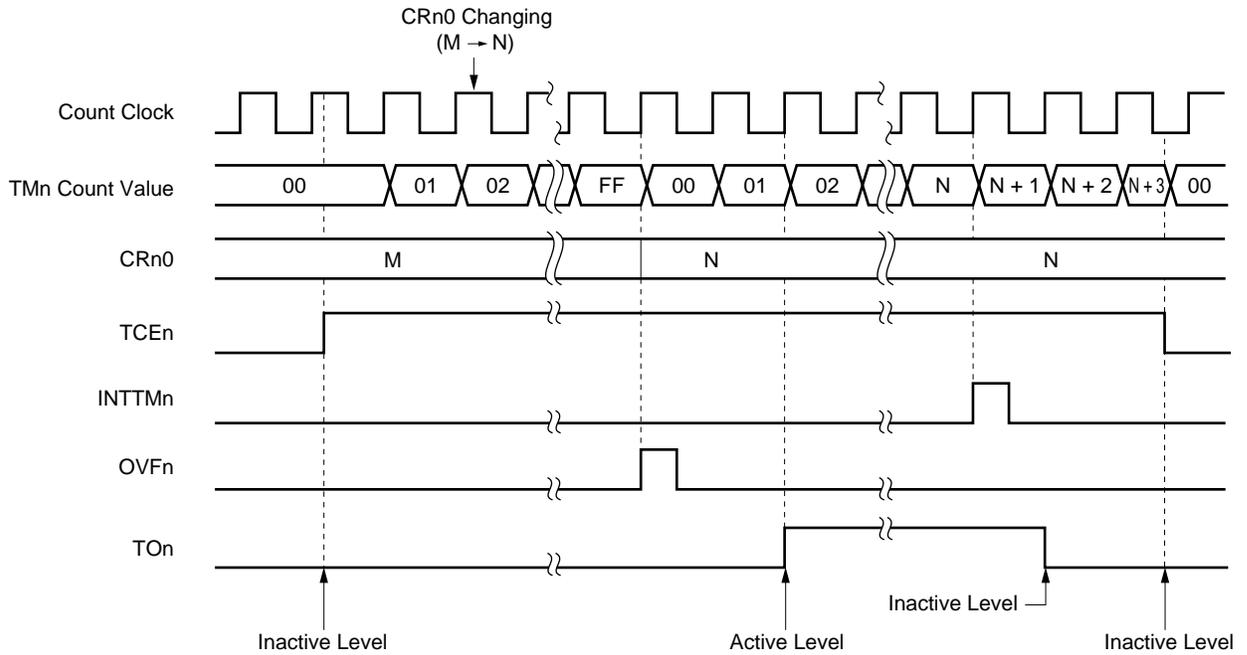
PWM output enable/disable can be selected with bit 0 (TOE5) of TMC5 or bit 0 (TOE6) of TMC6.

Figure 10-14. 8-Bit Timer Control Register Settings for PWM Output Operation



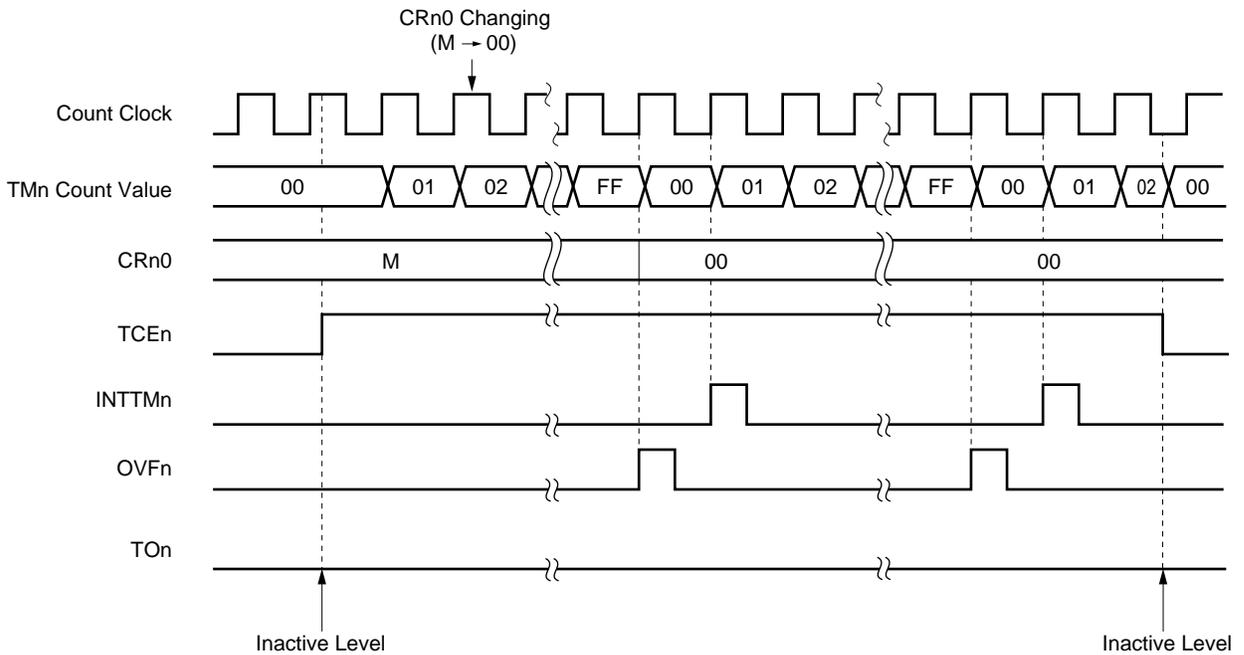
- Remarks**
1. n = 5, 6
 2. x : Don't care

Figure 10-15. PWM Output Operation Timing (Active High Setting)



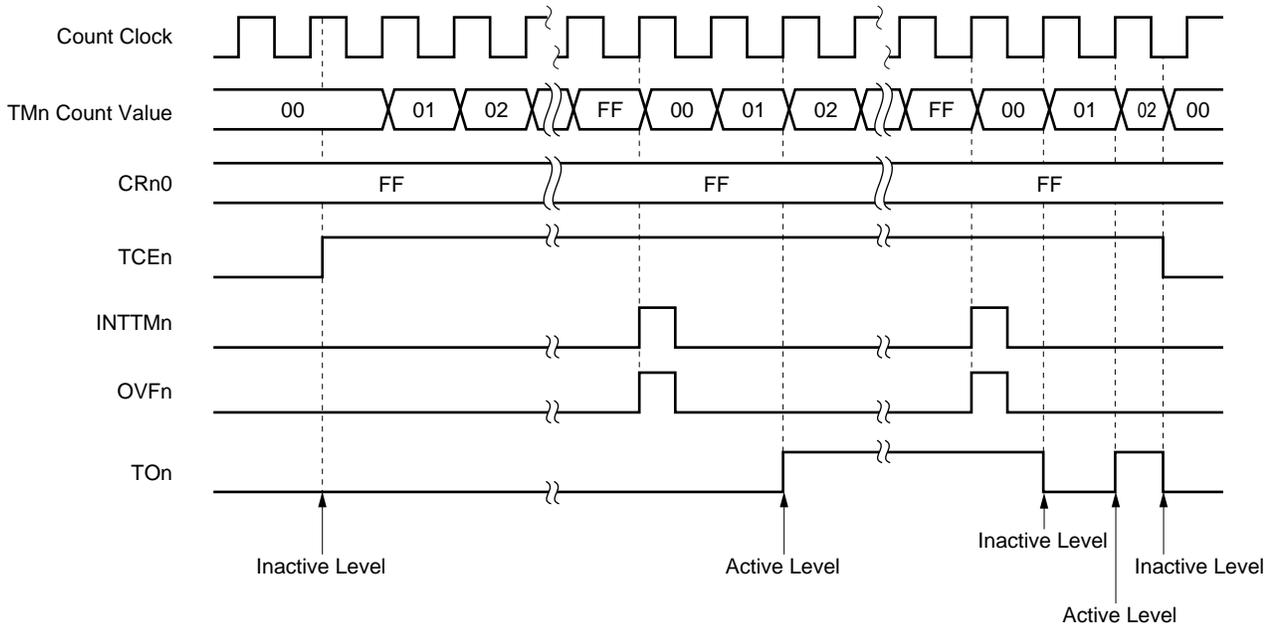
Remark $n = 5, 6$

Figure 10-16. PWM Output Operation Timings ($CR_n0 = 00H$, Active High Setting)



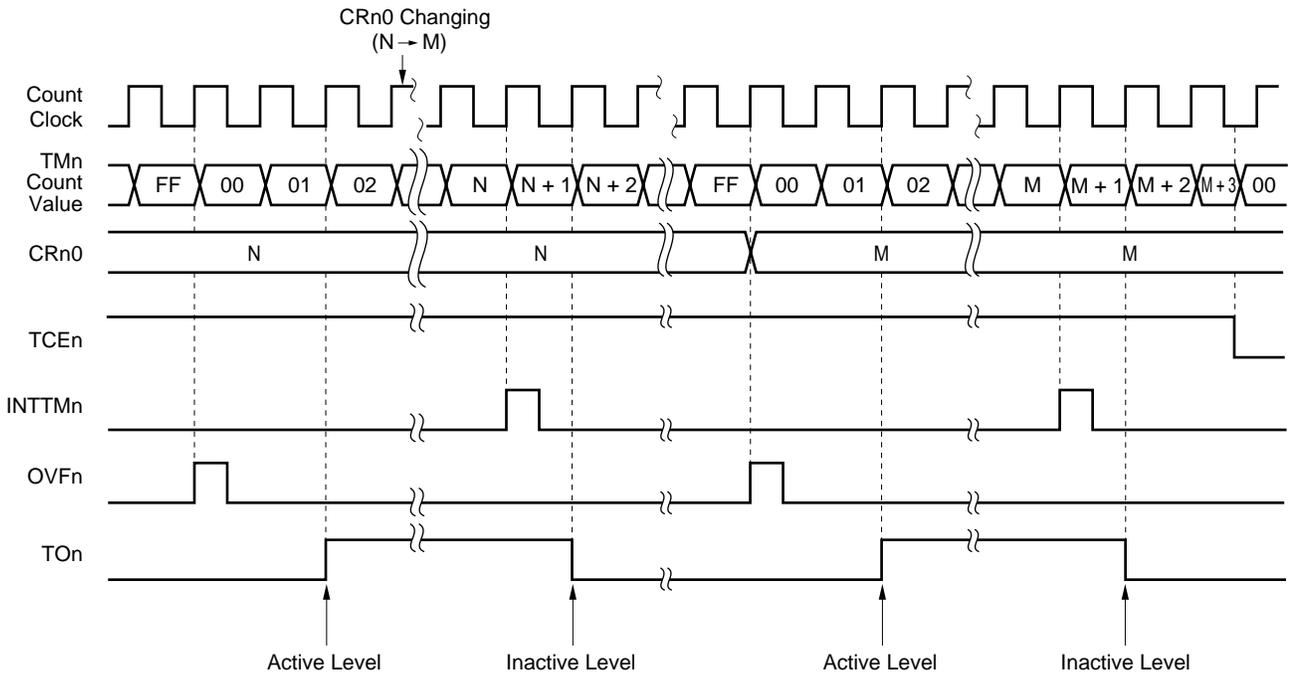
Remark $n = 5, 6$

Figure 10-17. PWM Output Operation Timings (CRn0 = FFH, Active High Setting)



Remark n = 5, 6

Figure 10-18. PWM Output Operation Timings (CRn0 Changing, Active High Setting)



Remark n = 5, 6

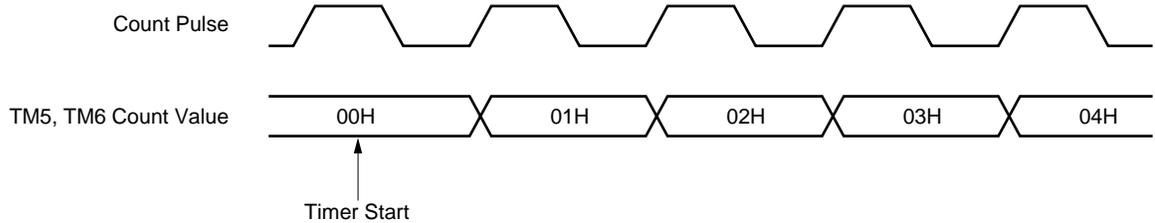
Caution If CRn0 is changed during TMn operation, the value changed is not reflected until TMn overflows.

10.5 8-Bit Timer/Event Counters 5 and 6 Precautions

(1) Timer start errors

An error with a maximum of one clock might occur concerning the time required for a match signal to be generated after the timer starts. This is because 8-bit timer registers 5 and 6 are started asynchronously with the count pulse.

Figure 10-19. 8-bit Timer Registers 5 and 6 Start Timings

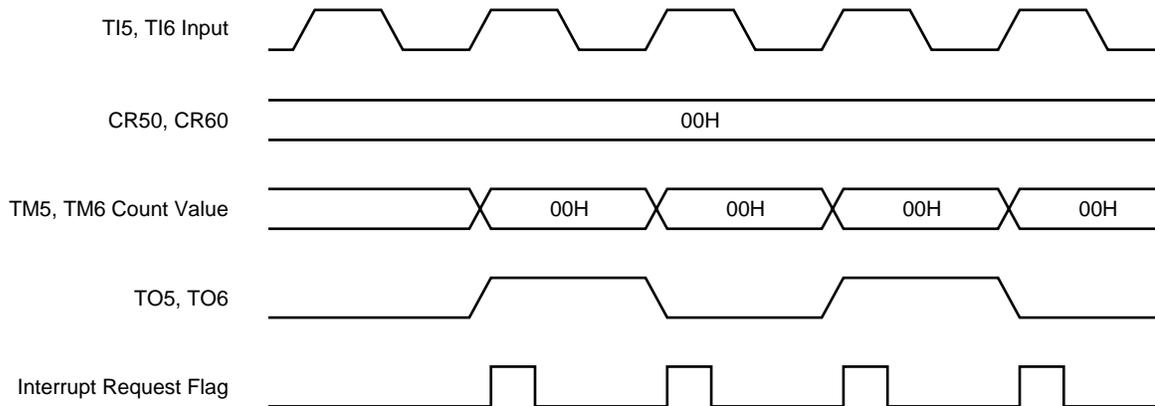


(2) Compare registers 50 and 60 sets

The 8-bit compare registers (CR50 and CR60) can be set to 00H.

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

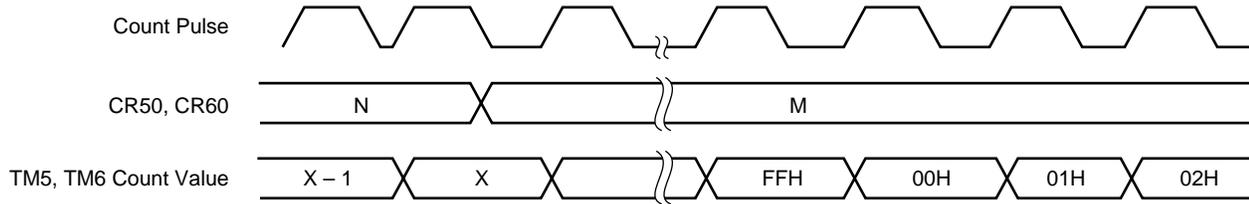
Figure 10-20. External Event Counter Operation Timings



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

If the values after the 8-bit compare registers (CR50 and CR60) are changed are smaller than those of 8-bit timer registers (TM5 and TM6), TM5 and TM6 continue counting, overflow and then restarts counting from 0. Thus, if the value (M) after CR50 and CR60 change is smaller than that (N) before change it is necessary to restart the timer after changing CR50 and CR60.

Figure 10-21. Timings after Compare Register Change during Timer Count Operation



Remark $N > X > M$

CHAPTER 11 WATCH TIMER

11.1 Watch Timer Functions

The watch timer has the following functions.

- Watch timer
- Interval timer

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

(1) Watch timer

When the 32.768-kHz subsystem clock is used, a flag (WTIF) is set at 0.5-second or 0.25-second intervals. When the 4.19-MHz (standard: 4.194304 MHz) main system clock is used, a flag (WTIF) is set at 0.5-second or 0.25-second intervals.

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

(2) Interval timer

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

Table 11-1. Interval Timer Interval Time

Interval Time	When operated at $f_{xx} = 5.0 \text{ MHz}$	When operated at $f_{xx} = 4.19 \text{ MHz}$	When operated at $f_{xT} = 32.768 \text{ kHz}$
$2^4 \times 1/f_w$	410 μs	488 μs	488 μs
$2^5 \times 1/f_w$	819 μs	977 μs	977 μs
$2^6 \times 1/f_w$	1.64 ms	1.95 ms	1.95 ms
$2^7 \times 1/f_w$	3.28 ms	3.91 ms	3.91 ms
$2^8 \times 1/f_w$	6.55 ms	7.81 ms	7.81 ms
$2^9 \times 1/f_w$	13.1 ms	15.6 ms	15.6 ms

f_{xx} : Main system clock frequency (f_x or $f_x/2$)

f_x : Main system clock oscillation frequency

f_{xT} : Subsystem clock oscillation frequency

f_w : Watch timer clock frequency ($f_{xx}/2^7$ or f_{xT})

11.2 Watch Timer Configuration

The watch timer consists of the following hardware.

Table 11-2. Watch Timer Configuration

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

11.3 Watch Timer Control Registers

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

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

(1) Timer clock select register 2 (TCL2)

This register sets the watch timer count clock.

TCL2 is set with an 8-bit memory manipulation instruction.

RESET input sets TCL2 to 00H.

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

Figure 11-1. Watch Timer Block Diagram

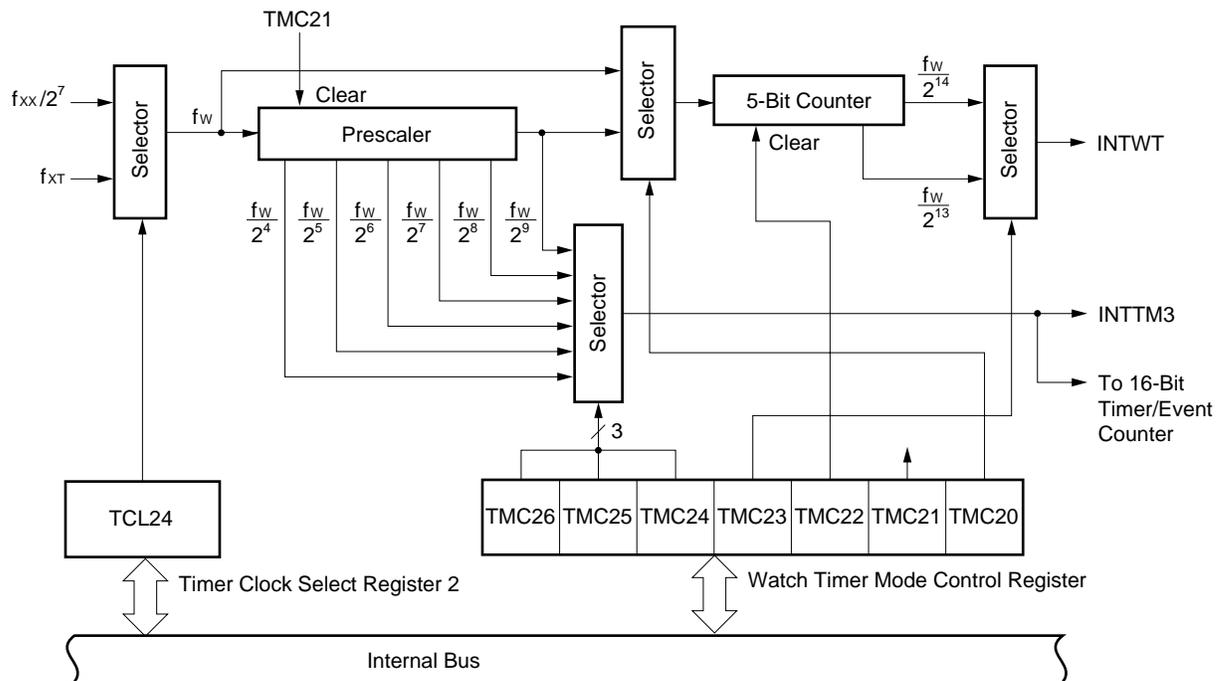


Figure 11-2. Timer Clock Select Register 2 Format

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

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

TCL24		Watch Timer Count Clock Selection		
		MCS = 1		MCS = 0
0	$f_{xx}/2^7$	$f_x/2^7$ (39.1 kHz)		$f_x/2^8$ (19.5 kHz)
1	f_{XT} (32.768 kHz)			

TCL27			TCL26			TCL25			Buzzer Output Frequency Selection			
						MCS = 1			MCS = 0			
0	x	x	Buzzer output disable									
1	0	0	$f_{xx}/2^9$				$f_x/2^9$ (9.8 kHz)			$f_x/2^{10}$ (4.9 kHz)		
1	0	1	$f_{xx}/2^{10}$				$f_x/2^{10}$ (4.9 kHz)			$f_x/2^{11}$ (2.4 kHz)		
1	1	0	$f_{xx}/2^{11}$				$f_x/2^{11}$ (2.4 kHz)			$f_x/2^{12}$ (1.2 kHz)		
1	1	1	Setting prohibited									

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

- Remarks**
1. f_{xx} : Main system clock frequency (f_x or $f_x/2$)
 2. f_x : Main system clock oscillation frequency
 3. f_{XT} : Subsystem clock oscillation frequency
 4. x : Don't care
 5. MCS : Bit 0 of oscillation mode selection register (OSMS)
 6. Figures in parentheses apply to operation with $f_x = 5.0$ MHz or $f_{XT} = 32.768$ kHz.

(2) Watch timer mode control register (TMC2)

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

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

Figure 11-3. Watch Timer Mode Control Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TMC2	0	TMC26	TMC25	TMC24	TMC23	TMC22	TMC21	TMC20	FF4AH	00H	R/W
TMC20	Watch Operating Mode Selection										
0	Normal operating mode (flag set at $f_w/2^{14}$)										
1	Fast feed operating mode (flag set at $f_w/2^5$)										
TMC21	Prescaler Operation Control										
0	Clear after operation stop										
1	Operation enable										
TMC22	5-Bit Counter Operation Control										
0	Clear after operation stop										
1	Operation enable										
TMC23	Watch Flag Set Time Selection										
	$f_{XX} = 5.0 \text{ MHz Operation}$				$f_{XX} = 4.19 \text{ MHz Operation}$				$f_{XT} = 32.768 \text{ kHz Operation}$		
0	$2^{14}/f_w$ (0.4 sec)				$2^{14}/f_w$ (0.5 sec)				$2^{14}/f_w$ (0.5 sec)		
1	$2^{13}/f_w$ (0.2 sec)				$2^{13}/f_w$ (0.25 sec)				$2^{13}/f_w$ (0.25 sec)		
TMC26	TMC25	TMC24	Prescaler Interval Time Selection								
			$f_{XX} = 5.0 \text{ MHz Operation}$			$f_{XX} = 4.19 \text{ MHz Operation}$			$f_{XT} = 32.768 \text{ kHz Operation}$		
0	0	0	$2^4/f_w$ (410 μs)			$2^4/f_w$ (488 μs)			$2^4/f_w$ (488 μs)		
0	0	1	$2^5/f_w$ (819 μs)			$2^5/f_w$ (977 μs)			$2^5/f_w$ (977 μs)		
0	1	0	$2^6/f_w$ (1.64 ms)			$2^6/f_w$ (1.95 ms)			$2^6/f_w$ (1.95 ms)		
0	1	1	$2^7/f_w$ (3.28 ms)			$2^7/f_w$ (3.91 ms)			$2^7/f_w$ (3.91 ms)		
1	0	0	$2^8/f_w$ (6.55 ms)			$2^8/f_w$ (7.81 ms)			$2^8/f_w$ (7.81 ms)		
1	0	1	$2^9/f_w$ (13.1 ms)			$2^9/f_w$ (15.6 ms)			$2^9/f_w$ (15.6 ms)		
Other than above			Setting prohibited								

Caution When the watch timer is used, the prescaler should not be cleared frequently.

- Remarks**
- f_w : Watch timer clock frequency ($f_{XX}/2^7$ or f_{XT})
 - f_{XX} : Main system clock frequency (f_x or $f_x/2$)
 - f_x : Main system clock oscillation frequency
 - f_{XT} : Subsystem clock oscillation frequency

11.4 Watch Timer Operations

11.4.1 Watch timer operation

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

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

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

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

11.4.2 Interval timer operation

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

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

Table 11-3. Interval Timer Interval Time

TMC26	TMC25	TMC24	Interval Time	When operated at $f_{xx} = 5.0$ MHz	When operated at $f_{xx} = 4.19$ MHz	When operated at $f_{XT} = 32.768$ kHz
0	0	0	$2^4 \times 1/f_w$	410 μ s	488 μ s	488 μ s
0	0	1	$2^5 \times 1/f_w$	819 μ s	977 μ s	977 μ s
0	1	0	$2^6 \times 1/f_w$	1.64 ms	1.95 ms	1.95 ms
0	1	1	$2^7 \times 1/f_w$	3.28 ms	3.91 ms	3.91 ms
1	0	0	$2^8 \times 1/f_w$	6.55 ms	7.81 ms	7.81 ms
1	0	1	$2^9 \times 1/f_w$	13.1 ms	15.6 ms	15.6 ms
Other than above			Setting prohibited			

f_{xx} : Main system clock frequency (fx or fx/2)

fx : Main system clock oscillation frequency

f_{XT} : Subsystem clock oscillation frequency

f_w : Watch timer clock frequency ($f_{xx}/2^7$ or f_{XT})

CHAPTER 12 WATCHDOG TIMER

12.1 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 the interval timer cannot be used simultaneously).

(1) Watchdog timer mode

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

Table 12-1. Watchdog Timer Runaway Times

Runaway Detection Time	MCS = 1	MCS = 0
$2^{11} \times 1/f_{XX}$	$2^{11} \times 1/f_x$ (410 μs)	$2^{12} \times 1/f_x$ (819 μs)
$2^{12} \times 1/f_{XX}$	$2^{12} \times 1/f_x$ (819 μs)	$2^{13} \times 1/f_x$ (1.64 ms)
$2^{13} \times 1/f_{XX}$	$2^{13} \times 1/f_x$ (1.64 ms)	$2^{14} \times 1/f_x$ (3.28 ms)
$2^{14} \times 1/f_{XX}$	$2^{14} \times 1/f_x$ (3.28 ms)	$2^{15} \times 1/f_x$ (6.55 ms)
$2^{15} \times 1/f_{XX}$	$2^{15} \times 1/f_x$ (6.55 ms)	$2^{16} \times 1/f_x$ (13.1 ms)
$2^{16} \times 1/f_{XX}$	$2^{16} \times 1/f_x$ (13.1 ms)	$2^{17} \times 1/f_x$ (26.2 ms)
$2^{17} \times 1/f_{XX}$	$2^{17} \times 1/f_x$ (26.2 ms)	$2^{18} \times 1/f_x$ (52.4 ms)
$2^{19} \times 1/f_{XX}$	$2^{19} \times 1/f_x$ (104.9 ms)	$2^{20} \times 1/f_x$ (209.7 ms)

- Remarks**
1. f_{XX} : Main system clock frequency (f_x or $f_x/2$)
 2. f_x : Main system clock oscillation frequency
 3. MCS : Bit 0 of oscillation mode selection register (OSMS)
 4. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

(2) Interval timer mode

Interrupt requests are generated at the preset time intervals.

Table 12-2. Interval Times

Interval Time	MCS = 1	MCS = 0
$2^{11} \times 1/f_{XX}$	$2^{11} \times 1/f_x$ (410 μ s)	$2^{12} \times 1/f_x$ (819 μ s)
$2^{12} \times 1/f_{XX}$	$2^{12} \times 1/f_x$ (819 μ s)	$2^{13} \times 1/f_x$ (1.64 ms)
$2^{13} \times 1/f_{XX}$	$2^{13} \times 1/f_x$ (1.64 ms)	$2^{14} \times 1/f_x$ (3.28 ms)
$2^{14} \times 1/f_{XX}$	$2^{14} \times 1/f_x$ (3.28 ms)	$2^{15} \times 1/f_x$ (6.55 ms)
$2^{15} \times 1/f_{XX}$	$2^{15} \times 1/f_x$ (6.55 ms)	$2^{16} \times 1/f_x$ (13.1 ms)
$2^{16} \times 1/f_{XX}$	$2^{16} \times 1/f_x$ (13.1 ms)	$2^{17} \times 1/f_x$ (26.2 ms)
$2^{17} \times 1/f_{XX}$	$2^{17} \times 1/f_x$ (26.2 ms)	$2^{18} \times 1/f_x$ (52.4 ms)
$2^{19} \times 1/f_{XX}$	$2^{19} \times 1/f_x$ (104.9 ms)	$2^{20} \times 1/f_x$ (209.7 ms)

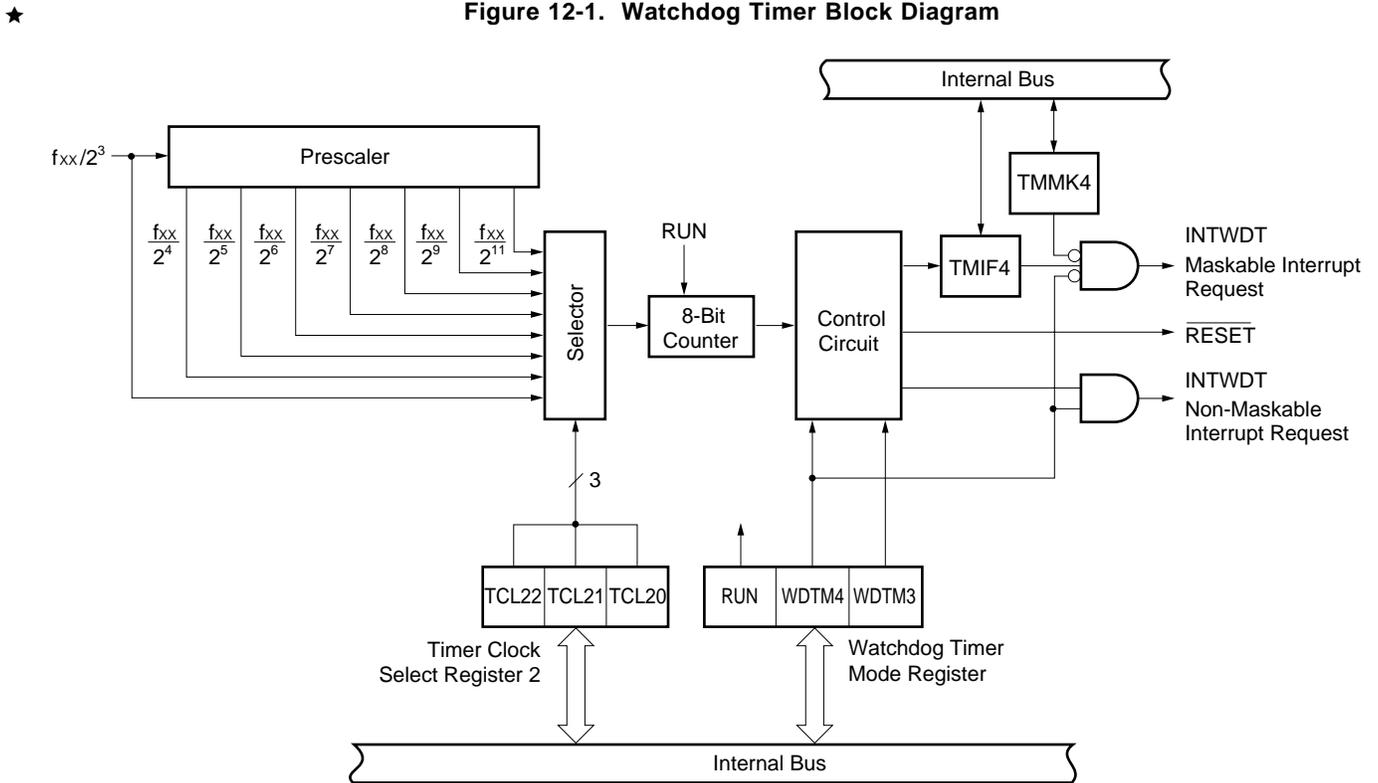
- Remarks**
1. f_{XX} : Main system clock frequency (f_x or $f_x/2$)
 2. f_x : Main system clock oscillation frequency
 3. MCS : Bit 0 of oscillation mode selection register (OSMS)
 4. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

12.2 Watchdog Timer Configuration

The watchdog timer consists of the following hardware.

Table 12-3. Watchdog Timer Configuration

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



12.3 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 watch timer count clock and buzzer output frequency.

Figure 12-2. Timer Clock Select Register 2 Format

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

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

TCL24	Watch Timer Count Clock Selection		
		MCS = 1	MCS = 0
0	$f_{xx}/2^7$	$f_x/2^7$ (39.1 kHz)	$f_x/2^8$ (19.5 kHz)
1	f_{XT} (32.768 kHz)		

TCL27	TCL26	TCL25	Buzzer Output Frequency Selection		
				MCS = 1	MCS = 0
0	x	x	Buzzer output disable		
1	0	0	$f_{xx}/2^9$	$f_x/2^9$ (9.8 kHz)	$f_x/2^{10}$ (4.9 kHz)
1	0	1	$f_{xx}/2^{10}$	$f_x/2^{10}$ (4.9 kHz)	$f_x/2^{11}$ (2.4 kHz)
1	1	0	$f_{xx}/2^{11}$	$f_x/2^{11}$ (2.4 kHz)	$f_x/2^{12}$ (1.2 kHz)
1	1	1	Setting prohibited		

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

- Remarks**
1. f_{xx} : Main system clock frequency (f_x or $f_x/2$)
 2. f_x : Main system clock oscillation frequency
 3. f_{XT} : Subsystem clock oscillation frequency
 4. x : Don't care
 5. MCS : Bit 0 of oscillation mode selection register (OSMS)
 6. Figures in parentheses apply to operation with $f_x = 5.0$ MHz or $f_{XT} = 32.768$ kHz.

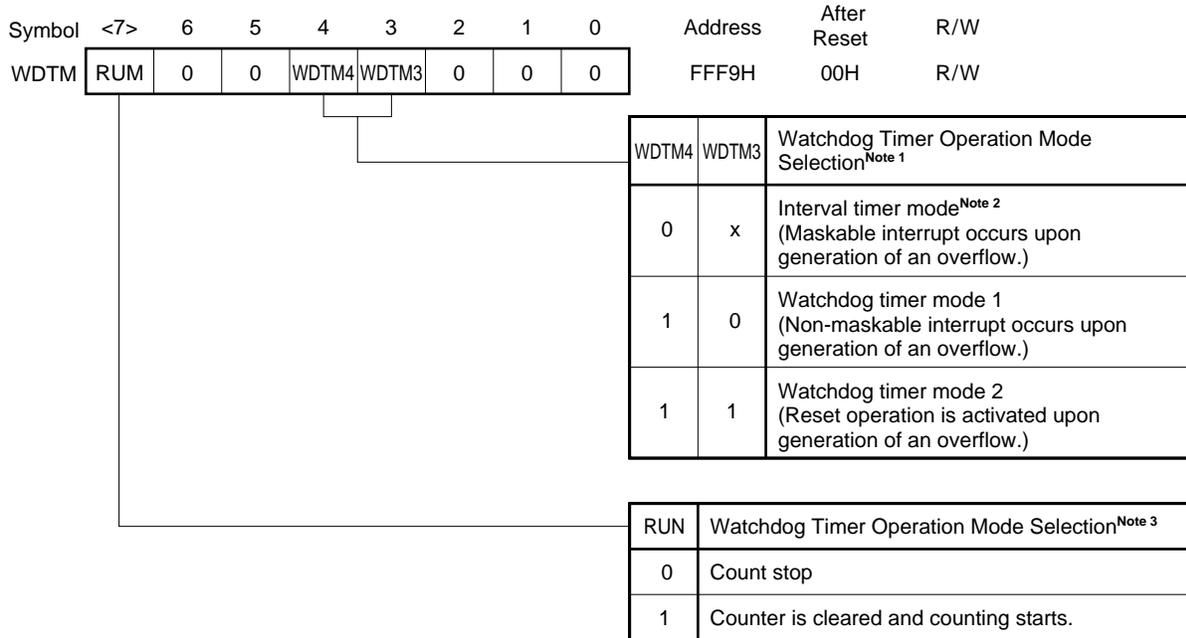
(2) Watchdog timer mode register (WDTM)

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

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

$\overline{\text{RESET}}$ input sets WDTM to 00H.

Figure 12-3. Watchdog Timer Mode Register Format



- Notes**
- Once set to 1, WDTM3 and WDTM4 cannot be cleared to 0 by software.
 - Operation as an interval timer is started as soon as is set to 1.
 - Once set to 1, RUN cannot be cleared to 0 by software.
Thus, once counting starts, it can only be stopped by $\overline{\text{RESET}}$ input.

- Cautions**
- 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 (TCL2).
 - When using the watchdog timer modes 1 and 2, be sure that the interrupt request flag (TMIF4) is 0 before setting WDTM4 to 1. If WDTM4 is set to 1 while TMIF4 is 1, a non-maskable interrupt requests generated regardless of the contents of WDTM3.

Remark x : Don't care

12.4 Watchdog Timer Operations

12.4.1 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 runaway.

The watchdog timer count clock (runaway 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 runaway detection 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 runaway 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 can be cleared when RUN is set to 1.

The watchdog timer continues operating in the HALT mode but it stops in the STOP mode. Thus, set RUN to 1 before the STOP mode is set, clear the watchdog timer and then execute the STOP instruction.

- Cautions**
1. The actual runaway detection time may be shorter than the set time by a maximum of 0.5 %.
 2. When the subsystem clock is selected for CPU clock, watchdog timer count operation is stopped.

Table 12-4. Watchdog Timer Runaway Detection Time

TCL22	TCL21	TCL20	Runaway Detection Time	MCS = 1	MCS = 0
0	0	0	$2^{11} \times 1/f_{xx}$	$2^{11} \times 1/f_x$ (410 μ s)	$2^{12} \times 1/f_x$ (819 μ s)
0	0	1	$2^{12} \times 1/f_{xx}$	$2^{12} \times 1/f_x$ (819 μ s)	$2^{13} \times 1/f_x$ (1.64 ms)
0	1	0	$2^{13} \times 1/f_{xx}$	$2^{13} \times 1/f_x$ (1.64 ms)	$2^{14} \times 1/f_x$ (3.28 ms)
0	1	1	$2^{14} \times 1/f_{xx}$	$2^{14} \times 1/f_x$ (3.28 ms)	$2^{15} \times 1/f_x$ (6.55 ms)
1	0	0	$2^{15} \times 1/f_{xx}$	$2^{15} \times 1/f_x$ (6.55 ms)	$2^{16} \times 1/f_x$ (13.1 ms)
1	0	1	$2^{16} \times 1/f_{xx}$	$2^{16} \times 1/f_x$ (13.1 ms)	$2^{17} \times 1/f_x$ (26.2 ms)
1	1	0	$2^{17} \times 1/f_{xx}$	$2^{17} \times 1/f_x$ (26.2 ms)	$2^{18} \times 1/f_x$ (52.4 ms)
1	1	1	$2^{19} \times 1/f_{xx}$	$2^{19} \times 1/f_x$ (104.9 ms)	$2^{20} \times 1/f_x$ (209.7 ms)

- Remarks**
1. f_{xx} : Main system clock frequency (f_x or $f_x/2$)
 2. f_x : Main system clock oscillation frequency
 3. MCS : Bit 0 of oscillation mode selection register (OSMS)
 4. TCL20 to TCL22 : Bits 0 to 2 of timer clock select register 2 (TCL2)
 5. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

12.4.2 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, respectively.

The count clock (interval time) can be selected with bits 0 to 2 (TCL20 to TCL22) of the timer clock select register 2 (TCL2). The watchdog timer starts operation as an interval timer when bit 7 (RUN) of WDTM is set to 1.

When the watchdog timer operated as interval timer, the interrupt mask flag (TMMK4) and priority specify flag (TMPR4) are validated and the maskable interrupt request (INTWDT) can be generated. Among maskable interrupt requests, the INTWDT default has the highest priority.

The interval timer continues operating in the HALT mode but it stops in STOP mode. Thus, set bit 7 (RUN) of WDTM to 1 before the STOP mode is set, clear the interval timer and then execute the STOP instruction.

- Cautions**
1. Once bit 4 (WDTM4) of WDTM is set to 1 (with the watchdog timer mode selected), the interval timer mode is not set unless $\overline{\text{RESET}}$ input is applied.
 2. The interval time just after setting with WDTM may be shorter than the set time by a maximum of 0.5 %.
 3. When the subsystem clock is selected for CPU clock, watchdog timer count operation is stopped.

Table 12-5. Interval Timer Interval Time

TCL22	TCL21	TCL20	Interval Time	MCS = 1	MCS = 0
0	0	0	$2^{11} \times 1/f_{XX}$	$2^{11} \times 1/f_x$ (410 μs)	$2^{12} \times 1/f_x$ (819 μs)
0	0	1	$2^{12} \times 1/f_{XX}$	$2^{12} \times 1/f_x$ (819 μs)	$2^{13} \times 1/f_x$ (1.64 ms)
0	1	0	$2^{13} \times 1/f_{XX}$	$2^{13} \times 1/f_x$ (1.64 ms)	$2^{14} \times 1/f_x$ (3.28 ms)
0	1	1	$2^{14} \times 1/f_{XX}$	$2^{14} \times 1/f_x$ (3.28 ms)	$2^{15} \times 1/f_x$ (6.55 ms)
1	0	0	$2^{15} \times 1/f_{XX}$	$2^{15} \times 1/f_x$ (6.55 ms)	$2^{16} \times 1/f_x$ (13.1 ms)
1	0	1	$2^{16} \times 1/f_{XX}$	$2^{16} \times 1/f_x$ (13.1 ms)	$2^{17} \times 1/f_x$ (26.2 ms)
1	1	0	$2^{17} \times 1/f_{XX}$	$2^{17} \times 1/f_x$ (26.2 ms)	$2^{18} \times 1/f_x$ (52.4 ms)
1	1	1	$2^{19} \times 1/f_{XX}$	$2^{19} \times 1/f_x$ (104.9 ms)	$2^{20} \times 1/f_x$ (209.7 ms)

- Remarks**
1. f_{XX} : Main system clock frequency (f_x or $f_x/2$)
 2. f_x : Main system clock oscillation frequency
 3. MCS : Bit 0 of oscillation mode selection register (OSMS)
 4. TCL20 to TCL22 : Bits 0 to 2 of timer clock select register 2 (TCL2)
 5. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

CHAPTER 13 CLOCK OUTPUT CONTROL CIRCUIT

13.1 Clock Output Control Circuit Functions

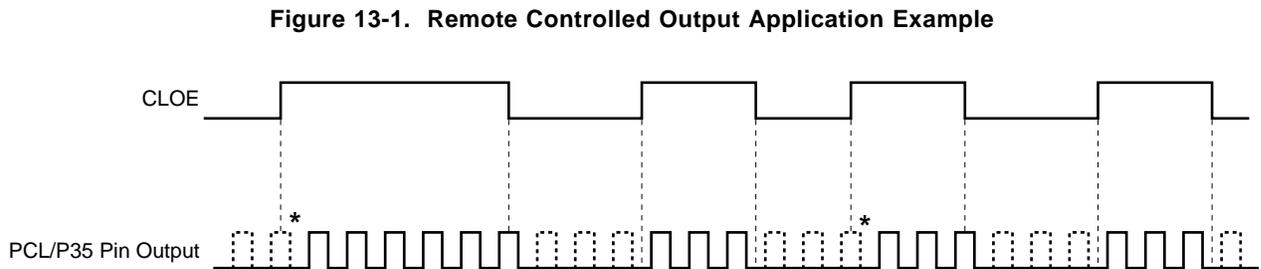
The clock output control circuit is intended for carrier output during remote controlled transmission and clock output for supply to peripheral LSI. Clocks selected with the timer clock select register 0 (TCL0) are output from the PCL/P35 pin.

Follow the procedure below to output clock pulses.

- (1) Select the clock pulse output frequency (with clock pulse output disabled) with bits 0 to 3 (TCL00 to TCL03) of TCL0.
- (2) Set the P35 output latch to 0.
- (3) Set bit 5 (PM35) of port mode register 3 (PM3) to 0 (set to output mode).
- (4) Set bit 7 (CLOE) of timer clock select register 0 (TCL0) to 1.

Caution Clock output cannot be used when setting P35 output latch to 1.

Remark When clock output enable/disable is switched, the clock output control circuit does not output pulses with small widths (See the portions marked with * in Figure 13-1).



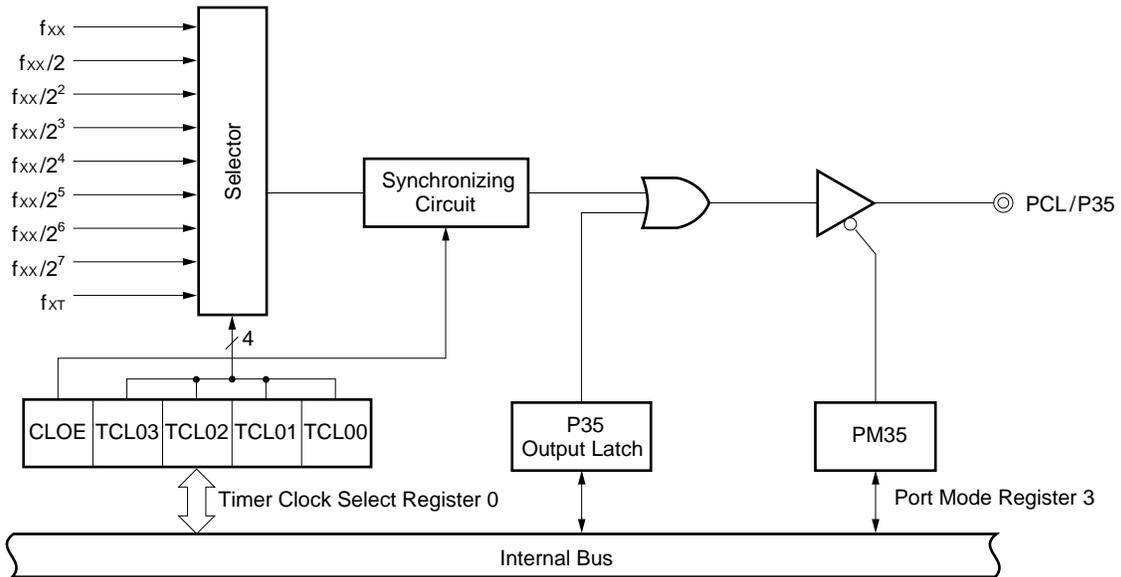
13.2 Clock Output Control Circuit Configuration

The clock output control circuit consists of the following hardware.

Table 13-1. Clock Output Control Circuit Configuration

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

Figure 13-2. Clock Output Control Circuit Block Diagram



13.3 Clock Output Function Control Registers

The following two types of registers are used to control the clock output function.

- Timer clock select register 0 (TCL0)
- Port mode register 3 (PM3)

(1) Timer clock select register 0 (TCL0)

This register sets PCL output clock.

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

RESET input sets TCL0 to 00H.

Remark Besides setting PCL output clock, TCL0 sets the 16-bit timer register count clock.

Figure 13-3. Timer Clock Select Register 0 Format

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

TCL03	TCL02	TCL01	TCL00	PCL Output Clock Selection		
				MCS = 1		MCS = 0
0	0	0	0	f_{XT} (32.768 kHz)		
0	1	0	1	f_{XX}	f_x (5.0 MHz)	$f_x/2$ (2.5 MHz)
0	1	1	0	$f_{XX}/2$	$f_x/2$ (2.5 MHz)	$f_x/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_x/2^2$ (1.25 MHz)	$f_x/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_x/2^3$ (625 kHz)	$f_x/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_x/2^4$ (313 kHz)	$f_x/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_x/2^5$ (156 kHz)	$f_x/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_x/2^6$ (78.1 kHz)	$f_x/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_x/2^7$ (39.1 kHz)	$f_x/2^8$ (19.5 kHz)
Other than above				Setting prohibited		

TCL06	TCL05	TCL04	16-Bit Timer Register Count Clock Selection			
			MCS = 1			MCS = 0
0	0	0	TI00 (Valid edge specifiable)			
0	0	1	$2f_{XX}$	Setting prohibited		f_x (5.0 MHz)
0	1	0	f_{XX}	f_x (5.0 MHz)	$f_x/2$ (2.5 MHz)	
0	1	1	$f_{XX}/2$	$f_x/2$ (2.5 MHz)	$f_x/2^2$ (1.25 MHz)	
1	0	0	$f_{XX}/2^2$	$f_x/2^2$ (1.25 MHz)	$f_x/2^3$ (625 kHz)	
1	1	1	Watch Timer Output (INTTM3)			
Other than above			Setting prohibited			

CLOE	PCL Output Control
0	Output disable
1	Output enable

- Cautions**
1. The TI00/P00/INTP0 pin valid edge is set by external interrupt mode register 0 (INTM0), and the sampling clock frequency is selected by the sampling clock selection register (SCS).
 2. When enabling PCL output, set TCL00 to TCL03, then set 1 in CLOE with a 1-bit memory manipulation instruction.
 3. To read the count value when TI00 has been specified as the TM0 count clock, the value should be read from TM0, not from capture/compare register 01 (CR01).
 4. When rewriting TCL0 to other data, stop the clock operation beforehand.

- Remarks**
1. f_{xx} : Main system clock frequency (f_x or $f_x/2$)
 2. f_x : Main system clock oscillation frequency
 3. f_{xT} : Subsystem clock oscillation frequency
 4. $TI00$: 16-bit timer/event counter input pin
 5. $TM0$: 16-bit timer register
 6. MCS : Bit 0 of oscillation mode selection register (OSMS)
 7. Figures in parentheses apply to operation with $f_x = 5.0$ MHz or $f_{xT} = 32.768$ kHz.

(2) Port mode register 3 (PM3)

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

When using the P35/PCL pin for clock output function, set PM35 and output latch of P35 to 0.

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

\overline{RESET} input sets PM3 to FFH.

Figure 13-4. Port Mode Register 3 Format



[MEMO]

CHAPTER 14 BUZZER OUTPUT CONTROL CIRCUIT

14.1 Buzzer Output Control Circuit Functions

The buzzer output control circuit outputs 1.2-kHz, 2.4-kHz, 4.9-kHz, or 9.8-kHz frequency square waves. The buzzer frequency selected with timer clock select register 2 (TCL2) is output from the BUZ/P36 pin.

Follow the procedure below to output the buzzer frequency.

- (1) Select the buzzer output frequency with bits 5 to 7 (TCL25 to TCL27) of TCL2.
- (2) Set the P36 output latch to 0.
- (3) Set bit 6 (PM36) of port mode register 3 (PM3) to 0 (Set to output mode).

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

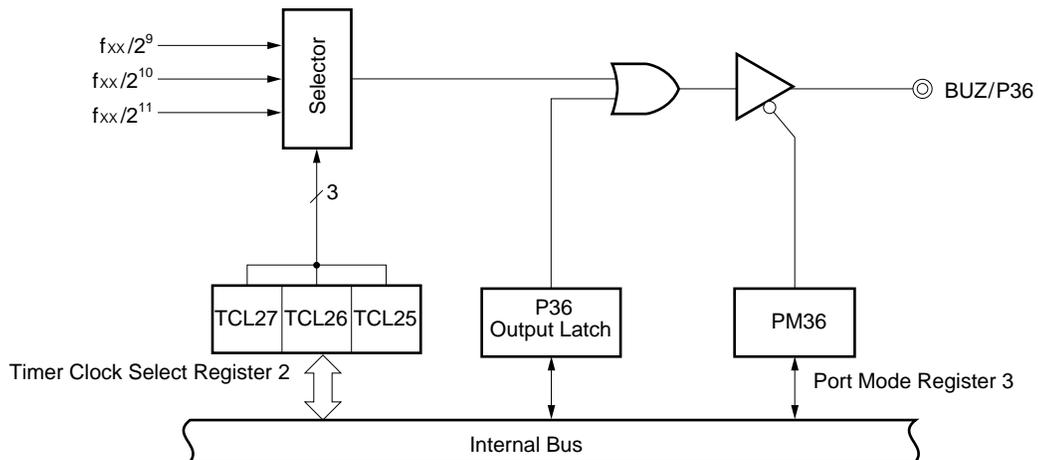
14.2 Buzzer Output Control Circuit Configuration

The buzzer output control circuit consists of the following hardware.

Table 14-1. Buzzer Output Control Circuit Configuration

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

Figure 14-1. Buzzer Output Control Circuit Block Diagram



14.3 Buzzer Output Function Control Registers

The following two types of registers are used to control the buzzer output function.

- Timer clock select register 2 (TCL2)
- Port mode register 3 (PM3)

(1) Timer clock select register 2 (TCL2)

This register sets the buzzer output frequency.

TCL2 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TCL2 to 00H.

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

Figure 14-2. Timer Clock Select Register 2 Format

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

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

TCL24		Watch Timer Count Clock Selection	
		MCS = 1	MCS = 0
0	$f_{xx}/2^7$	$f_x/2^7$ (39.1 kHz)	$f_x/2^8$ (19.5 kHz)
1	f_{XT} (32.768 kHz)		

TCL27			TCL26			TCL25			Buzzer Output Frequency Selection		
						MCS = 1			MCS = 0		
0	x	x	Buzzer output disable								
1	0	0	$f_{xx}/2^9$			$f_x/2^9$ (9.8 kHz)			$f_x/2^{10}$ (4.9 kHz)		
1	0	1	$f_{xx}/2^{10}$			$f_x/2^{10}$ (4.9 kHz)			$f_x/2^{11}$ (2.4 kHz)		
1	1	0	$f_{xx}/2^{11}$			$f_x/2^{11}$ (2.4 kHz)			$f_x/2^{12}$ (1.2 kHz)		
1	1	1	Setting prohibited								

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

- Remarks**
1. f_{xx} : Main system clock frequency (f_x or $f_x/2$)
 2. f_x : Main system clock oscillation frequency
 3. f_{XT} : Subsystem clock oscillation frequency
 4. x : Don't care
 5. MCS : Bit 0 of oscillation mode selection register (OSMS)
 6. Figures in parentheses apply to operation with $f_x = 5.0$ MHz or $f_{XT} = 32.768$ kHz.

(2) Port mode register 3 (PM3)

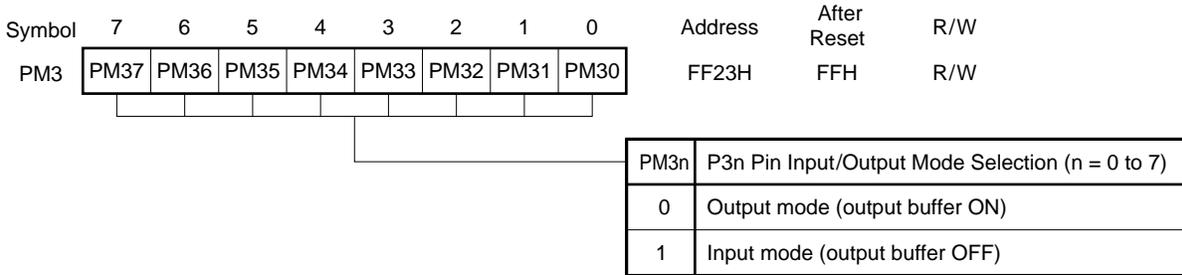
This register sets port 3 input/output in 1-bit units.

When using the P36/BUZ pin for buzzer output function, set PM36 and output latch of P36 to 0.

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

RESET input sets PM3 to FFH.

Figure 14-3. Port Mode Register 3 Format



CHAPTER 15 A/D CONVERTER

15.1 A/D Converter Functions

The A/D converter converts an analog input into a digital value. It consists of 8 channels (ANI0 to ANI7) with an 8-bit resolution.

The conversion method is based on successive approximation and the conversion result is held in the 8-bit A/D conversion result register (ADCR).

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

(1) Hardware start

Conversion is started by trigger input (INTP3).

(2) Software start

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

One channel of analog input is selected from ANI0 to ANI7 and A/D conversion is carried out. In the case of hardware start, A/D conversion operation stops and an interrupt request (INTAD) is generated when an A/D conversion operation ends. 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.

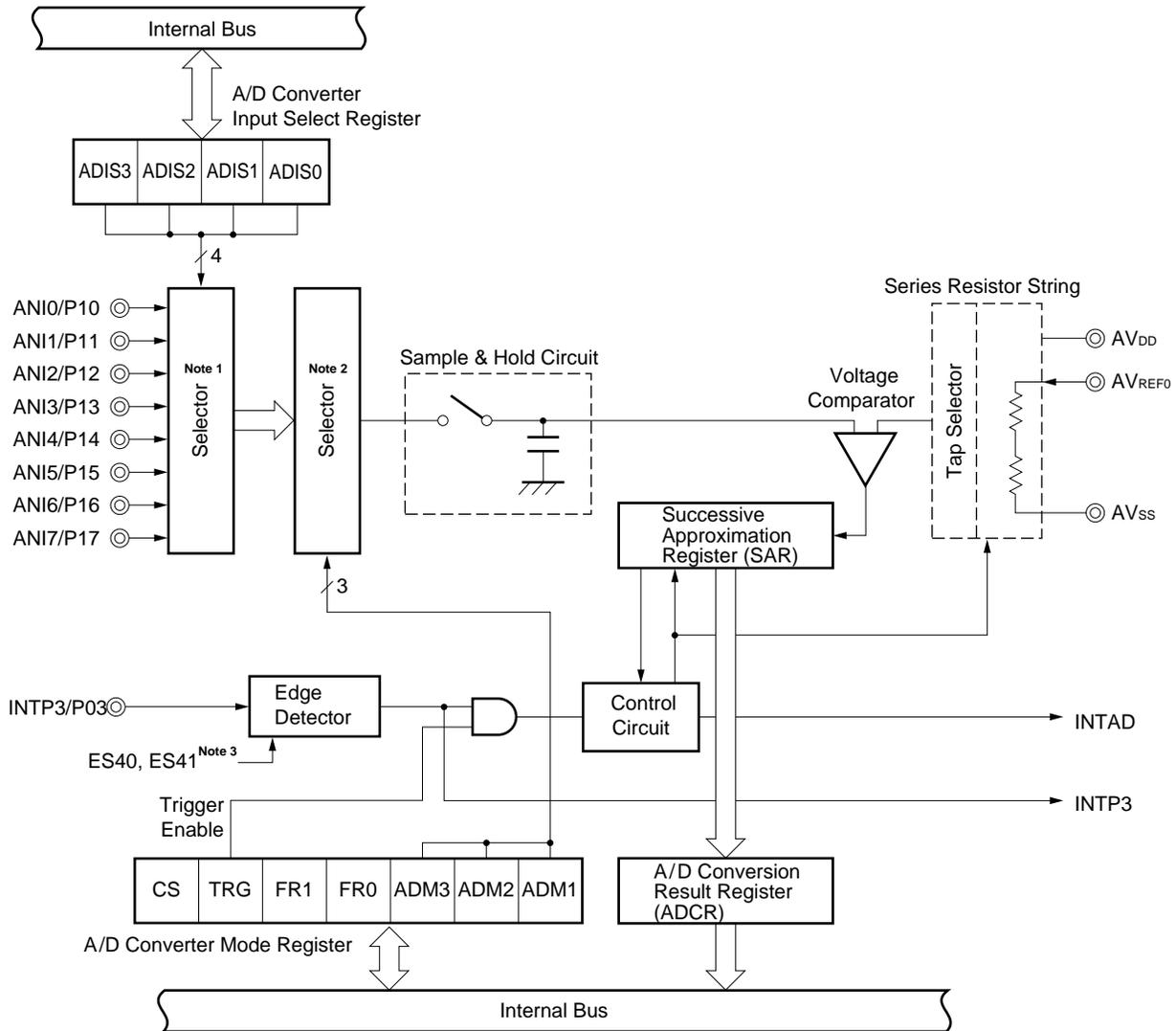
15.2 A/D Converter Configuration

The A/D converter consists of the following hardware.

Table 15-1. A/D Converter Configuration

Item	Configuration
Analog input	8 channels (ANI0 to ANI7)
Control register	A/D converter mode register (ADM) A/D converter input select register (ADIS) External interrupt mode register 1 (INTM1)
Register	Successive approximation register (SAR) A/D conversion result register (ADCR)

Figure 15-1. 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, 1 of external interrupt mode register 1 (INTM1)

(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 the result of comparison is held to the least significant bit (LSB) (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.

ADCR is read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes ADCR undefined.

(3) Sample & hold circuit

The sample & hold circuit samples each analog input signal sequentially applied from the input circuit and sends it to the voltage comparator. This circuit holds the sampled analog input voltage value during A/D conversion.

(4) Voltage comparator

The voltage comparator compares the analog input to the series resistor string output voltage.

(5) Series resistor string

The series resistor string is connected between AV_{REF0} to AV_{SS} and generates a voltage to be compared to the analog input.

(6) ANI0 to ANI7 pins

These are 8-channel analog input pins to input analog signals to undergo A/D conversion to the A/D converter.

Pins other than those selected as analog input by the A/D converter input select register (ADIS) can be used as input/output ports.

Cautions 1. Use ANI0 to ANI7 input voltages within the specified range. If a voltage higher than AV_{REF0} or lower than AV_{SS} is applied (even if within the absolute maximum ratings), the converted value of the corresponding channel becomes indeterminate and may adversely affect the converted values of other channels.

2. Analog input (ANI0 to ANI7) pins are alternate function pins corresponding to input/output port (port 1) pin. When A/D conversion is performed with any one of ANI0 to ANI7 pins selected, do not execute input instruction to port 1 during conversion operation, otherwise, the conversion resolution may be deteriorated.

If digital pulse is applied to a pin adjacent to the pin performing A/D conversion, desired A/D conversion value may not be achieved due to the coupling noise. Do not apply pulse to a pin adjacent to the pin performing A/D conversion.

(7) AV_{REF0} pin

This pin inputs the A/D converter reference voltage.

It converts signals input to ANI0 to ANI7 into digital signals according to the voltage applied between AV_{REF0} and AV_{SS} .

The current flowing in the series resistor string can be reduced by setting the voltage to be input to the AV_{REF0} pin to AV_{SS} level in standby mode.

Caution A series resistor string of approximately 10 k Ω is connected between AV_{REF0} pin and AV_{SS} pin. Therefore, if the output impedance of the reference voltage source is high, AV_{REF0} pin is connected in parallel with the series resistor string between AV_{REF0} pin and AV_{SS} pin. As a result, the reference voltage error will increase.

(8) AV_{SS} pin

This is a GND potential pin of the A/D converter. Keep it at the same potential as the V_{SS} pin when not using the A/D converter.

(9) AV_{DD} pin

This is an A/D converter analog power supply pin. Keep it at the same potential as the V_{SS} pin when not using the A/D converter.

15.3 A/D Converter Control Registers

The following three types of registers are used to control the A/D converter.

- A/D converter mode register (ADM)
- A/D converter input select register (ADIS)
- External interrupt mode register 1 (INTM1)

(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 15-2. A/D Converter Mode Register Format

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After Reset	R/W
ADM	CS	TRG	FR1	FR0	ADM3	ADM2	ADM1	HSC	FF80H	01H	R/W

ADM3	ADM2	ADM1	Analog Input Channel Selection
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

FR1	FR0	HSC	A/D Conversion Time Selection ^{Note 1}			
			f _x = 5.0-MHz Operation		f _x = 4.19-MHz Operation	
			MCS = 1	MCS = 0	MCS = 1	MCS = 0
0	0	1	80/f _x (Setting prohibited ^{Note 2})	160/f _x (32.0 μs)	80/f _x (19.1 μs)	160/f _x (38.1 μs)
0	1	1	40/f _x (Setting prohibited ^{Note 2})	80/f _x (Setting prohibited ^{Note 2})	40/f _x (Setting prohibited ^{Note 2})	80/f _x (19.1 μs)
1	0	0	50/f _x (Setting prohibited ^{Note 2})	100/f _x (20.0 μs)	50/f _x (Setting prohibited ^{Note 2})	100/f _x (23.8 μs)
1	0	1	100/f _x (20.0 μs)	200/f _x (40.0 μs)	100/f _x (23.8 μs)	200/f _x (47.7 μs)
Other combinations			Setting prohibited			

TRG	External Trigger Selection
0	No external trigger (software starts)
1	Conversion started by external trigger (hardware starts)

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

- Notes**
1. Set so that the A/D conversion time is 19.1 μs or more.
 2. Setting prohibited because A/D conversion time is less than 19.1 μs.

- Cautions**
1. 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 : Main system clock oscillation frequency
 2. MCS : Bit 0 of oscillation mode selection register (OSMS)

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

This register determines whether the ANI0/P10 to ANI7/P17 pins should be used for analog input channels or ports. 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 the A/D converter mode register (ADM), select one channel to undergo A/D conversion from among the channels set for analog input with ADIS.

2. No on-chip pull-up resistor can be used for the channels set for analog input with ADIS, irrespective of the value of bit 1 (PUO1) of the pull-up resistor option register L (PUOL).

Figure 15-3. A/D Converter Input Select Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADIS	0	0	0	0	ADIS3	ADIS2	ADIS1	ADIS0	FF84H	00H	R/W

ADIS3	ADIS2	ADIS1	ADIS0	Number of Analog Input Channel Selection
0	0	0	0	No analog input channel (P10 to P17)
0	0	0	1	1 channel (ANI0, P11 to P17)
0	0	1	0	2 channel (ANI0, ANI1, P12 to P17)
0	0	1	1	3 channel (ANI0 to ANI2, P13 to P17)
0	1	0	0	4 channel (ANI0 to ANI3, P14 to P17)
0	1	0	1	5 channel (ANI0 to ANI4, P15 to P17)
0	1	1	0	6 channel (ANI0 to ANI5, P16, P17)
0	1	1	1	7 channel (ANI0 to ANI6, P17)
1	0	0	0	8 channel (ANI0 to ANI7)
Other than above				Setting prohibited

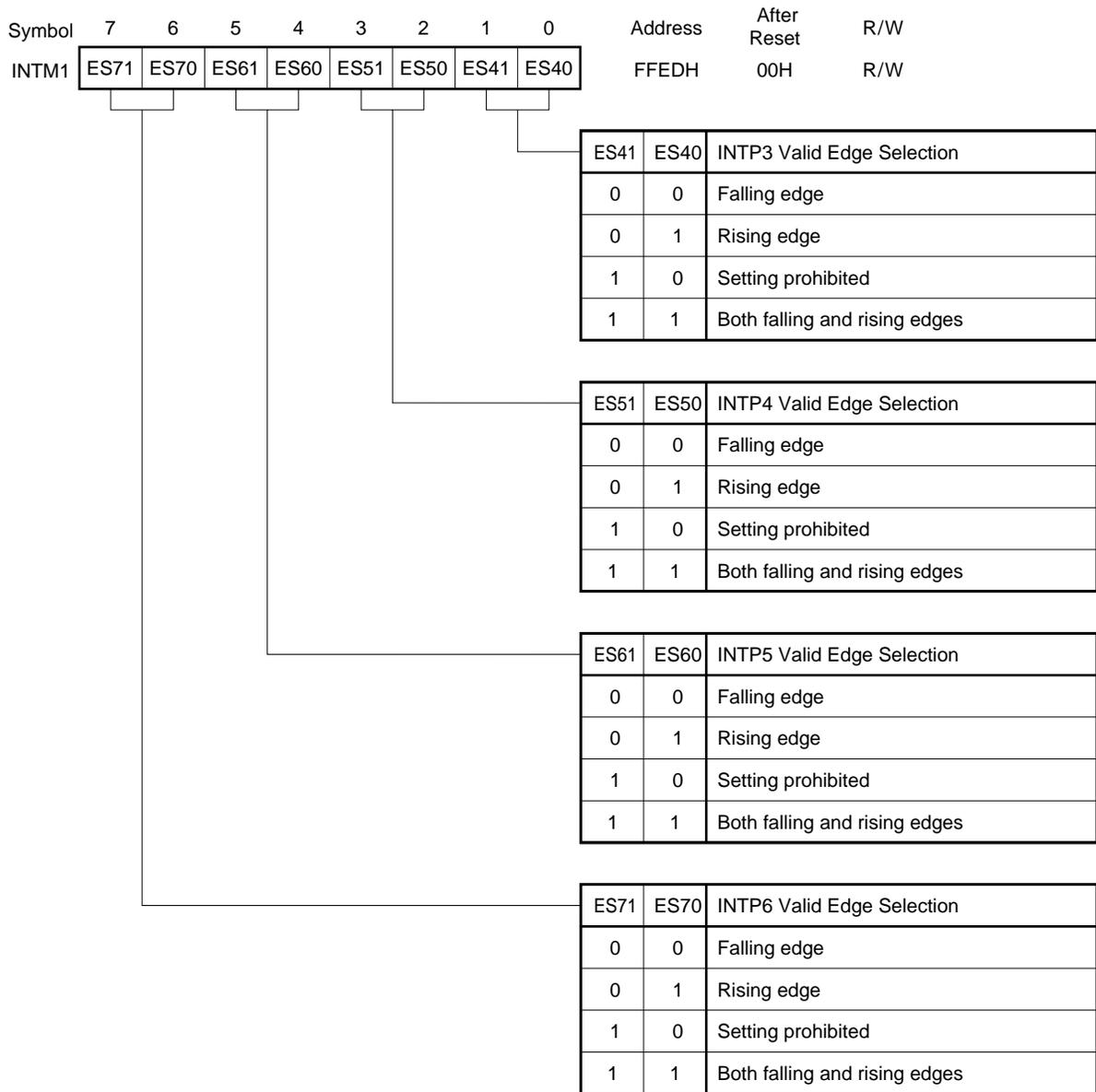
(3) External interrupt mode register 1 (INTM1)

This register sets the valid edge for INTP3 to INTP6.

INTM1 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets INTM1 to 00H.

Figure 15-4. External Interrupt Mode Register 1 Format



15.4 A/D Converter Operations

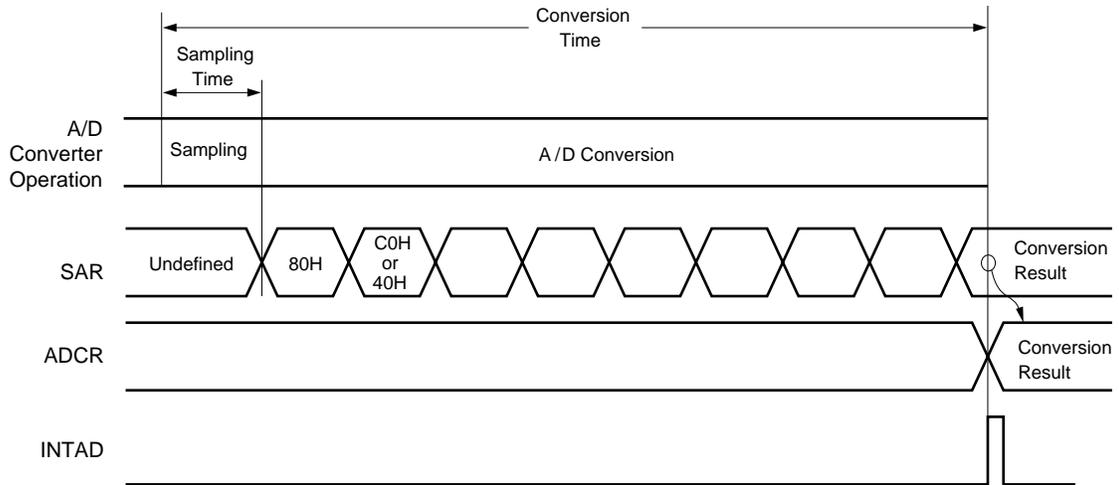
15.4.1 Basic operations of A/D converter

- (1) Set the number of analog input channels with A/D converter input select register (ADIS).
- (2) From among the analog input channels set with ADIS, select one channel for A/D conversion with 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 series resistor string voltage tap to $(1/2) AV_{REF0}$.
- (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 $(1/2) AV_{REF0}$, the MSB of SAR remains set. If the input is smaller than $(1/2) AV_{REF0}$, the MSB is reset.
- (7) Next, bit 6 of SAR is automatically set and the operation proceeds to the next comparison. In this case, the series resistor string voltage tap is selected according to the preset value of bit 7 as described below.
 - Bit 7 = 1 : $(3/4) AV_{REF0}$
 - Bit 7 = 0 : $(1/4) AV_{REF0}$

The voltage tap and analog input voltage are compared and bit 6 of SAR is manipulated with the result as follows.

- Analog input voltage \geq 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 15-5. A/D Converter Basic Operation



A/D conversion operations are performed continuously until bit 7 (CS) of ADM is reset (0) by software.

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

After RESET input, the value of ADCR is undefined.

15.4.2 Input voltage and conversion results

The relation between the analog input voltage input to the analog input pins (ANI0 to ANI7) and the A/D conversion result (the value stored in ADCR) is shown by the following expression.

$$ADCR = \text{INT} \left(\frac{V_{IN}}{AV_{REF0}} \times 256 + 0.5 \right)$$

or

$$(ADCR - 0.5) \times \frac{AV_{REF0}}{256} \leq V_{IN} < (ADCR + 0.5) \times \frac{AV_{REF0}}{256}$$

Where, INT() : Function which returns integer parts of value in parentheses.

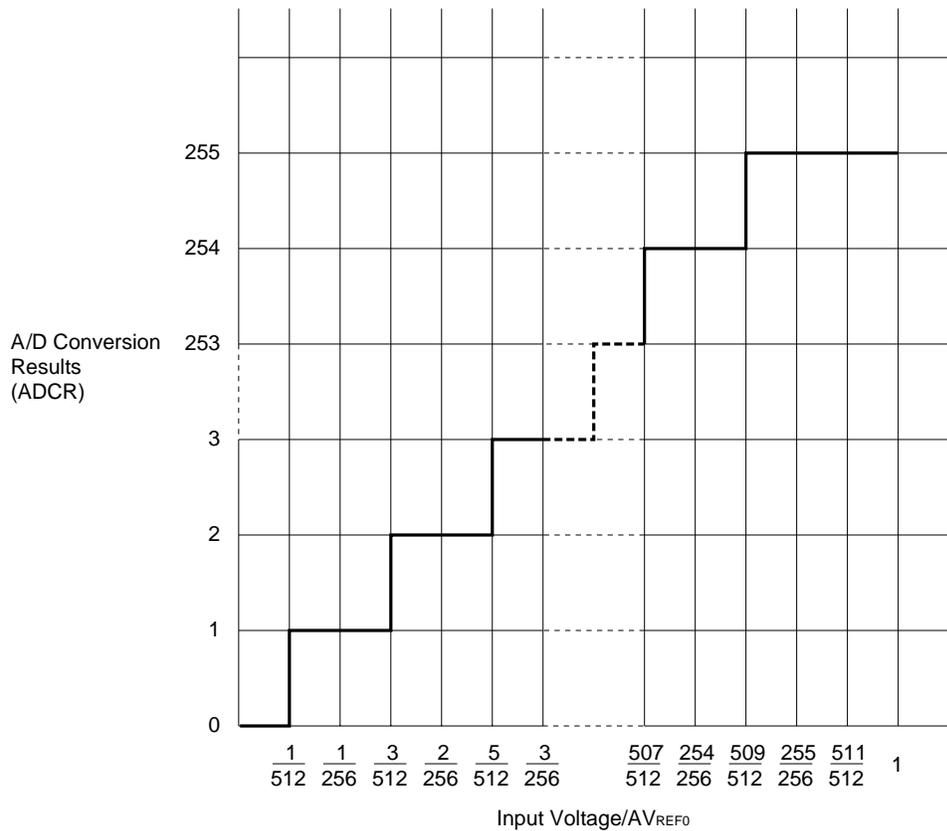
V_{IN} : Analog input voltage

AV_{REF0} : AV_{REF0} pin voltage

ADCR : A/D conversion result register (ADCR)

Figure 15-6 shows the relation between the analog input voltage and the A/D conversion result.

Figure 15-6. Relationships between Analog Input Voltage and A/D Conversion Result



15.4.3 A/D converter operating mode

One analog input channel is selected from among ANI0 to ANI7 with the A/D converter input select register (ADIS) and A/D converter mode register (ADM) and A/D conversion is executed.

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

- Hardware start: Conversion is started by trigger input (INTP3).
- Software start: Conversion is started by setting ADM.

The A/D conversion result is stored in the A/D conversion result register (ADCR) and the interrupt request signal (INTAD) is simultaneously generated.

(1) A/D conversion by hardware start

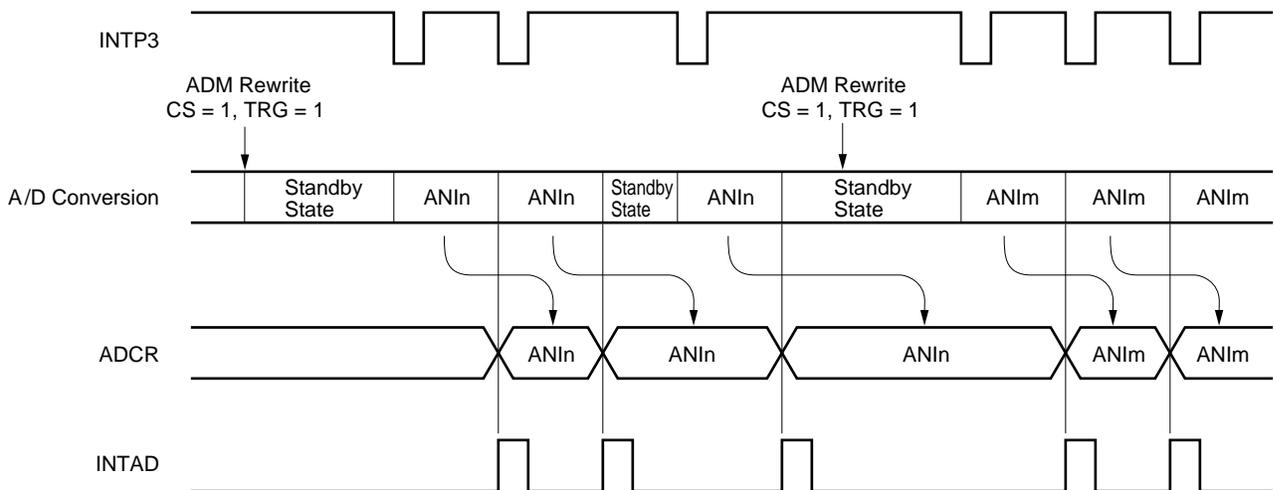
When bit 6 (TRG) and bit 7 (CS) of the A/D converter mode register (ADM) are set to 1, the A/D conversion standby state is set. When the external trigger signal (INTP3) is input, the A/D conversion starts on the voltage applied to the analog input pins specified with bits 1 to 3 (ADM1 to ADM3) of ADM.

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 15-7. A/D Conversion by Hardware Start



Remark $n = 0, 1, \dots, 7$
 $m = 0, 1, \dots, 7$

(2) A/D conversion operation in software start

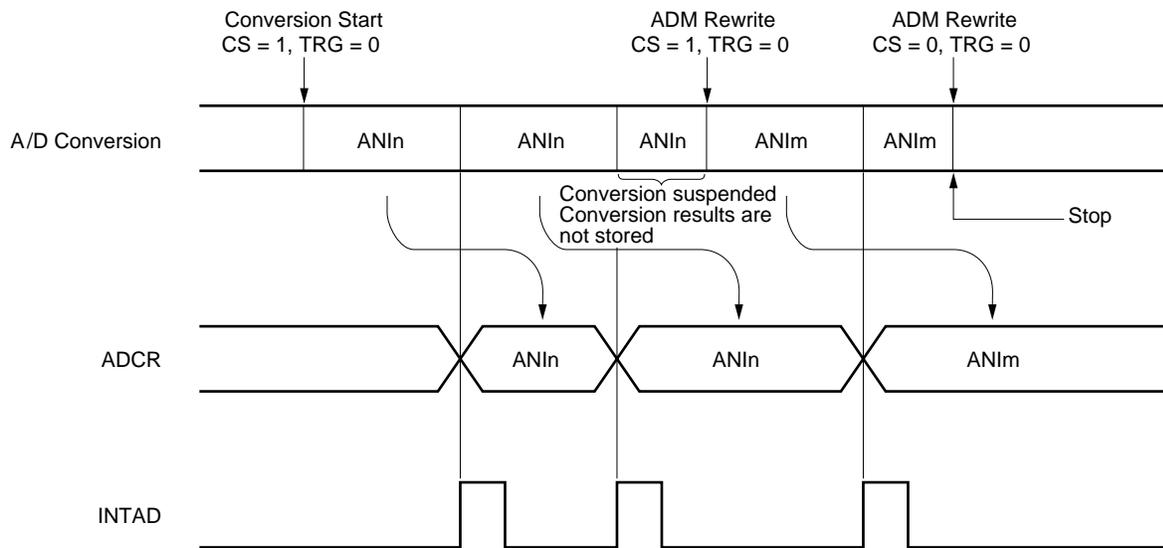
When bit 6 (TRG) and bit 7 (CS) of A/D converter mode register (ADM) are set to 0 and 1, respectively, the A/D conversion starts on the voltage applied to the analog input pins specified with bits 1 to 3 (ADM1 to ADM3) of ADM.

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 15-8. A/D Conversion by Software Start



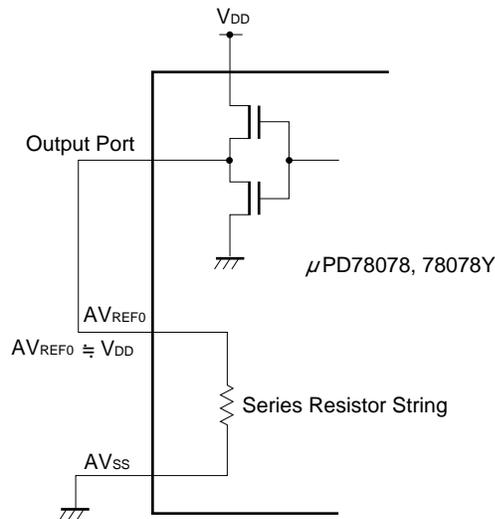
Remark $n = 0, 1, \dots, 7$
 $m = 0, 1, \dots, 7$

15.5 A/D Converter Cautions

(1) Current consumption in standby mode

The A/D converter operates on the main system clock. Therefore, its operation stops in STOP mode or in HALT mode with the subsystem clock. As a current still flows in the AV_{REF0} pin at this time, this current must be cut in order to minimize the overall system power dissipation. In Figure 15-9, the power dissipation can be reduced by outputting a low-level signal to the output port in standby mode. However, there is no precision to the actual AV_{REF0} voltage, and therefore the conversion values themselves lack precision and can only be used for relative comparison.

Figure 15-9. Example of Method of Reducing Current Consumption in Standby Mode



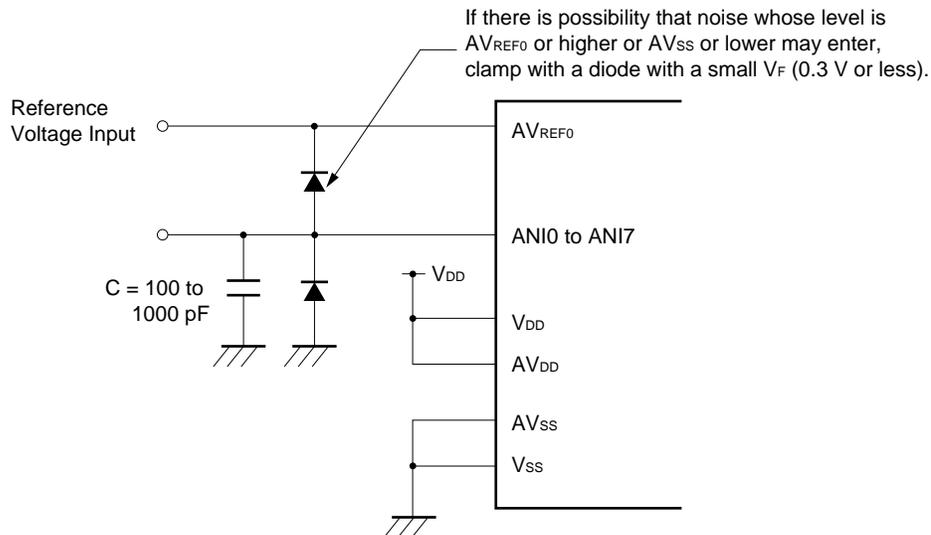
(2) Input range of ANI0 to ANI7

The input voltages of ANI0 to ANI7 should be within the specification range. In particular, if a voltage above AV_{REF0} or below AV_{SS} is input (even if within the absolute maximum rating range), the conversion value for that channel will be indeterminate. The conversion values of the other channels may also be affected.

(3) Noise countermeasures

In order to maintain 8-bit resolution, attention must be paid to noise on pins AV_{REF0} and ANI0 to ANI7. Since the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally as shown in Figure 15-10 in order to reduce noise.

Figure 15-10. Analog Input Pin Disposition

**(4) Pins ANI0/P10 to ANI7/P17**

The analog input pins ANI0 to ANI7 also function as input/output port (PORT1) pins. When A/D conversion is performed with any of pins ANI0 to ANI7 selected, be sure not to execute an input instruction to PORT 1 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.

(5) AV_{REF0} pin input impedance

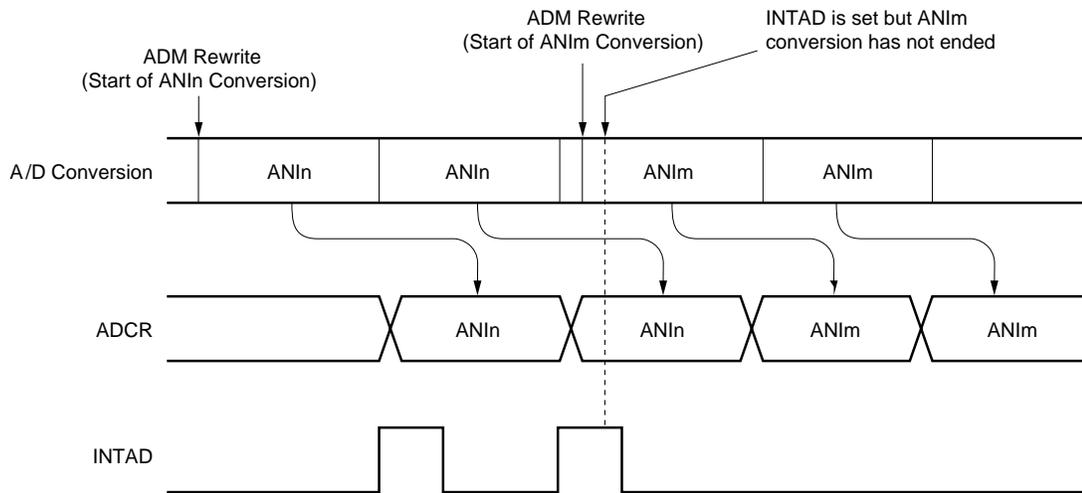
A series resistor string of approximately 10 kΩ is connected between the AV_{REF0} pin and the AV_{SS} pin. Therefore, if the output impedance of the reference voltage source is high, this will result in parallel connection to the series resistor string between the AV_{REF0} pin and the AV_{SS} pin, and there will be a large reference voltage error.

(6) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the A/D converter mode register (ADM) is changed. Caution is therefore required since, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF for the pre-change analog input may be set just before the 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 ADIF before it is resumed.

Figure 15-11. A/D Conversion End Interrupt Request Generation Timing



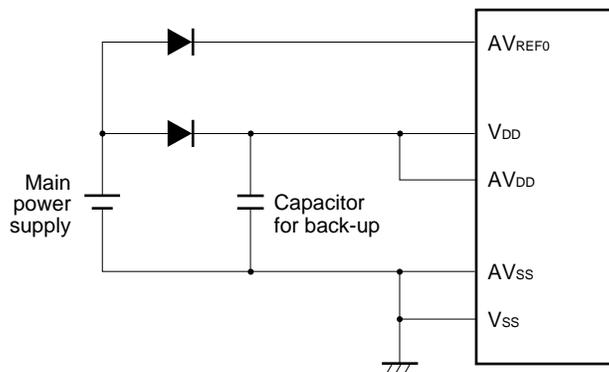
Remark n = 0, 1, ..., 7
m = 0, 1, ..., 7

(7) AV_{DD} pin

The AV_{DD} pin is the analog circuit power supply pin, and supplies power to the input circuits of ANI0/P10 to ANI7/P17.

Therefore, be sure to apply the same voltage as V_{DD} to this pin even when the application circuit is designed so as to switch to a backup battery.

Figure 15-12. Handling of AV_{DD} Pin



[MEMO]

CHAPTER 16 D/A CONVERTER

16.1 D/A Converter Functions

The D/A converter converts a digital input into an analog value. It consists of two 8-bit resolution channels of voltage output type D/A converter.

The conversion method used is the R-2R resistor ladder method.

Start the A/D conversion by setting the DACE0 and DACE1 of the D/A converter mode register (DAM).

There are two types of modes for the D/A converter, as follows.

(1) Normal mode

Outputs an analog voltage signal immediately after the D/A conversion.

(2) Real-time output mode

Outputs an analog voltage signal synchronously with the output trigger after the D/A conversion.

Since a sine wave can be generated in the mode, the use of this mode easily realizes an MSK modem for cordless telephone sets.

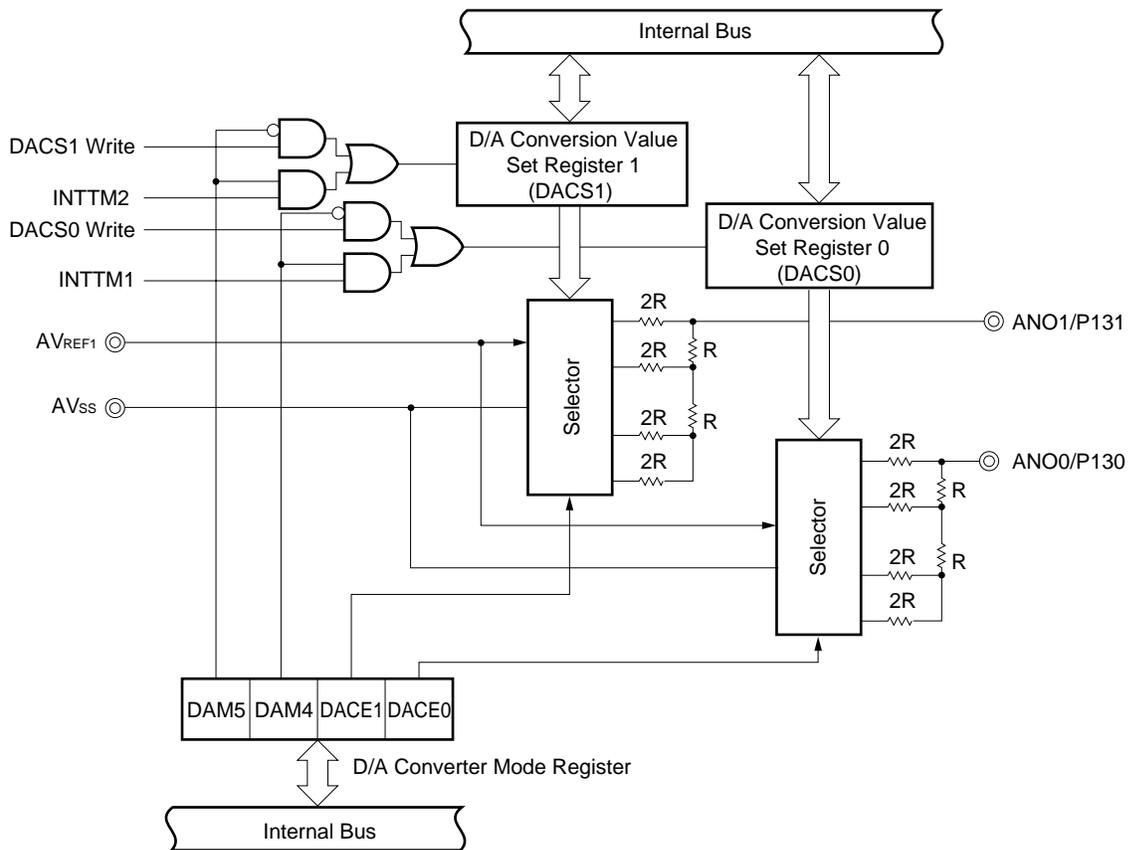
16.2 D/A Converter Configuration

The D/A converter consists of the following hardware.

Table 16-1. D/A Converter Configuration

Item	Configuration
Register	D/A conversion value set register 0 (DACS0) D/A conversion value set register 1 (DACS1)
Control register	D/A converter mode register (DAM)

Figure 16-1. D/A Converter Block Diagram



(1) D/A conversion value set register 0, 1 (DACS0, DACS1)

DACS0 and DACS1 are registers that set the value used to determine analog voltage values output to the ANO0 and ANO1 pins, re-spectively.

DACS0 and DACS1 are set with 8-bit memory manipulation instructions.

$\overline{\text{RESET}}$ input sets these registers to 00H.

Analog voltage output to the ANO0 and ANO1 pins is determined by the following expression.

$$\text{ANOn output voltage} = AV_{\text{REF1}} \times \frac{\text{DACS}_n}{256}$$

where, $n = 0, 1$

- Cautions**
1. In the real-time output mode, when data that are set in DACS0 and DACS1 are read before an output trigger is generated, the previous data are read rather than the set data.
 2. In the real-time output mode, data should be set to DACS0 and DACS1 after an output trigger and before the next output trigger.

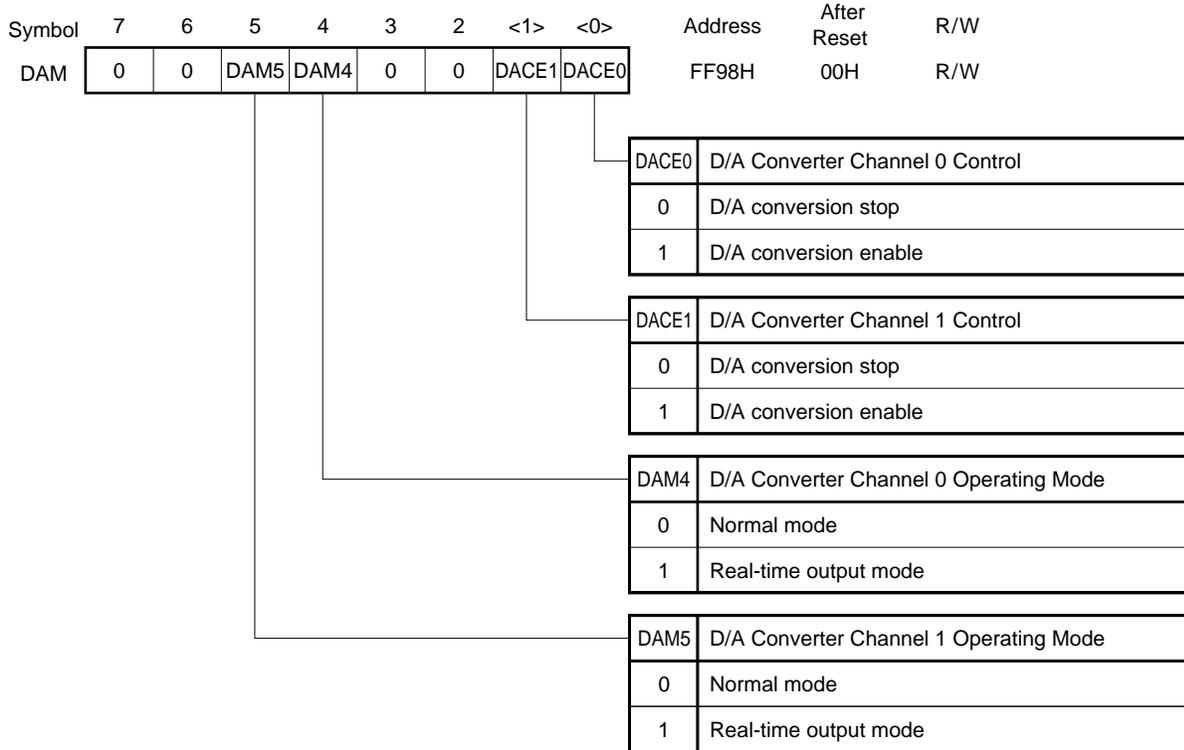
16.3 D/A Converter Control Registers

The D/A converter mode register (DAM) controls the D/A converter. This register sets D/A converter operation enable/stop.

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

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

Figure 16-2. D/A Converter Mode Register Format



- Cautions**
1. When using the D/A converter, a dual-function port pin should be set to the input mode, and a pull-up resistor should be disconnected.
 2. Always set bits 2, 3, 6, and 7 to 0.
 3. When D/A conversion is stopped, the output state is high-impedance.
 4. The output triggers are INTTM1 and INTTM2 for channel 0 and channel 1, respectively, in the real-time output mode.

16.4 D/A Converter Operations

- (1) Select the channel 0 operating mode and channel 1 operating mode with DAM4 and DAM5, respectively, of the D/A converter mode register (DAM).
- (2) Set the data corresponding to the analog voltages output to the ANO0/P130 and ANO1/P131 pins to the D/A conversion value setting registers 0 and 1 (DACS0 and DACS1), respectively.
- (3) Start the channel 0 and channel 1 D/A conversion operations by setting DACE0 and DACE1, respectively, of the DAM.
- (4) In the normal mode, the analog voltage signals are output to the ANO0/P130 and ANO1/P131 pins immediately after the D/A conversion. In the real-time output mode, the analog voltage signals are output synchronously with the output triggers.
- (5) In the normal mode, the analog voltage signals to be output are held until new data are set in DACS0 and DACS1. In the realtime output mode, new data are set in DACS0 and DACS1 and then they are held until the next trigger is generated.

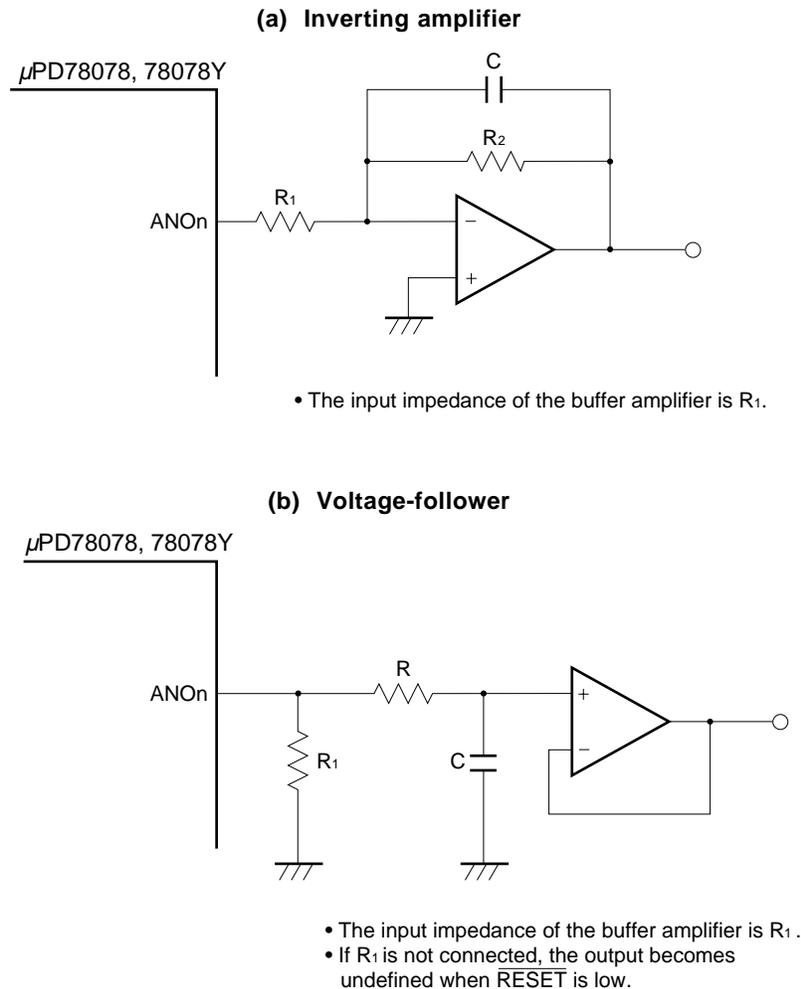
Caution Set DACE0 and DACE1 after setting data in DACS0 and DACS1.

16.5 D/A Converter Cautions

(1) Output impedance of D/A converter

Because the output impedance of the D/A converter is high, use of current flowing from the ANOn pins ($n = 0, 1$) is prohibited. If the input impedance of the load for the converter is low, insert a buffer amplifier between the load and the ANOn pins. In addition, wiring from the ANOn pins to the buffer amplifier or the load should be as short as possible (because of high output impedance). If the wiring may be long, design the ground pattern so as to be close to those lines or use some other expedient to achieve shorter wiring.

Figure 16-3. Use Example of Buffer Amplifier



(2) Output voltage of D/A converter

Because the output voltage of the converter changes in steps, use the D/A converter output signals in general by connecting a low-pass filter.

(3) AV_{REF1} pin

When only either one of the D/A converter channels is used with $AV_{REF1} < V_{DD}$, the other pins that are not used as analog outputs must be set as follows:

- Set PM13x bit of the port mode register 13 (PM13) to 1 (input mode) and connect the pin to V_{SS} .
- Set PM13x bit of the port mode register 13 (PM13) to 0 (output mode) and the output latch to 0, to output low level from the pin.

CHAPTER 17 SERIAL INTERFACE CHANNEL 0 (μ PD78078 SUBSERIES)

The μ PD78078 Subseries incorporates three channels of serial interfaces. Differences between channels 0, 1, and 2 are as follows (Refer to **CHAPTER 19 SERIAL INTERFACE CHANNEL 1** for details of the serial interface channel 1. Refer to **CHAPTER 20 SERIAL INTERFACE CHANNEL 2** for details of the serial interface channel 2).

Table 17-1. Differences between Channels 0, 1, and 2

Serial Transfer Mode		Channel 0	Channel 1	Channel 2
3-wire serial I/O	Clock selection	$f_{xx}/2$, $f_{xx}/2^2$, $f_{xx}/2^3$, $f_{xx}/2^4$, $f_{xx}/2^5$, $f_{xx}/2^6$, $f_{xx}/2^7$, $f_{xx}/2^8$, external clock, TO2 output	$f_{xx}/2$, $f_{xx}/2^2$, $f_{xx}/2^3$, $f_{xx}/2^4$, $f_{xx}/2^5$, $f_{xx}/2^6$, $f_{xx}/2^7$, $f_{xx}/2^8$, external clock, TO2 output	External clock, baud rate generator output
	Transfer method	MSB/LSB switchable as the start bit	MSB/LSB switchable as the start bit Automatic transmit/ receive function	MSB/LSB switchable as the start bit
	Transfer end flag	Serial transfer end interrupt request flag (CSIF0)	Serial transfer end interrupt request flag (CSIF1)	Serial transfer end interrupt request flag (SRIF)
SBI (serial bus interface)	Use possible	None	None	None
2-wire serial I/O				Use possible
UART (Asynchronous serial interface)	None			Use possible

17.1 Serial Interface Channel 0 Functions

Serial interface channel 0 employs the following four modes.

- Operation stop mode
- 3-wire serial I/O mode
- SBI (serial bus interface) mode
- 2-wire serial I/O mode

★ **Caution** Do not change the operation mode (3-wire serial I/O/2-wire serial I/O/SBI) while operation of the serial interface channel 0 is enabled. Stop the serial operation before changing the operation mode.

(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 ($\overline{\text{SCK0}}$), serial output (SO0) and serial input (SI0). This 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 ($\overline{\text{SCK0}}$) and serial data bus (SB0 or SB1).

SBI mode complies with the NEC serial bus format. In the SBI mode, transfer data is transmitted/received as one of the three data types: "address", "command", or "data".

- Address : Data to select the target device for serial communication
- Command : Data to give an instruction to the target device
- Data : Data actually transmitted

The actual transmission is performed in the following procedure: The master device outputs an "address" on the serial bus and selects a slave device with which the master device is to perform communication from among several devices. The master device and the slave device mutually transmit and receive "commands" and "data" to achieve serial transfer. The receiving side can automatically identify the received data as an "address", "command", 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.

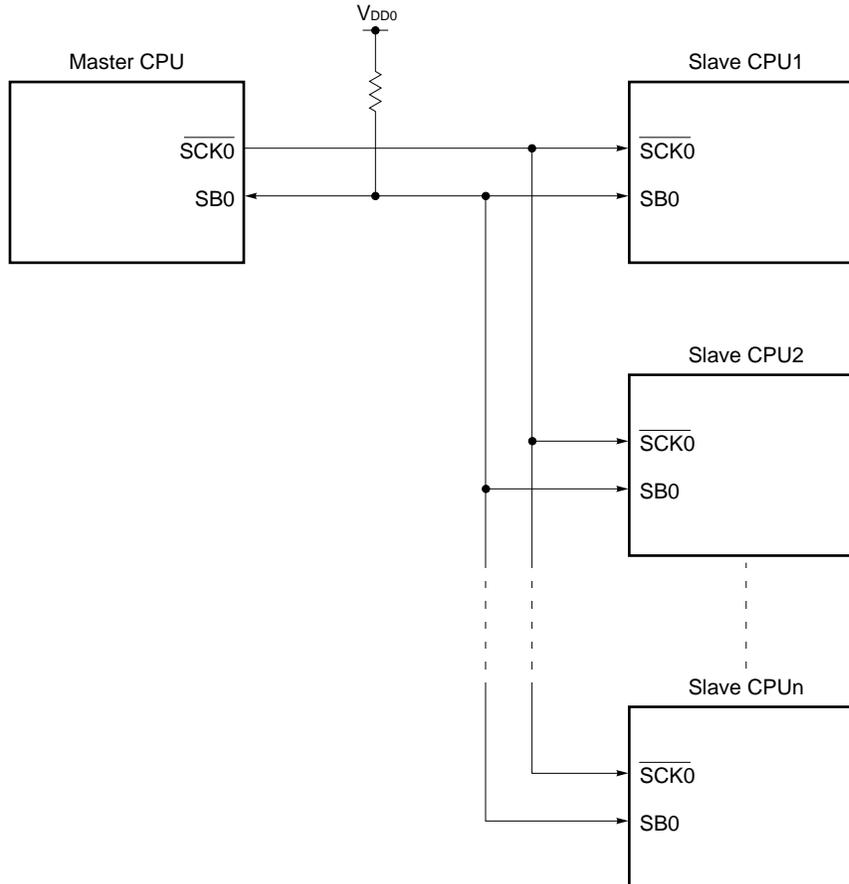
In this mode, the wake-up function for handshake and the output function of acknowledge and busy signals can also be used.

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

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

This mode enables to cope with any one of the possible data transfer formats by controlling the $\overline{\text{SCK0}}$ 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 port pins.

Figure 17-1. Serial Bus Interface (SBI) System Configuration Example



17.2 Serial Interface Channel 0 Configuration

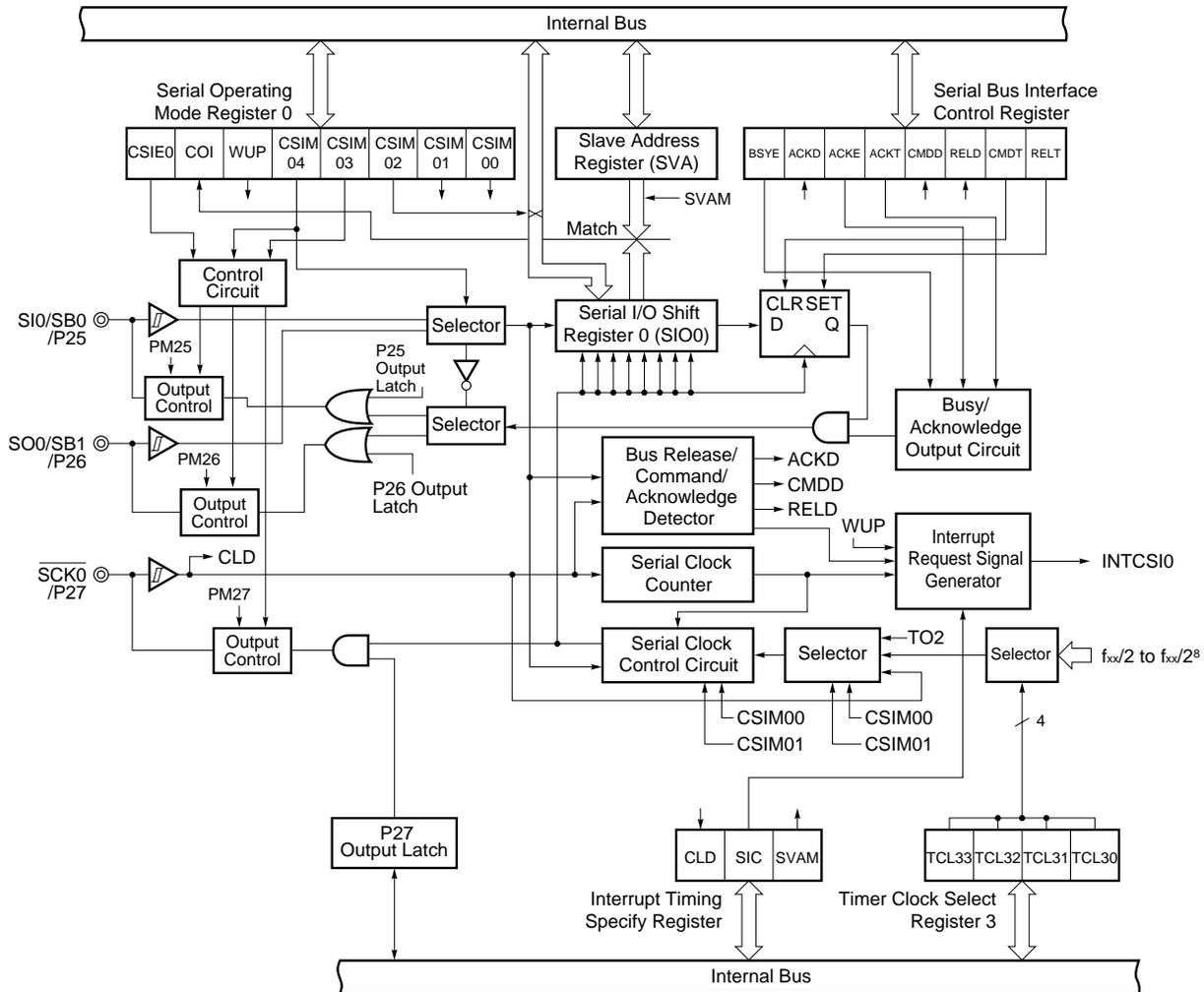
Serial interface channel 0 consists of the following hardware.

Table 17-2. 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 specify register (SINT) Port mode register 2 (PM2) ^{Note}

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

Figure 17-2. Serial Interface Channel 0 Block Diagram



Remark Output control selects either CMOS output or 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 or 2-wire serial I/O 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 SBI mode, the busy state can be cleared by writing data to SIO0. In this case, bit 7 (BSYE) of the serial bus interface control register (SBIC) is not cleared to 0.

$\overline{\text{RESET}}$ input makes SIO0 undefined.

(2) Slave address register (SVA)

This is an 8-bit register to set the slave address value for connection of a slave device to the serial bus. SVA is set with an 8-bit memory manipulation instruction. This register is not used in the 3-wire serial I/O mode.

The master device outputs a slave address for selection of a particular slave device to the connected slave device. These two data (the slave address output from the master device and the SVA value) are compared with an address comparator. If they match, the slave device has been selected. In that case, bit 6 (COI) of serial operating mode register 0 (CSIM0) becomes 1.

Address comparison can also be executed on the data of LSB-masked high-order 7 bits by setting (1) bit 4 (SVAM) of the interrupt timing specify register (SINT).

If no matching is detected in address reception, bit 2 (RELD) of the serial bus interface control register (SBIC) is cleared to 0. In the SBI mode, when bit 5 (WUP) of the CSIM0 is set (1), the wake-up function can be used. In this case, the interrupt request signal (INTCSI0) is generated only if the slave address output from the master matches the SVA value. This interrupt request enables to recognize the generation of the communication request from the master device. If the bit 5 (SIC) of the interrupt timing specify register (SINT) is set (1), the wake-up function does not operate even when WUP is set (1) (the interrupt request signal is generated when a bus release is detected). When using the wake-up function, clear SIC to 0.

Further, when SVA transmits data as master or slave device in the SBI or 2-wire serial I/O mode, errors are detected if any.

$\overline{\text{RESET}}$ input makes SVA undefined.

(3) SO0 latch

This latch holds SI0/SB0/P25 and SO0/SB1/P26 pin levels. It can be directly controlled by software. In the SBI mode, this latch is set upon termination of the 8th serial clock.

(4) Serial clock counter

This counter counts the serial clocks to be output and input during transmission/reception and to check whether 8-bit data has been transmitted/received.

(5) Serial clock control circuit

This circuit controls serial clock supply to the serial I/O shift register 0 (SIO0). When the internal system clock is used, the circuit also controls clock output to the $\overline{\text{SCK0}}$ /P27 pin.

(6) Interrupt request signal generator

This circuit controls interrupt request signal generation. It generates the interrupt request signal in the following cases.

- In the 3-wire serial I/O mode and 2-wire serial I/O mode
This circuit generates an interrupt request signal every eight serial clocks.
- In the SBI mode
When WUP^{Note} is 0 Generates an interrupt request signal every eight serial clocks.
When WUP^{Note} is 1 Generates an interrupt request signal when the serial I/O shift register 0 (SIO0) value matches the slave address register (SVA) value after address reception.

Note WUP is wake-up function specify bit. It is bit 5 of serial operating mode register 0 (CSIM0). When using the wake-up function (WUP = 1), clear bit 5 (SIC) of the interrupt timing specify register (SINT) to 0.

(7) Busy/acknowledge output circuit and bus release/command/acknowledge detector

These two circuits output and detect various control signals in the SBI mode.

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

17.3 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.

$\overline{\text{RESET}}$ input sets TCL3 to 88H.

Figure 17-3. Timer Clock Select Register 3 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33				Serial Interface Channel 0 Serial Clock Selection		
TCL33	TCL32	TCL31	TCL30		MCS = 1	MCS = 0
0	1	1	0	$f_{xx}/2$	Setting prohibited	$f_x/2^2$ (1.25 MHz)
0	1	1	1	$f_{xx}/2^2$	$f_x/2^2$ (1.25 MHz)	$f_x/2^3$ (625 kHz)
1	0	0	0	$f_{xx}/2^3$	$f_x/2^3$ (625 kHz)	$f_x/2^4$ (313 kHz)
1	0	0	1	$f_{xx}/2^4$	$f_x/2^4$ (313 kHz)	$f_x/2^5$ (156 kHz)
1	0	1	0	$f_{xx}/2^5$	$f_x/2^5$ (156 kHz)	$f_x/2^6$ (78.1 kHz)
1	0	1	1	$f_{xx}/2^6$	$f_x/2^6$ (78.1 kHz)	$f_x/2^7$ (39.1 kHz)
1	1	0	0	$f_{xx}/2^7$	$f_x/2^7$ (39.1 kHz)	$f_x/2^8$ (19.5 kHz)
1	1	0	1	$f_{xx}/2^8$	$f_x/2^8$ (19.5 kHz)	$f_x/2^9$ (9.8 kHz)
Other than above				Setting prohibited		

TCL37				Serial Interface Channel 1 Serial Clock Selection		
TCL37	TCL36	TCL35	TCL34		MCS = 1	MCS = 0
0	1	1	0	$f_{xx}/2$	Setting prohibited	$f_x/2^2$ (1.25 MHz)
0	1	1	1	$f_{xx}/2^2$	$f_x/2^2$ (1.25 MHz)	$f_x/2^3$ (625 kHz)
1	0	0	0	$f_{xx}/2^3$	$f_x/2^3$ (625 kHz)	$f_x/2^4$ (313 kHz)
1	0	0	1	$f_{xx}/2^4$	$f_x/2^4$ (313 kHz)	$f_x/2^5$ (156 kHz)
1	0	1	0	$f_{xx}/2^5$	$f_x/2^5$ (156 kHz)	$f_x/2^6$ (78.1 kHz)
1	0	1	1	$f_{xx}/2^6$	$f_x/2^6$ (78.1 kHz)	$f_x/2^7$ (39.1 kHz)
1	1	0	0	$f_{xx}/2^7$	$f_x/2^7$ (39.1 kHz)	$f_x/2^8$ (19.5 kHz)
1	1	0	1	$f_{xx}/2^8$	$f_x/2^8$ (19.5 kHz)	$f_x/2^9$ (9.8 kHz)
Other than above				Setting prohibited		

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

- Remarks**
1. f_{xx} : Main system clock frequency (f_x or $f_x/2$)
 2. f_x : Main system clock oscillation frequency
 3. MCS : Bit 0 of oscillation mode selection register (OSMS)
 4. Figures in parentheses apply to operation with $f_x = 5.0$ 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.

$\overline{\text{RESET}}$ input sets CSIM0 to 00H.

- ★ **Caution** Do not change the operation mode (3-wire serial I/O/2-wire serial I/O/SBI) while operation of the serial interface channel 0 is enabled. Stop the serial operation before changing the operation mode.

Figure 17-4. Serial Operating Mode Register 0 Format

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	After Reset	R/W			
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W ^{Note 1}			
R/W	CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection											
	0	x	Input Clock to $\overline{\text{SCK0}}$ pin from off-chip											
	1	0	8-bit timer register 2 (TM2) output											
	1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)											
R/W	CSIM04	CSIM03	CSIM02	PM25	P25	PM26	P26	PM27	P27	Operation Mode	Start Bit	SI0/SB0/P25 Pin Function	SO0/SB1/P26 Pin Function	$\overline{\text{SCK0}}$ /P27 Pin Function
	0	x	0	^{Note 2} 1	^{Note 2} x	0	0	0	1	3-wire serial I/O mode	MSB	SI0 ^{Note 2} (Input)	SO0 (CMOS output)	$\overline{\text{SCK0}}$ (CMOS input/output)
			1						LSB					
	1	0	0	^{Note 3} x	^{Note 3} x	0	0	0	1	SBI mode	MSB	P25 (CMOS input/output)	SB1 (N-ch open-drain input/output)	$\overline{\text{SCK0}}$ (CMOS input/output)
			1	0	0	^{Note 3} x	^{Note 3} x	0	1			SB0 (N-ch open-drain input/output)	P26 (CMOS input/output)	
	1	1	0	^{Note 3} x	^{Note 3} x	0	0	0	1	2-wire serial I/O mode	MSB	P25 (CMOS input/output)	SB1 (N-ch open-drain input/output)	$\overline{\text{SCK0}}$ (N-ch open-drain input/output)
			1	0	0	^{Note 3} x	^{Note 3} x	0	1			SB0 (N-ch open-drain input/output)	P26 (CMOS input/output)	
R/W	WUP	Wake-up Function Control ^{Note 4}												
	0	Interrupt request signal generation with each serial transfer in any mode												
	1	Interrupt request signal generation when the address received after bus release (when CMDD = RELD = 1) matches the slave address register data in SBI mode												
R	COI	Slave Address Comparison Result Flag ^{Note 5}												
	0	Slave address register not equal to serial I/O shift register 0 data												
	1	Slave address register equal to serial I/O shift register 0 data												
R/W	CSIE0	Serial Interface Channel 0 Operation Control												
	0	Operation stopped												
	1	Operation enable												

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

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

3. Can be used freely as port function.

4. When using the wake-up function (WUP = 1), set bit 5 (SIC) of the interrupt timing specify register (SINT) to 0.

5. When CSIE0 = 0, COI becomes 0.

Remark x : Don't care

PMxx : Port mode register

Pxx : Port output latch

(3) Serial bus interface control register (SBIC)

This register sets serial bus interface operation and displays statuses.

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

$\overline{\text{RESET}}$ input sets SBIC to 00H.

Figure 17-5. Serial Bus Interface Control Register Format (1/2)

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After Reset	R/W																										
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W ^{Note}																										
R/W	<table border="1"> <tr> <td>RELT</td> <td>Used for bus release signal output. When RELT = 1, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.</td> </tr> </table>											RELT	Used for bus release signal output. When RELT = 1, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.																								
RELT	Used for bus release signal output. When RELT = 1, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.																																				
R/W	<table border="1"> <tr> <td>CMDT</td> <td>Used for command signal output. When CMDT = 1, SO0 latch is cleared to 0. After SO0 latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.</td> </tr> </table>											CMDT	Used for command signal output. When CMDT = 1, SO0 latch is cleared to 0. After SO0 latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.																								
CMDT	Used for command signal output. When CMDT = 1, SO0 latch is cleared to 0. After SO0 latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.																																				
R	<table border="1"> <tr> <td>RELD</td> <td>Bus Release Detection</td> </tr> <tr> <td colspan="5">Clear Conditions (RELD = 0)</td> <td colspan="7">Set Conditions (RELD = 1)</td> </tr> <tr> <td colspan="5"> <ul style="list-style-type: none"> • When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied </td> <td colspan="7"> <ul style="list-style-type: none"> • When bus release signal (REL) is detected </td> </tr> </table>											RELD	Bus Release Detection	Clear Conditions (RELD = 0)					Set Conditions (RELD = 1)							<ul style="list-style-type: none"> • When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied 					<ul style="list-style-type: none"> • When bus release signal (REL) is detected 						
RELD	Bus Release Detection																																				
Clear Conditions (RELD = 0)					Set Conditions (RELD = 1)																																
<ul style="list-style-type: none"> • When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied 					<ul style="list-style-type: none"> • When bus release signal (REL) is detected 																																
R	<table border="1"> <tr> <td>CMDD</td> <td>Command Detection</td> </tr> <tr> <td colspan="5">Clear Conditions (CMDD = 0)</td> <td colspan="7">Set Conditions (CMDD = 1)</td> </tr> <tr> <td colspan="5"> <ul style="list-style-type: none"> • When transfer start instruction is executed • When bus release signal (REL) is detected • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied </td> <td colspan="7"> <ul style="list-style-type: none"> • When command signal (CMD) is detected </td> </tr> </table>											CMDD	Command Detection	Clear Conditions (CMDD = 0)					Set Conditions (CMDD = 1)							<ul style="list-style-type: none"> • When transfer start instruction is executed • When bus release signal (REL) is detected • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied 					<ul style="list-style-type: none"> • When command signal (CMD) is detected 						
CMDD	Command Detection																																				
Clear Conditions (CMDD = 0)					Set Conditions (CMDD = 1)																																
<ul style="list-style-type: none"> • When transfer start instruction is executed • When bus release signal (REL) is detected • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied 					<ul style="list-style-type: none"> • When command signal (CMD) is detected 																																
R/W	<table border="1"> <tr> <td>ACKT</td> <td>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 ACE = 0. Also cleared to 0 upon start of serial interface transfer or when CSIE0 = 0.</td> </tr> </table>											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 ACE = 0. Also cleared to 0 upon start of serial interface transfer or when CSIE0 = 0.																								
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 ACE = 0. Also cleared to 0 upon start of serial interface transfer or when CSIE0 = 0.																																				

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

Remarks 1. Bits 0, 1, and 4 (RELD, CMDT, and ACKT) are 0 when read after data setting.
2. CSIE0 : Bit 7 of serial operating mode register 0 (CSIM0)

Figure 17-5. Serial Bus Interface Control Register Format (2/2)

R/W	ACKE	Acknowledge Signal Output Control	
	0	Acknowledge signal automatic output disable (output with ACKT enable)	
	1	Before completion of transfer	Acknowledge signal is output in synchronization with the 9th clock falling edge of SCK0 (automatically output when ACKE = 1).
After completion of transfer		Acknowledge signal is output in synchronization with the falling edge of 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)
	<ul style="list-style-type: none"> Falling edge of the SCK0 immediately after the busy mode is released while executing the transfer start instruction When CSIE0 = 0 When RESET input is applied 		<ul style="list-style-type: none"> When acknowledge signal (ACK) is detected at the rising edge of 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 SCK0 clock just after execution of the instruction to be cleared to 0.	
	1	Outputs busy signal at the falling edge of SCK0 clock following the acknowledge signal.	

★ **Note** The busy mode can be cancelled with start of serial interface transfer. However, the BSYE flag is not cleared to 0.

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

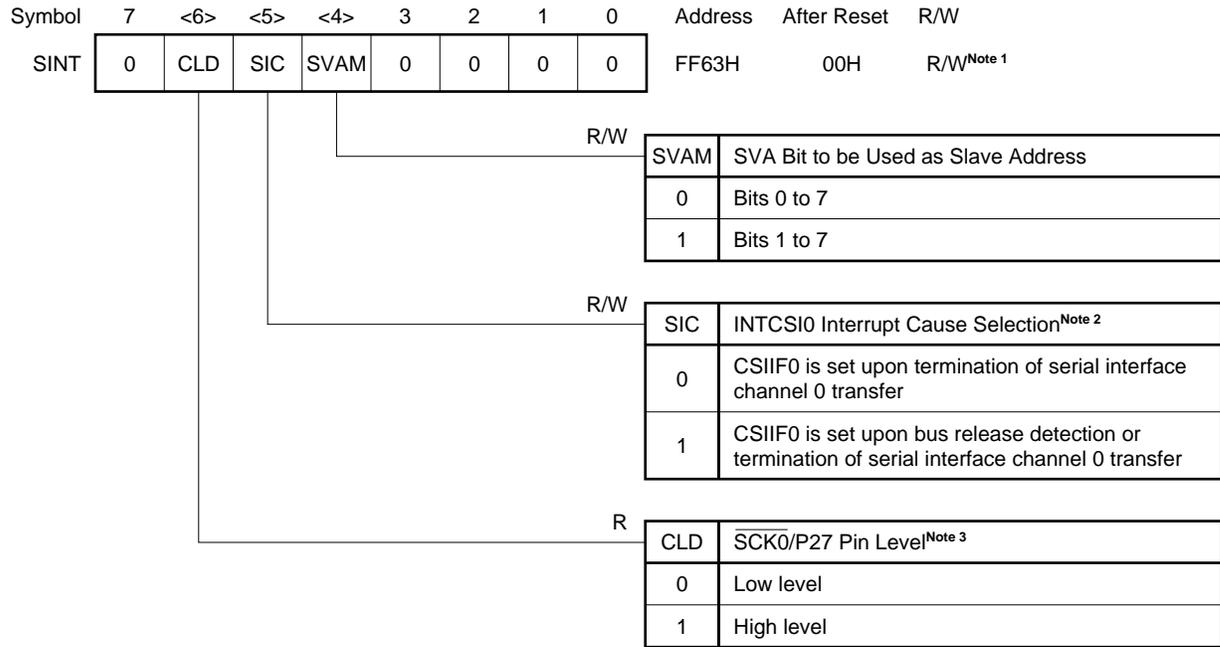
(4) Interrupt timing specify register (SINT)

This register sets the bus release interrupt and address mask functions and displays the $\overline{\text{SCK0/P27}}$ pin level status.

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

$\overline{\text{RESET}}$ input sets SINT to 00H.

Figure 17-6. Interrupt Timing Specify Register Format



- 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 Set bits 0 to 3 to 0.

Remark

- SVA : Slave address register
- CSIF0 : Interrupt request flag corresponding to INTCSI0
- CSIE0 : Bit 7 of serial operating mode register 0 (CSIM0)

17.4 Serial Interface Channel 0 Operations

The following four operating modes are available to the serial interface channel 0.

- Operation stop mode
- 3-wire serial I/O mode
- SBI mode
- 2-wire serial I/O mode

17.4.1 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, P26/SO0/SB1 and P27/ $\overline{\text{SCK0}}$ pins can be used as ordinary input/output ports.

(1) Register setting

The operation stop mode is set with the serial operating mode register 0 (CSIM0).

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM0 to 00H.

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W

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

17.4.2 3-wire serial I/O mode operation

The 3-wire serial I/O mode is valid for connection of peripheral I/O units and display controllers which incorporate a conventional synchronous clocked serial interface as is the case with the 75X/XL, 78K, and 17K series.

Communication is carried out with three lines of serial clock ($\overline{\text{SCK0}}$), serial output (SO0), and serial input (SI0).

(1) Register setting

The 3-wire serial I/O mode is set with the serial operating mode register 0 (CSIM0) and serial bus interface control register (SBIC).

(a) Serial operating mode register 0 (CSIM0)

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM0 to 00H.

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W ^{Note 1}

R/W	CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection								
	0	x	Input Clock to $\overline{\text{SCK0}}$ pin from off-chip								
	1	0	8-bit timer register 2 (TM2) output								
	1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)								

R/W	CSIM04	CSIM03	CSIM02	PM25	P25	PM26	P26	PM27	P27	Operation Mode	Start Bit	SI0/SB0/P25 Pin Function	SO0/SB1/P26 Pin Function	$\overline{\text{SCK0}}$ /P27 Pin Function		
	0	x	0	Note 2	Note 2	1	x	0	0	0	1	3-wire serial I/O mode	MSB	SI0 ^{Note 2} (Input)	SO0 (CMOS output)	$\overline{\text{SCK0}}$ (CMOS input/output)
			1									LSB				
	1	0	SBI mode (See 17.4.3 SBI mode operation.)													
	1	1	2-wire serial I/O mode (See 17.4.4 2-wire serial I/O mode operation.)													

R/W	WUP	Wake-up Function Control ^{Note 3}									
	0	Interrupt request signal generation with each serial transfer in any mode									
	1	Interrupt request signal generation when the address received after bus release (when CMDD = RELD = 1) matches the slave address register (SVA) data in SBI mode									

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

Remark

- x : Don't care
- PMxx : Port mode register
- Pxx : Port output latch

(b) Serial bus interface control register (SBIC)

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

$\overline{\text{RESET}}$ input 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, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.
-----	------	---

R/W	CMDT	When CMDT = 1, SO0 latch is cleared to 0. After SO0 latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.
-----	------	---

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

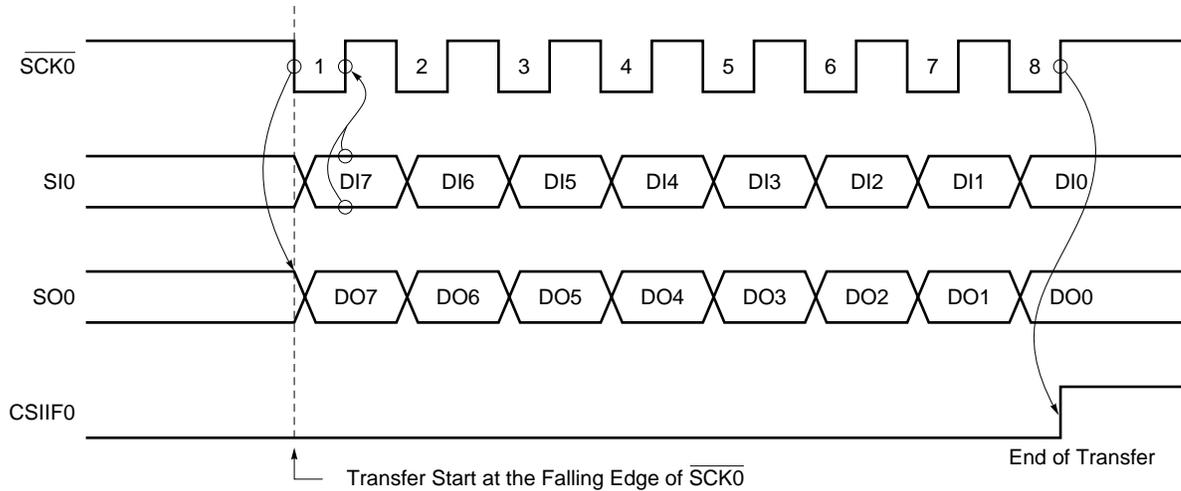
(2) Communication operation

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

Upon termination of 8-bit transfer, SIO0 operation stops automatically and the interrupt request flag (CSIF0) is set.

Figure 17-7. 3-Wire Serial I/O Mode Timings



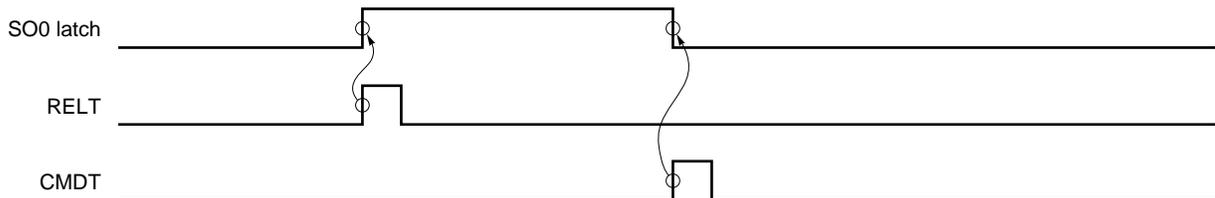
The SO0 pin is a CMOS output pin and outputs current SO0 latch statuses. Thus, the SO0 pin output status can be manipulated by setting bit 0 (RELT) and bit 1 (CMDT) of the serial bus interface control register (SBIC). However, do not carry out this manipulation during serial transfer.

Control the $\overline{\text{SCK0}}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **17.4.5 $\overline{\text{SCK0}}$ /P27 pin output manipulation**).

(3) Other signals

Figure 17-8 shows RELT and CMDT operations.

Figure 17-8. RELT and CMDT Operations

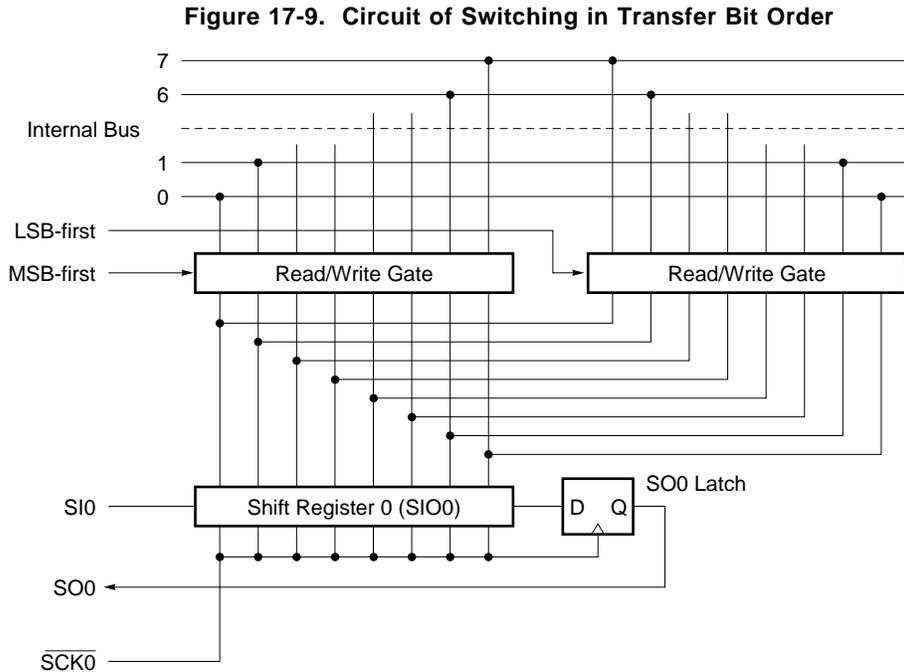


(4) MSB/LSB switching as the start bit

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

Figure 17-9 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).



Start bit switching is realized by switching the bit order for data write to SIO0. The SIO0 shift order remains unchanged.

Thus, switching between the 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.

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (CSIF0) is set.

17.4.3 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 microcontrollers and peripheral ICs, the number of ports to be used and the number of wires on the board can be decreased.

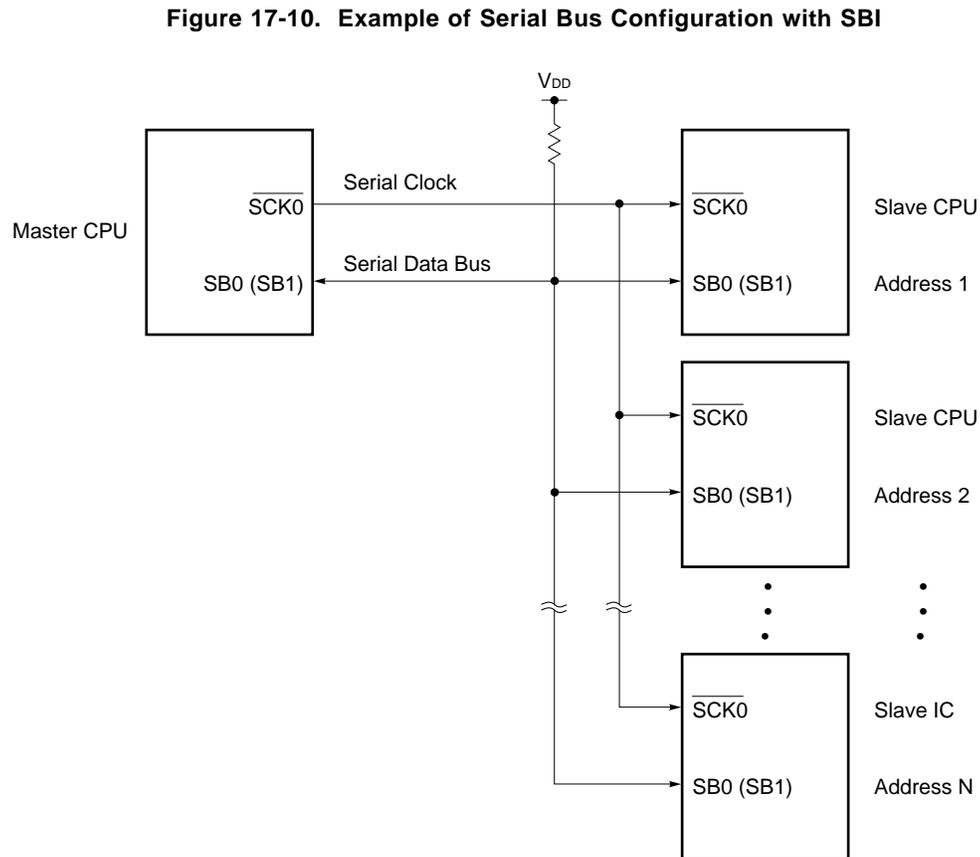
The master device outputs three kinds of data to slave devices on the serial data bus: “addresses” to select a device to be communicated with, “commands” to instruct the selected device, and “data” which is actually required.

The slave device can identify the received data into “address,” “command,” or “data,” by hardware. This function simplifies application programs which control serial interface channel 0.

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

Figure 17-10 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.



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.

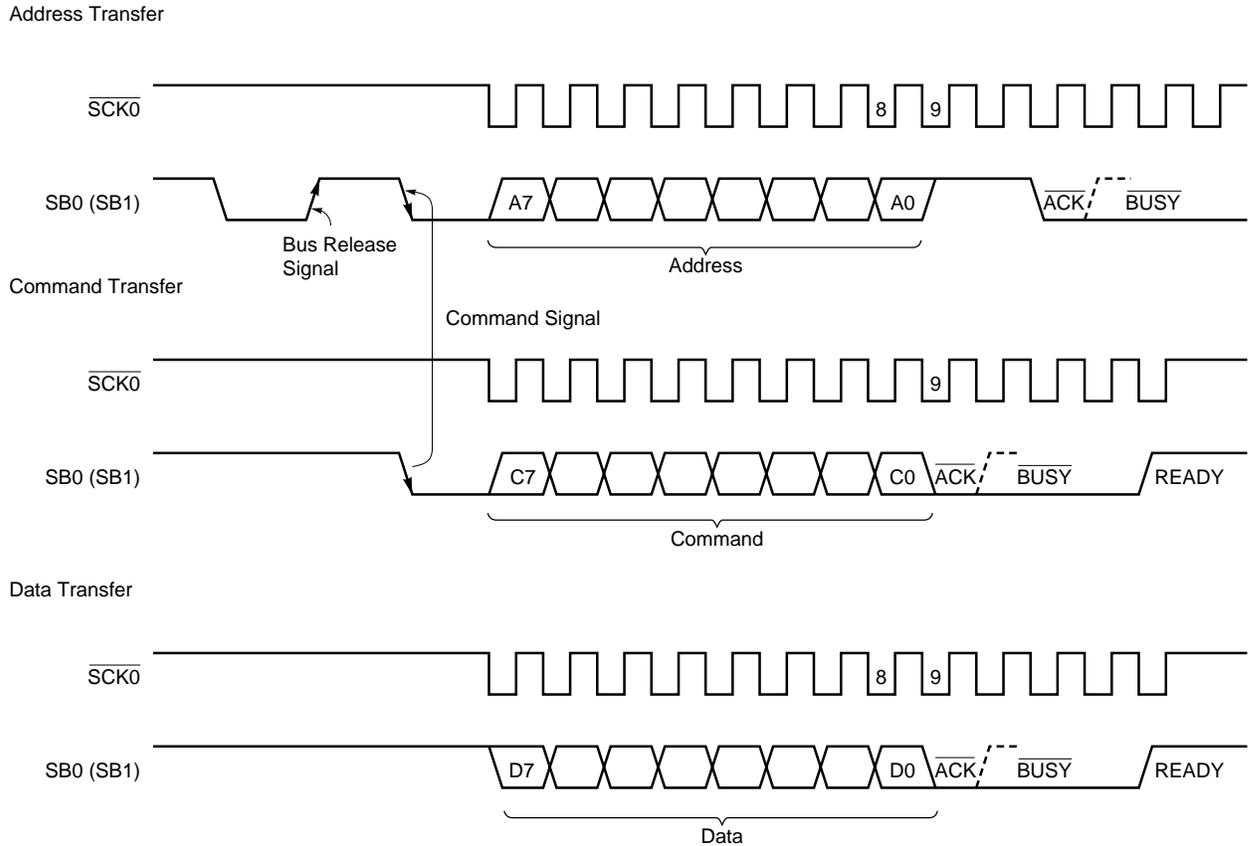
(2) SBI definition

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 17-11 shows the address, command, and data transfer timings.

Figure 17-11. SBI Transfer Timings



Remark The broken lines indicate the **READY** state.

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).

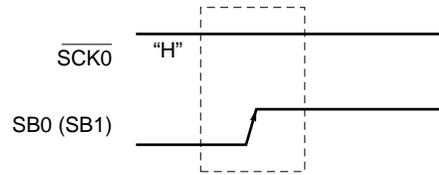
Serial clocks continue to be output by the master device from 8-bit data transfer start to $\overline{\text{BUSY}}$ reset.

(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 $\overline{\text{SCK0}}$ line is at the high level (without serial clock output).

This signal is output by the master device.

Figure 17-12. Bus Release Signal



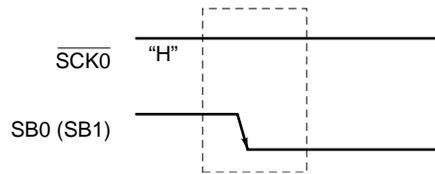
The bus release signal indicates that the master device is going to transmit an address to the slave device. The slave device incorporates hardware to detect the bus release signal.

- ★ **Caution** If the SB0 (SB1) line changes from low level to high level while the $\overline{\text{SCK0}}$ line is in high level, this is recognized as a bus release signal. Therefore, if the changing timing of bus fluctuates because of the substrate capacity, etc., it may be recognized as a bus release signal even while data is being transmitted. Care should be taken for the 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 $\overline{\text{SCK0}}$ line is at the high level (without serial clock output). This signal is output by the master device.

Figure 17-13. Command Signal



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

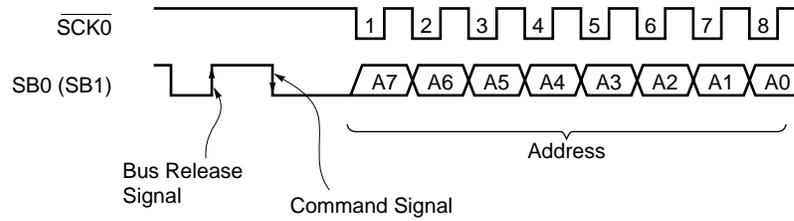
The slave device incorporates hardware to detect the command signal.

- ★ **Caution** If the SB0 (SB1) line changes from high level to low level while the $\overline{\text{SCK0}}$ line is in high level, this is recognized as a command signal. Therefore, if the changing timing of bus fluctuates because of the substrate capacity, etc., it may be recognized as a command signal even while data is being transmitted. Care should be taken for the wiring.

(c) Address

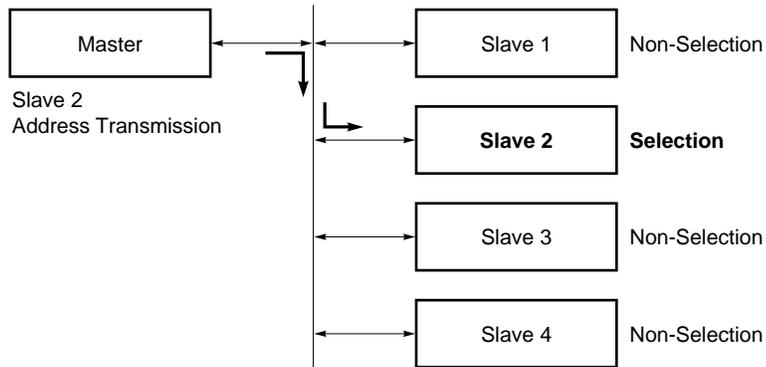
An address is 8-bit data which the master device outputs to the slave device connected to the bus line in order to select a particular slave device.

Figure 17-14. Addresses



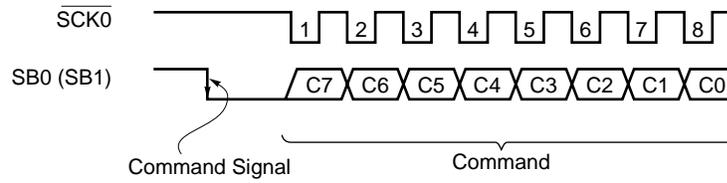
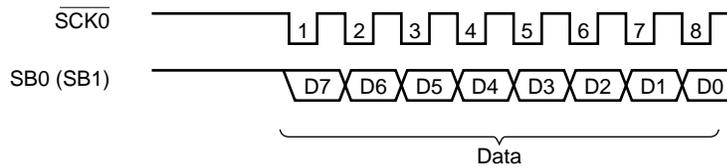
8-bit data following bus release and command signals is defined as an “address”. In the slave device, this condition is detected by hardware and whether or not 8-bit data matches the own specification number (slave address) is checked by hardware. If the 8-bit data matches the slave address, the slave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

Figure 17-15. 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.

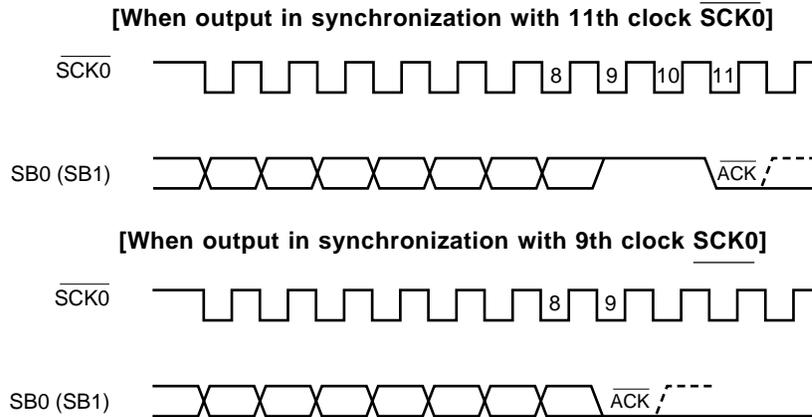
Figure 17-16. Commands**Figure 17-17. Data**

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 arbitrarily according to communications specifications.

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

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

Figure 17-18. Acknowledge Signal



Remark The broken lines indicate the READY state.

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

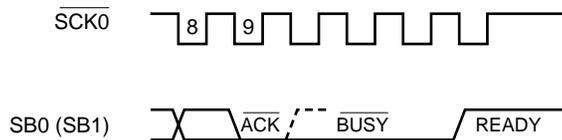
After 8-bit data transmission, the transmitter checks whether the receiver has returned the acknowledge signal. If the acknowledge signal is not returned for the preset period of time after data transmission, it can be judged that data reception has not been carried out correctly.

(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 17-19. $\overline{\text{BUSY}}$ 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, the $\overline{\text{BUSY}}$ signal continues to be output after $\overline{\text{BUSY}}$ clear instruction generation to the falling edge of the next serial clock ($\overline{\text{SCK0}}$). If WUP is set to 1 in this period, $\overline{\text{BUSY}}$ will not be released. Therefore, before setting WUP to 1, clear $\overline{\text{BUSY}}$ and then check that SB0 (SB1) pin has become high level.

(3) Register setting

The SBI mode is set with the serial operating mode register 0 (CSIM0), the serial bus interface control register (SBIC) and the interrupt timing specify register (SINT).

(a) Serial operating mode register 0 (CSIM0)

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM0 to 00H.

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W ^{Note 1}

R/W	CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection
	0	x	Input Clock to $\overline{\text{SCK0}}$ pin from off-chip
	1	0	8-bit timer register 2 (TM2) output
	1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W	CSIM04	CSIM03	CSIM02	PM25	P25	PM26	P26	PM27	P27	Operation Mode	Start Bit	SI0/SB0/P25 Pin Function	SO0/SB1/P26 Pin Function	$\overline{\text{SCK0}}$ /P27 Pin Function
	0	x	3-wire serial I/O mode (See 17.4.2 3-wire serial I/O mode operation.)											
	1	0	0	^{Note 2} x	^{Note 2} x	0	0	0	1	SBI mode	MSB	P25 (CMOS input/output)	SB1 (N-ch open-drain input/output)	$\overline{\text{SCK0}}$ (CMOS input/output)
			1	0	0	^{Note 2} x	^{Note 2} x	0	1			SB0 (N-ch open-drain input/output)	P26 (CMOS input/output)	
	1	1	2-wire serial I/O mode (see 17.4.4 2-wire serial I/O mode operation.)											

R/W	WUP	Wake-up Function Control ^{Note 3}
	0	Interrupt request signal generation with each serial transfer in any mode
	1	Interrupt request signal generation when the address received after bus release (when CMDD = RELD = 1) matches the slave address register (SVA) data in SBI mode

R	COI	Slave Address Comparison Result Flag ^{Note 4}
	0	Slave address register (SVA) not equal to serial I/O shift register 0 (SIO0) data
	1	Slave address register (SVA) equal to serial I/O shift register 0 (SIO0) 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. When using the wake-up function (WUP = 1), set bit 5 (SIC) of the interrupt timing specify register (SINT) to 0.
 4. When CSIE0 = 0, COI becomes 0.

Remark x : Don't care
 PMxx : Port mode register
 Pxx : Port output latch

(b) Serial bus interface control register (SBIC)

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

$\overline{\text{RESET}}$ input 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, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.
-----	------	--

R/W	CMDT	Used for command signal output. When CMDT = 1, SO0 latch is cleared to 0. After SO0 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)
<ul style="list-style-type: none"> • When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception (when WUP = 1) • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied 		<ul style="list-style-type: none"> • When bus release signal (REL) is detected

R	CMDD	Command Detection
Clear Conditions (CMDD = 0)		Set Conditions (CMDD = 1)
<ul style="list-style-type: none"> • When transfer start instruction is executed • When bus release signal (REL) is detected • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied 		<ul style="list-style-type: none"> • 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	<table border="1"> <tr> <td>Before completion of transfer</td> <td>Acknowledge signal is output in synchronization with the 9th clock falling edge of $\overline{\text{SCK0}}$ (automatically output when ACKE = 1).</td> </tr> <tr> <td>After completion of transfer</td> <td>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.</td> </tr> </table>	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.
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. CSIE0 : Bit 7 of serial operating mode register 0 (CSIM0)

R	ACKD	Acknowledge Detection	
	Clear Conditions (ACKD = 0)		Set Conditions (ACKD = 1)
	<ul style="list-style-type: none"> • $\overline{SCK0}$ fall immediately after the busy mode is released during the transfer start instruction execution. • When CSIE0 = 0 • When \overline{RESET} input is applied 		<ul style="list-style-type: none"> • When acknowledge signal (\overline{ACK}) is detected at the rising edge of $\overline{SCK0}$ clock after completion of transfer

R/W	Note	Synchronizing Busy Signal Output Control	
	BSYE		
	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) (sets ready state).	
1	Outputs busy signal at the falling edge of $\overline{SCK0}$ clock following the acknowledge signal.		

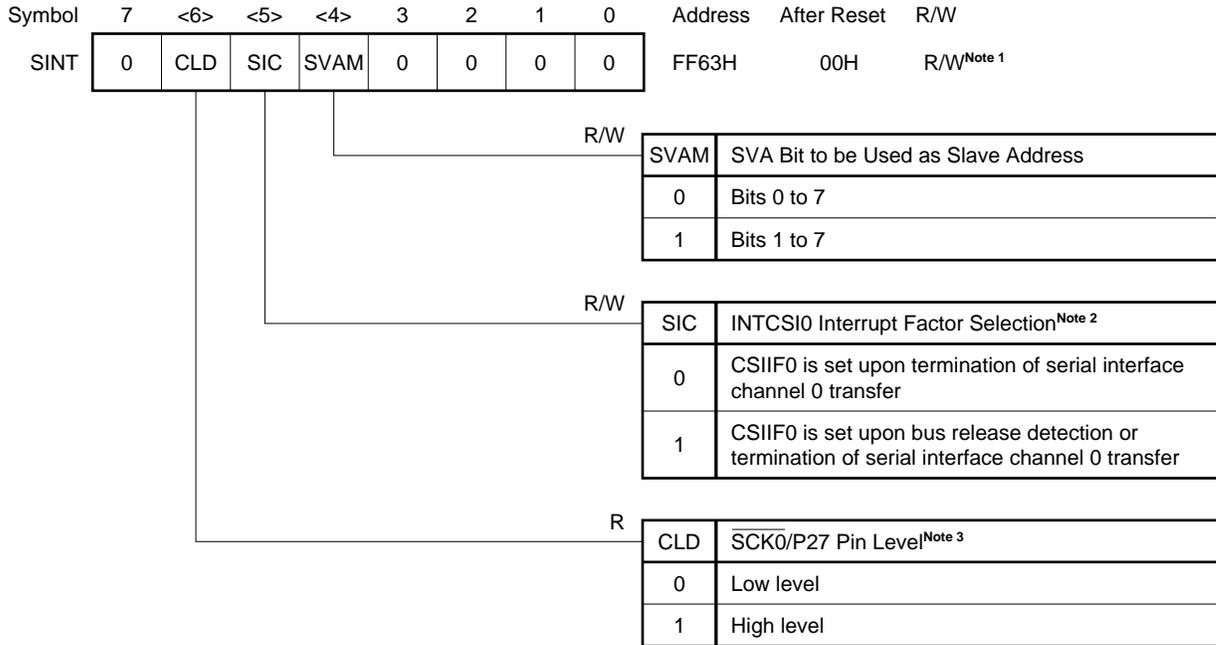
★ **Note** Busy mode can be cleared by start of serial interface transfer. However, BSYE flag is not cleared to 0.

Remark CSIE0 : Bit 7 of 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.



- 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 Set bits 0 to 3 to 0.

Remark SVA : Slave address register
 CSIF0 : Interrupt request flag corresponding to INTCSI0
 CSIE0 : Bit 7 of serial operating mode register 0 (CSIM0)

(4) Various signals

Figures 17-20 to 17-25 show various signals and flag operations in the serial bus interface control register (SBIC). Table 17-3 lists various signals in SBI.

Figure 17-20. RELT, CMDT, RELD, and CMDD Operations (Master)

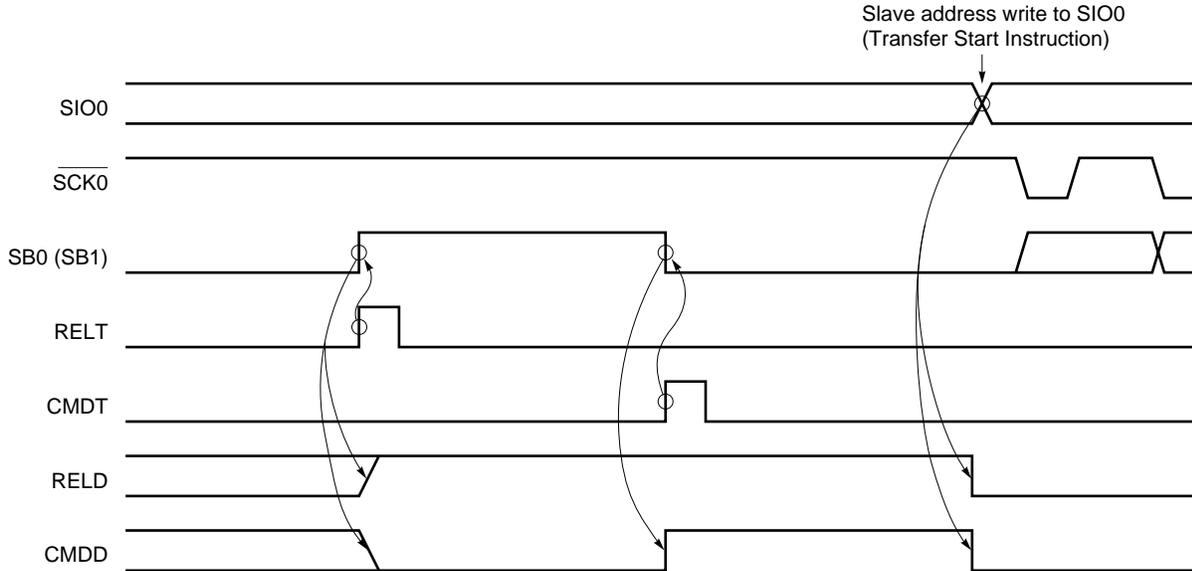


Figure 17-21. RELD and CMDD Operations (Slave)

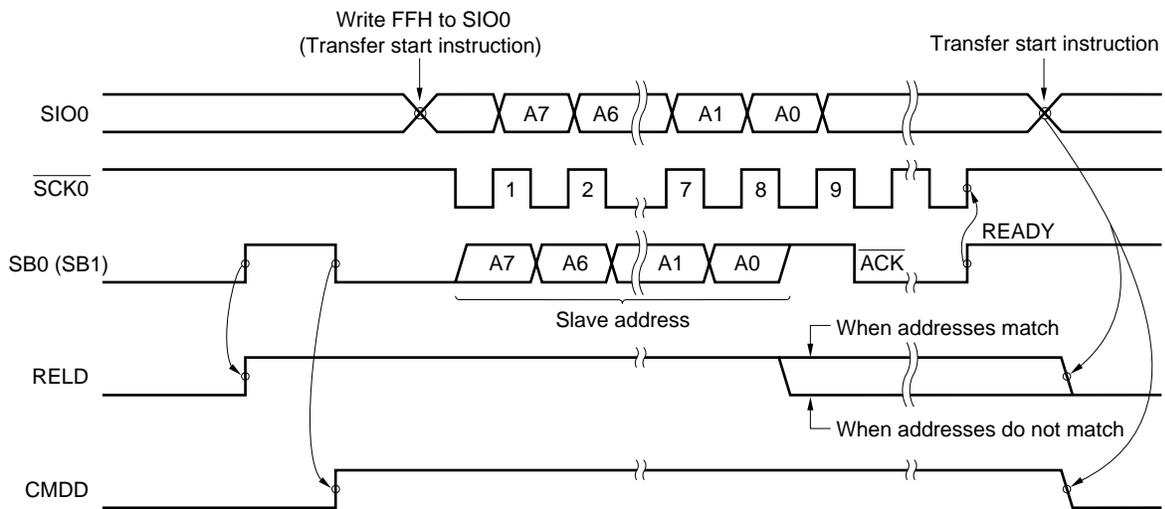
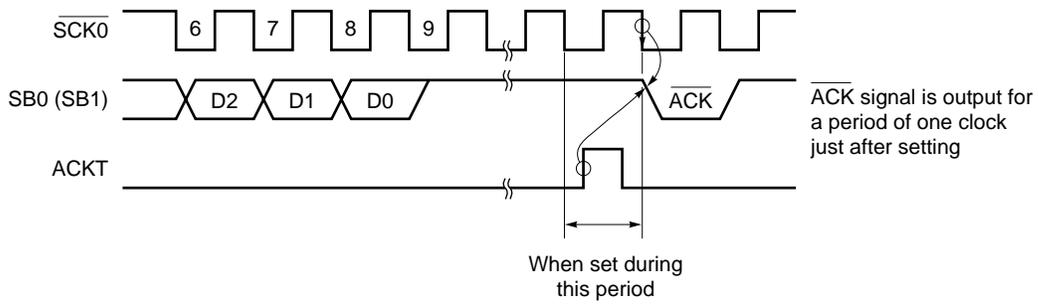


Figure 17-22. ACKT Operation



Caution Do not set ACKT before termination of transfer.

Figure 17-23. ACKE Operations

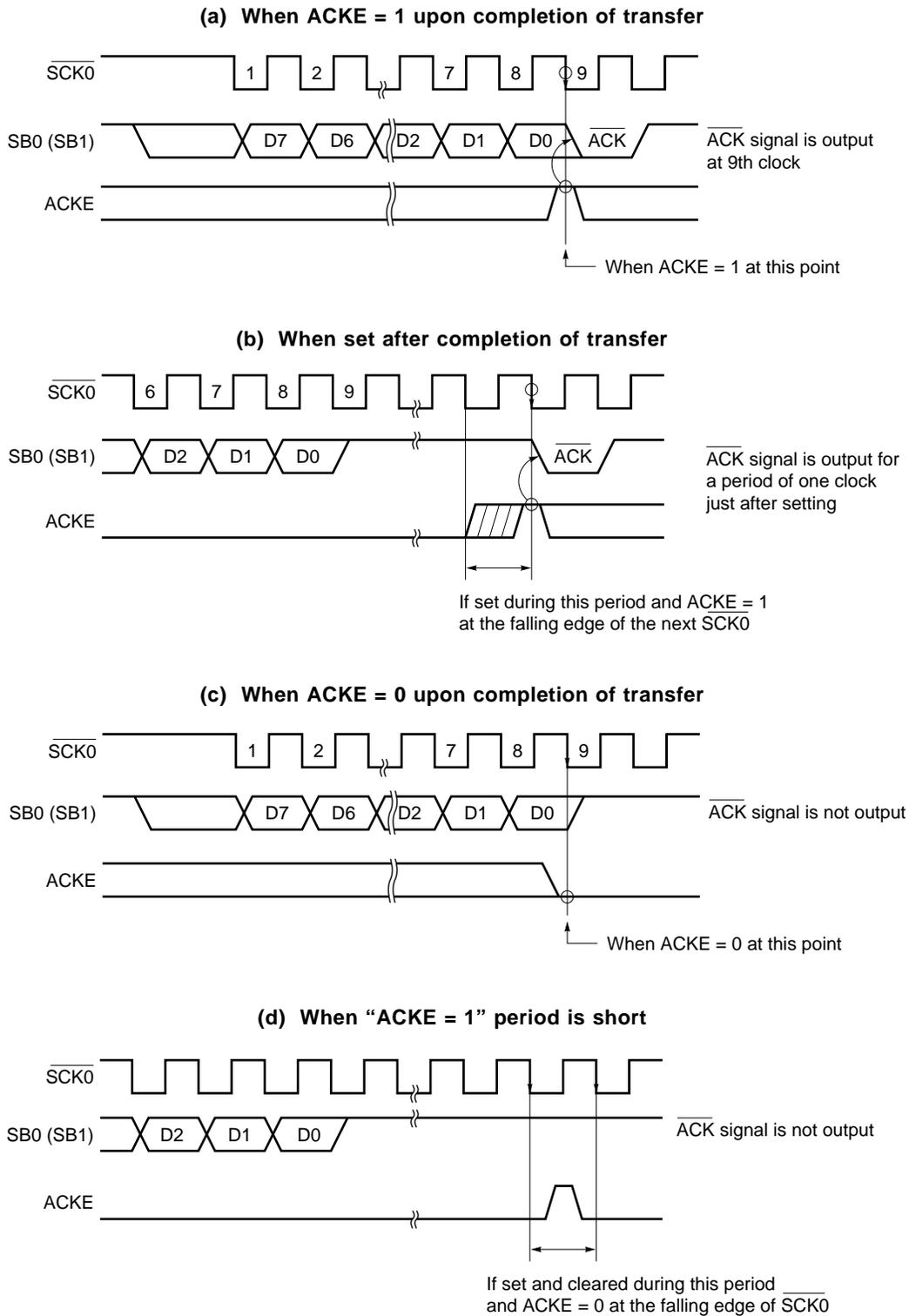


Figure 17-24. ACKD Operations

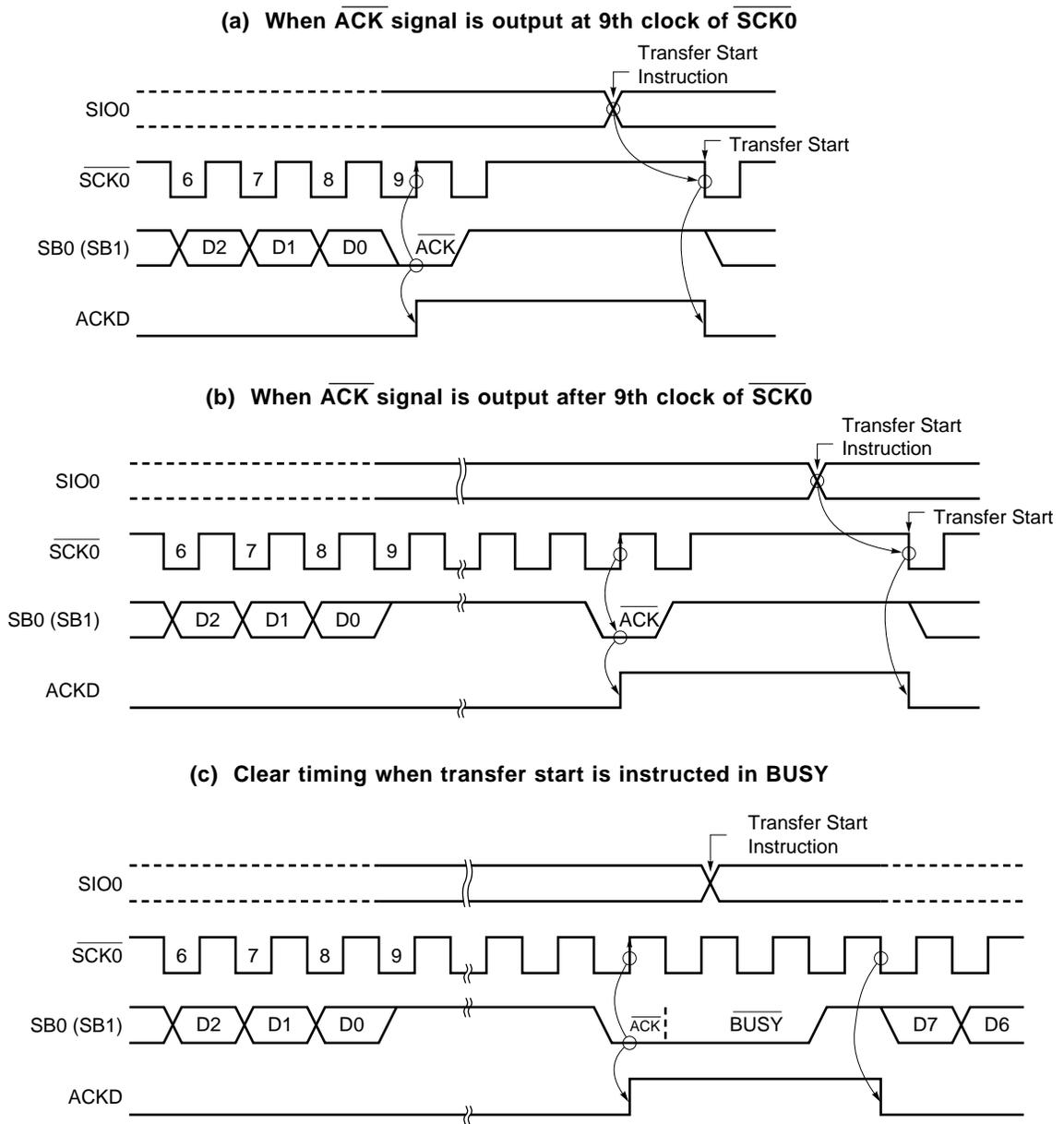


Figure 17-25. BSYE Operation

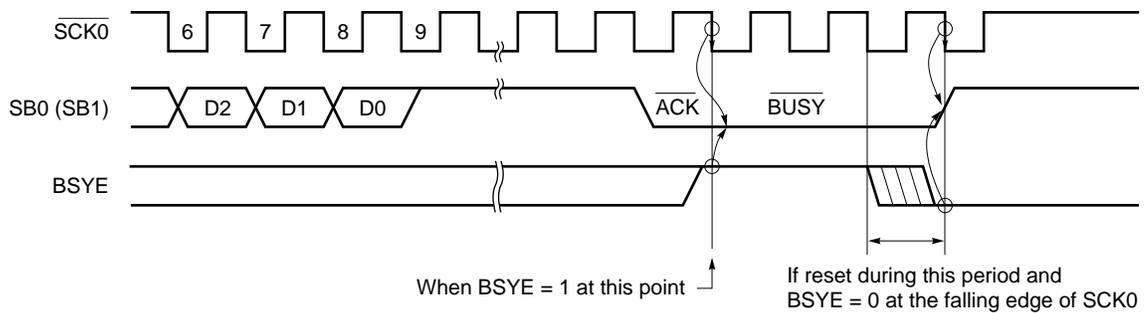


Table 17-3. Various Signals in SBI Mode (1/2)

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 $\overline{SCK0} = 1$		<ul style="list-style-type: none"> • RELT set 	<ul style="list-style-type: none"> • 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 $\overline{SCK0} = 1$		<ul style="list-style-type: none"> • CMDT set 	<ul style="list-style-type: none"> • 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]	(1) ACKE = 1 (2) ACKT set	<ul style="list-style-type: none"> • 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		<ul style="list-style-type: none"> • BSYE = 1 	—	Serial receive disable because of processing
Ready signal (READY)	Slave	High-level signal to be output to SB0 (SB1) before serial transfer start and after completion of serial transfer		(1) BSYE = 0 (2) Execution of instruction for data write to SIO0 (transfer start instruction)	—	Serial receive enable

Table 17-3. Various Signals in SBI Mode (2/2)

Signal Name	Output Device	Definition	Timing Chart	Output Condition	Effects on Flag	Meaning of Signal
Serial clock (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.				Timing of signal output to serial data bus
Address (A7 to A0)	Master	8-bit data to be transferred in synchronization with SCK0 after output of REL and CMD signals		When CSIE0 = 1, execution of instruction for data write to SIO0 (serial transfer start instruction) ^{Note 2}	CSIF0 set (rising edge of 9th clock of SCK0) ^{Note 1}	Address value of slave device on the serial bus
Commands (C7 to C0)	Master	8-bit data to be transferred in synchronization with 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 SCK0 without output of REL and CMD signals				Numeric values to be processed with slave or master device

- Notes**
1. When WUP = 0, CSIF0 is set at the rising edge of the 9th clock of SCK0.
When WUP = 1, an address is received. Only when the address matches the slave address register (SVA) value, CSIF0 is set (when they do not match, RELD is cleared).
 2. In \overline{BUSY} state, transfer starts after the READY state is set.

(5) Pin configuration

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

(a) $\overline{\text{SCK0}}$ Serial clock input/output pin

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

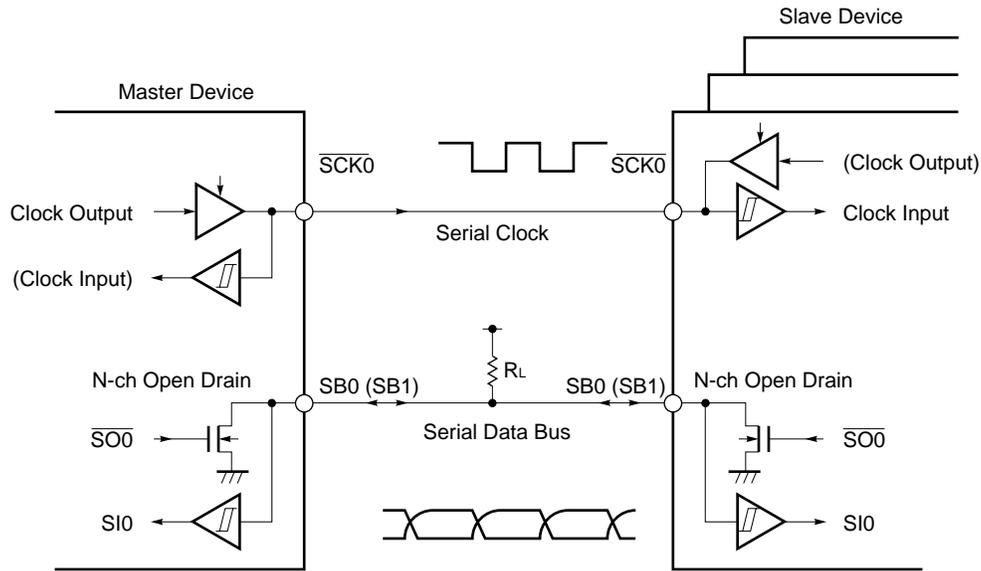
<2> Slave Schmitt input

(b) SB0 (SB1) Serial data input/output dual-function pin

Both master and slave devices have an N-ch open drain output and a Schmitt input.

Because the serial data bus line has an N-ch open-drain output, an external pull-up resistor is necessary.

Figure 17-26. Pin Configuration



Caution Because the N-ch open-drain output must be high-impedance state at time of data reception, write FFH to the serial I/O shift register 0 (SIO0) in advance. The N-ch open-drain can be high-impedance state at any time of transfer. However, when the wake-up function specify bit (WUP) = 1, the N-ch open-drain output is always 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 can be selected by transmitting slave address from the master device.

Address match detection can be automatically executed by hardware. With slave address register, CSIF0 is set only when the wake-up function specify bit (WUP) = 1 and the address transmitted from the master device matches the value set to SVA.

When bit 5 (SIC) of the interrupt timing specify register (SINT) is set (1), the wake-up function does not operate even if WUP is set (1) (the interrupt request signal is generated when a bus release is detected). When using the wake-up function, clear SIC to 0.

Cautions 1. Slave selection/non-selection is detected by matching of the slave address received after bus release (RELD = 1).

For this match detection, match interrupt request (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 17-27 to 17-30 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 ($\overline{SCK0}$). 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 $\overline{SCK0}$ is latched into the SIO0.

Figure 17-27. Address Transmission from Master Device to Slave Device (WUP = 1)

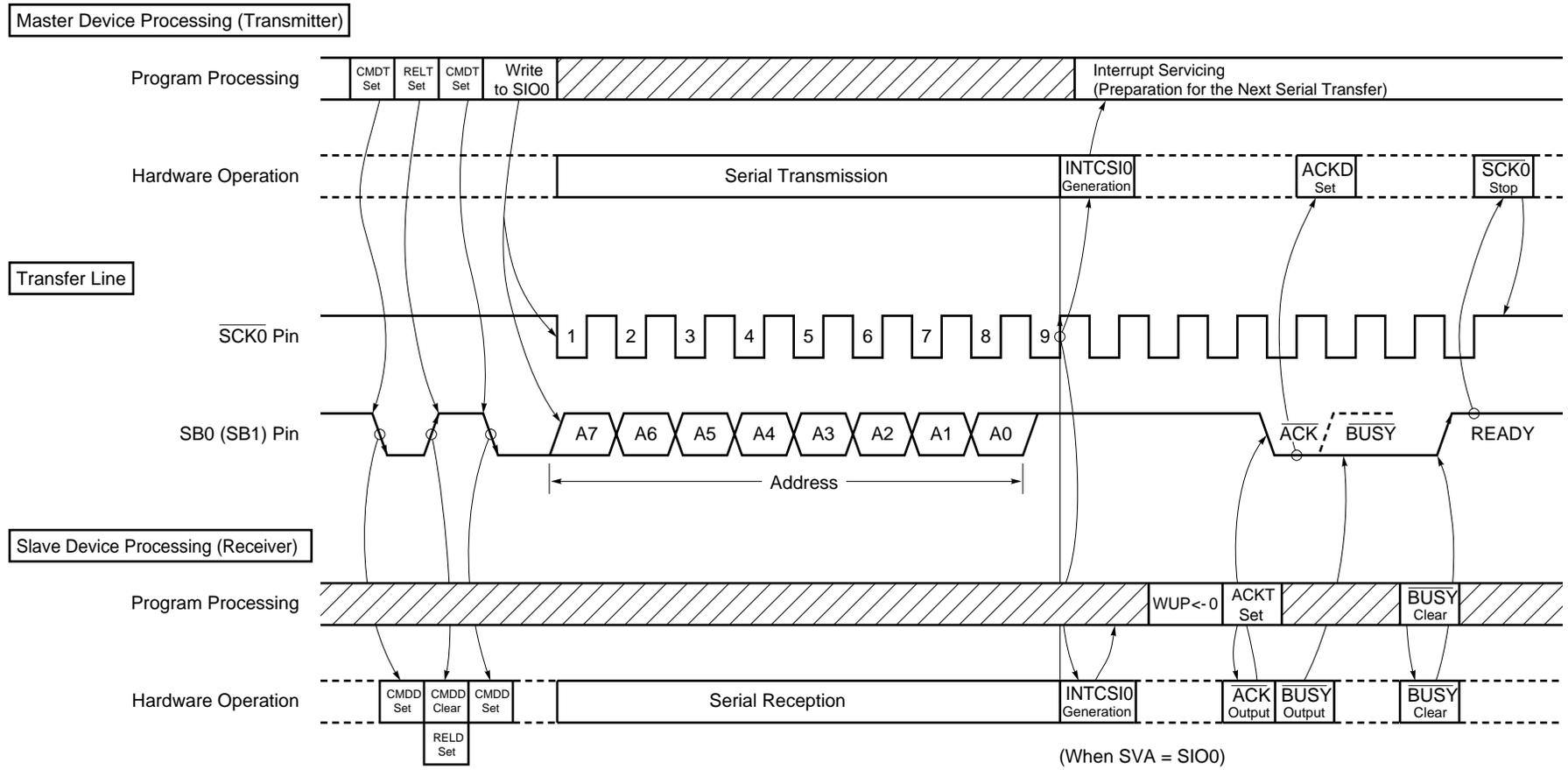


Figure 17-28. Command Transmission from Master Device to Slave Device

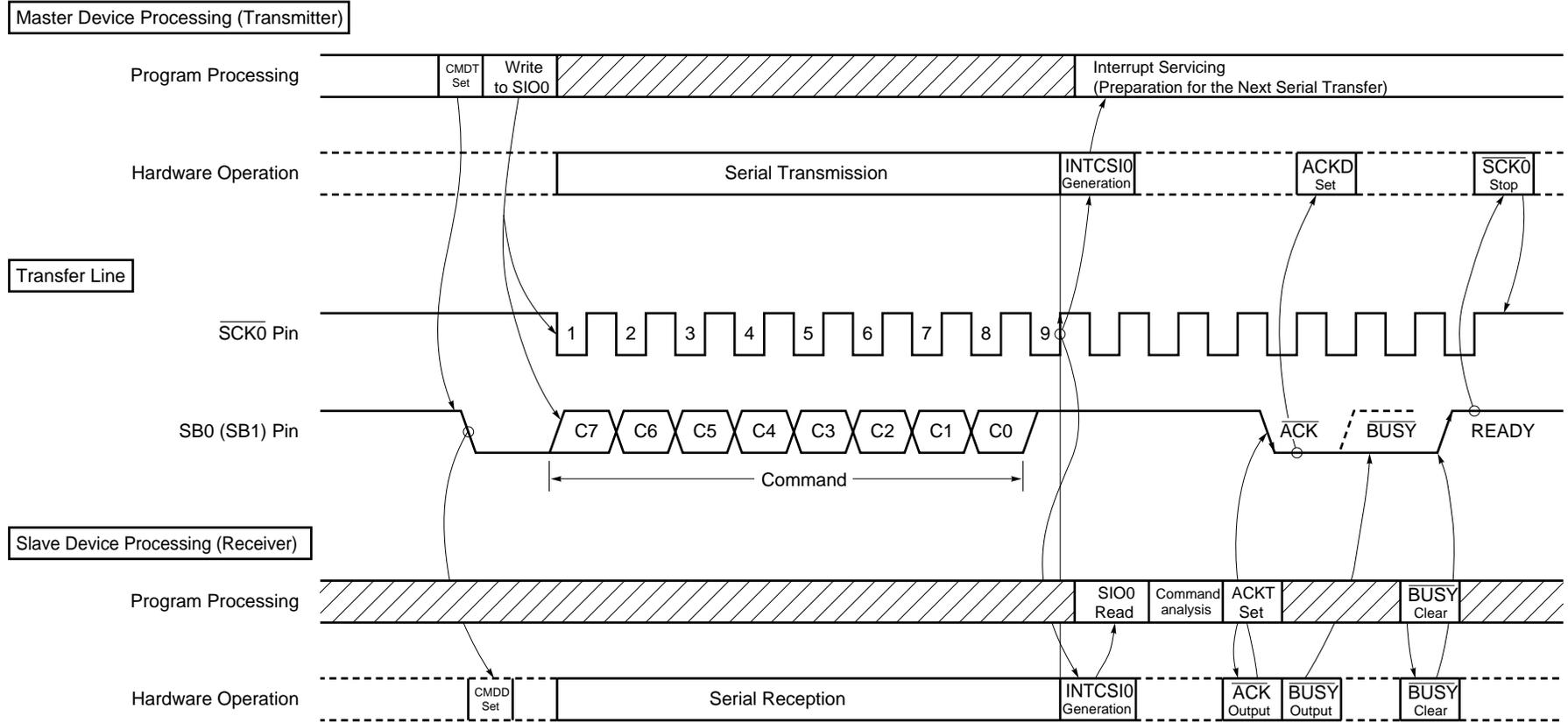


Figure 17-29. Data Transmission from Master Device to Slave Device

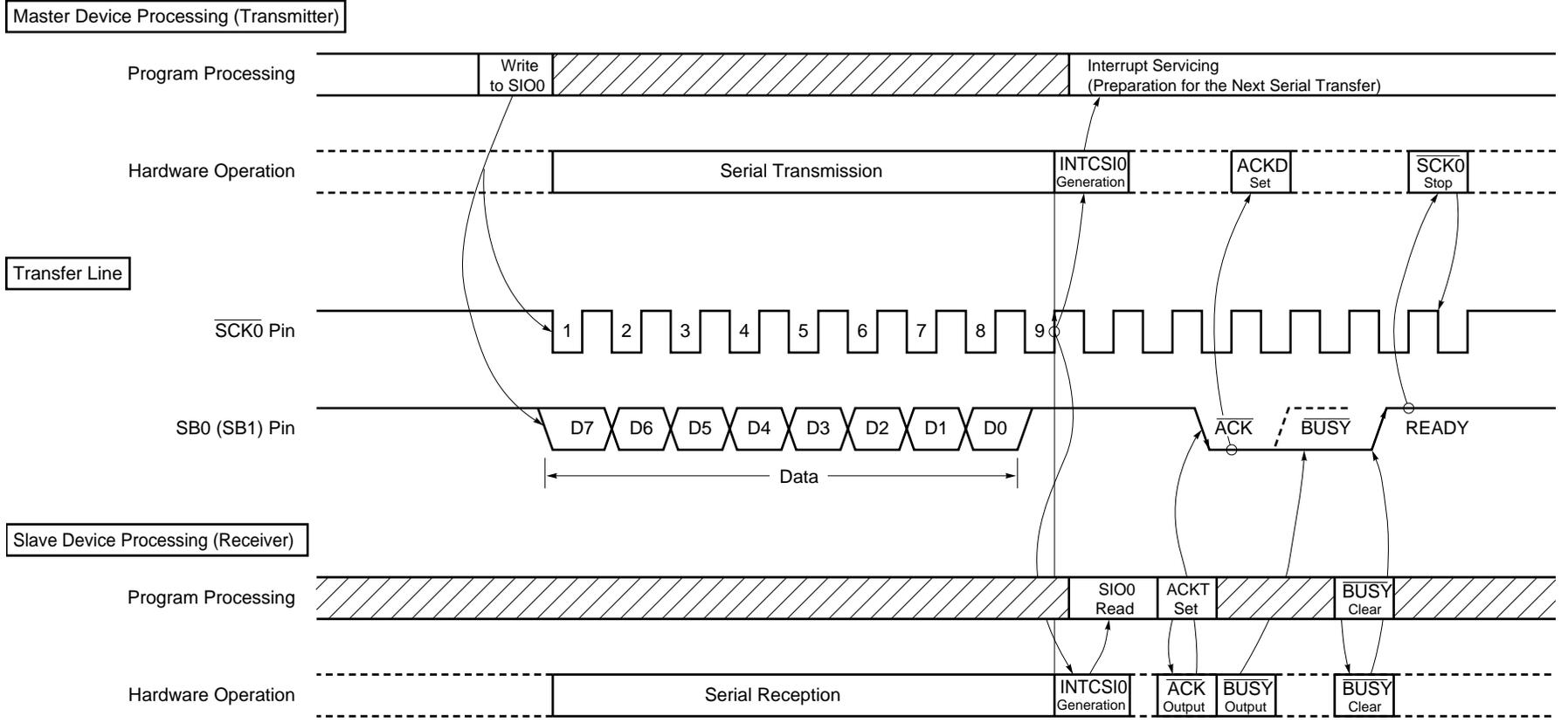
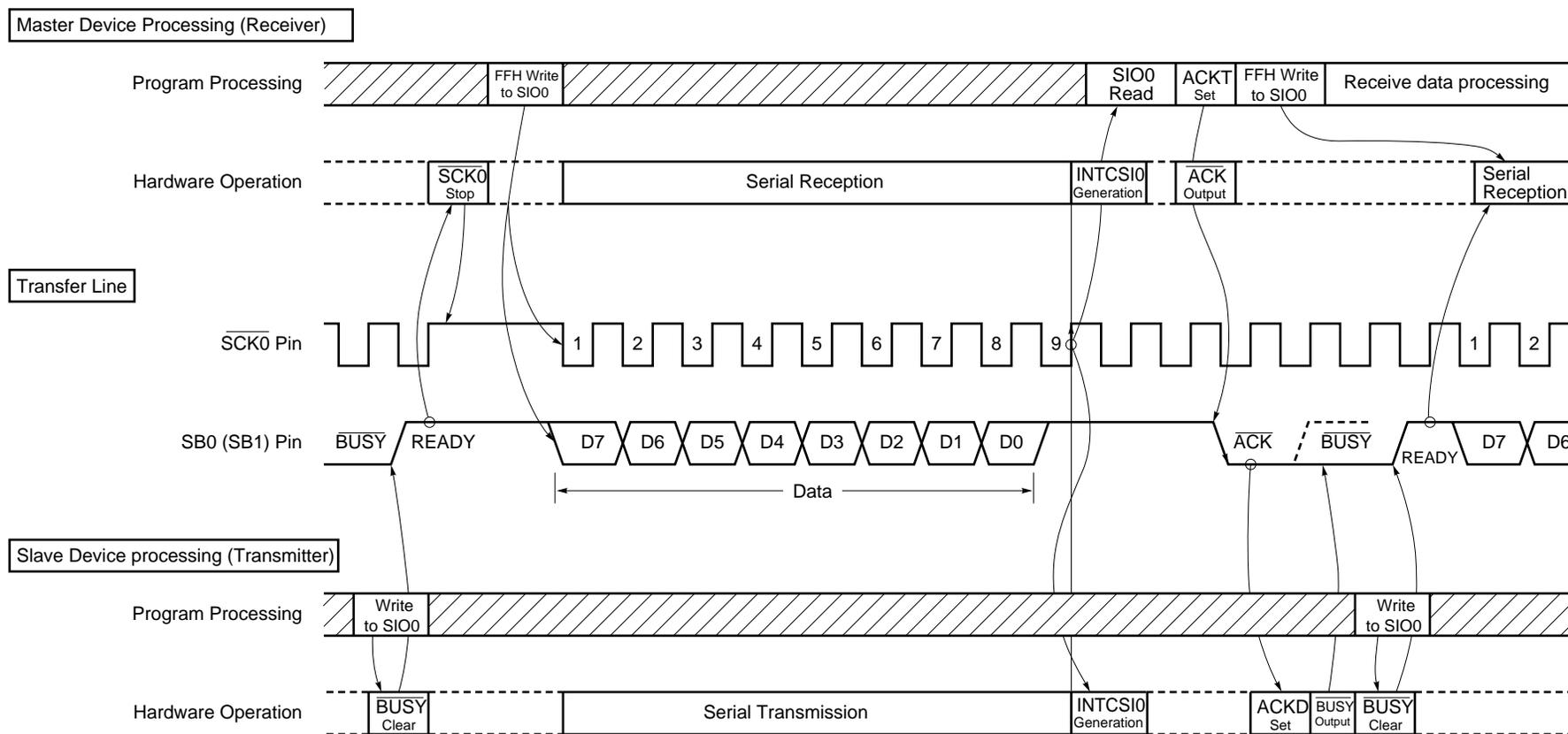


Figure 17-30. Data Transmission from Slave Device to Master Device



(9) Transfer start

Serial transfer is started by setting transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied.

- Serial interface channel 0 operation control bit (CSIE0) = 1
- Internal serial clock is stopped or $\overline{\text{SCK0}}$ is at high level after 8-bit serial transfer.

Cautions 1. If CSIE0 is set to “1” after data write to SIO0, transfer does not start.

- 2. Because the N-ch open-drain output must be high-impedance state for data reception, write FFH to SIO0 in advance.**

However, when the wake-up function specify bit (WUP) = 1, the N-ch open-drain output is always 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 end of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (CSIF0) is set. After $\overline{\text{RESET}}$ input, perform the following settings to the pins used as data input/output (SB0 or SB1) before serial transfer of the first byte.

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

(10) How to detect the busy state in a slave

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

- <1> Detect acknowledge signal ($\overline{\text{ACK}}$) or interrupt request signal generation.
- <2> Set the port mode register PM25 (or PM26) of the SB0/P25 (or SB1/P26) pin into the input mode.
- <3> Read out the pin state (when the pin level is high, the READY state is set).

After the detection of the READY state, set the port mode register to 0 and return to the output mode.

(11) SBI mode precautions

- (a) Slave selection/non-selection is detected by match detection of the slave address received after bus release (RELD = 1).
For this match detection, match interrupt (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, the $\overline{\text{BUSY}}$ signal continues to be output after $\overline{\text{BUSY}}$ clear instruction generation to the falling edge of the next serial clock ($\overline{\text{SCK0}}$). If WUP is set to 1 in this period, $\overline{\text{BUSY}}$ will not be released. Therefore, before setting WUP to 1, clear $\overline{\text{BUSY}}$ and then check that SB0 (SB1) pin has become high level.

- (d) For pins which are to be used for data input/output, be sure to carry out the following settings before serial transfer of the 1st byte after RESET input.

<1> Set 1 to the output latch of P25 and P26

<2> Set 1 to bit 0 (RELT) of the serial bus interface control register (SBIC).

<3> Set 0 to the output latch of P25 and P26, to which 1 has been set.

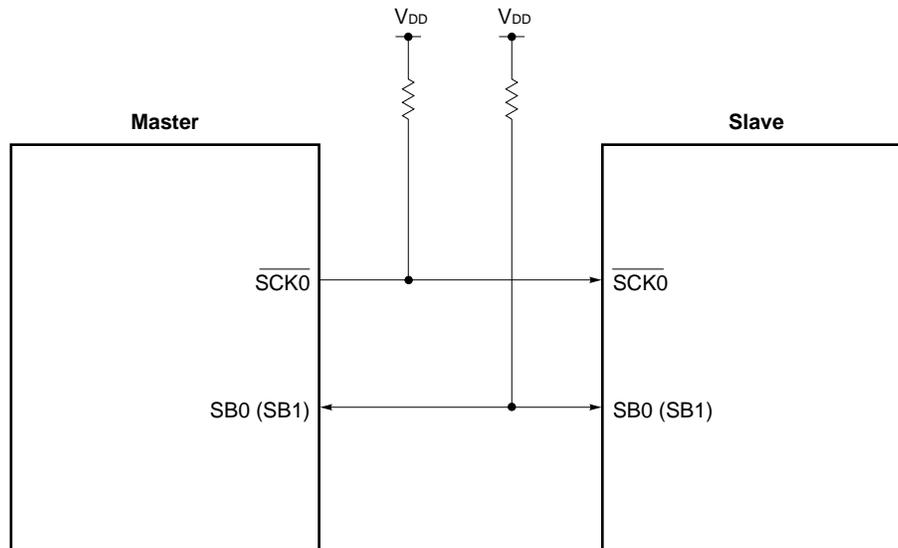
- (e) If the SB0 (SB1) line changes from low level to high level or from high level to low level while the $\overline{\text{SCK0}}$ line is at high level, it is recognized as either a bus release signal or a command signal. Therefore, if the changing timing of bus fluctuates because of the wiring capacitance, etc., this may be wrongly interpreted as a bus release signal (or a command signal) even while data is being transmitted. Care should be taken in the wiring.

17.4.4 2-wire serial I/O mode operation

The 2-wire serial I/O mode can cope with any communication format by program.

Communication is basically carried out with two lines of serial clock ($\overline{\text{SCK0}}$) and serial data input/output (SB0 or SB1).

Figure 17-31. 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.

$\overline{\text{RESET}}$ input sets CSIM0 to 00H.

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W ^{Note 1}

R/W	CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection
	0	x	Input Clock to $\overline{\text{SCK0}}$ pin from off-chip
	1	0	8-bit timer register 2 (TM2) output
	1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W	CSIM04	CSIM03	CSIM02	PM25	P25	PM26	P26	PM27	P27	Operation Mode	Start Bit	SIO/SB0/P25 Pin Function	SO0/SB1/P26 Pin Function	$\overline{\text{SCK0}}$ /P27 Pin Function
	0	x	3-wire Serial I/O mode (See 17.4.2 3-wire serial I/O mode operation.)											
	1	0	SBI mode (See 17.4.3 SBI mode operation.)											
	1	1	0	^{Note 2} x	^{Note 2} x	0	0	0	1	2-wire serial I/O mode	MSB	P25 (CMOS input/output)	SB1 (N-ch open-drain input/output)	$\overline{\text{SCK0}}$ (N-ch open-drain input/output)
			1	0	0	^{Note 2} x	^{Note 2} x	0	1			SB0 (N-ch open-drain input/output)	P26 (CMOS input/output)	

R/W	WUP	Wake-up Function Control ^{Note 3}
	0	Interrupt request signal generation with each serial transfer in any mode
	1	Interrupt request signal generation when the address received after bus release (when CMDD = RELD = 1) matches the slave address register (SVA) data in SBI mode

R	COI	Slave Address Comparison Result Flag ^{Note 4}
	0	Slave address register (SVA) not equal to serial I/O shift register 0 (SIO0) data
	1	Slave address register (SVA) equal to serial I/O shift register 0 (SIO0) 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. Set WUP to 0 when the 2-wire serial I/O mode is selected.
 4. When CSIE0 = 0, COI becomes 0.

Remark x : Don't care
 PMxx : Port mode register
 Pxx : Port output latch

(b) Serial bus interface control register (SBIC)

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

$\overline{\text{RESET}}$ input 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, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.
-----	------	---

R/W	CMDT	When CMDT = 1, SO0 latch is cleared to 0. After SO0 latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.
-----	------	---

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

(c) Interrupt timing specify register (SINT)

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

$\overline{\text{RESET}}$ input 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 Factor Selection
	0	CSIIF0 is set upon termination of serial interface channel 0 transfer
	1	CSIIF0 is set upon bus release detection or termination of serial interface channel 0 transfer

R	CLD	$\overline{\text{SCK0/P27}}$ 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 Set bits 0 to 3 to 0.

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

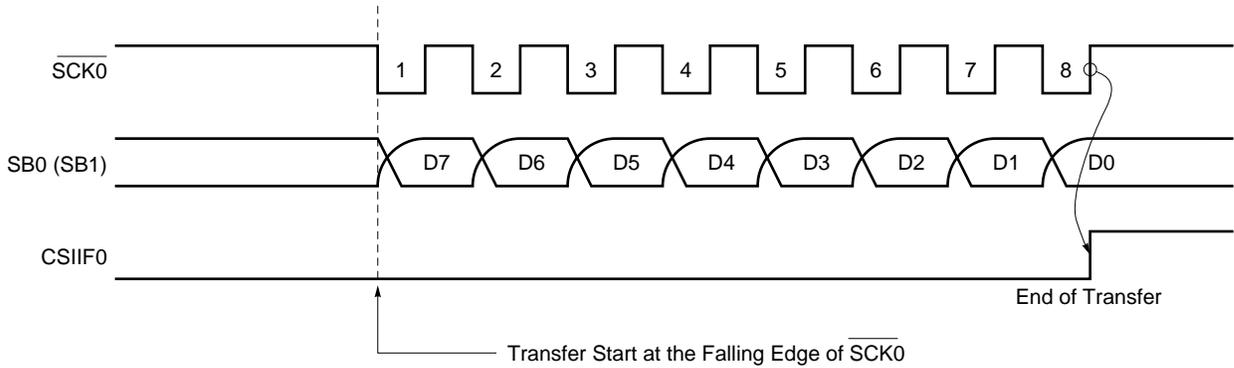
(2) Communication operation

The 2-wire serial I/O mode is used for data transmission/reception in 8-bit units. Data transmission/reception is carried out bit-wise in synchronization with the serial clock.

Shift operation of the serial I/O shift register 0 (SIO0) is carried out in synchronization with the falling edge of the serial clock ($\overline{\text{SCK0}}$). The transmit data is held in the SO0 latch and is output from the SB0/P25 (or SB1/P26) pin on an MSB-first basis. The receive data input from the SB0 (or SB1) pin is latched into the SIO0 at the rising edge of $\overline{\text{SCK0}}$.

Upon termination of 8-bit transfer, the SIO0 operation stops automatically and the interrupt request flag (CSIF0) is set.

Figure 17-32. 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 pin and thus it must be externally connected to a pull-up resistor. Because it is necessary to set the N-ch open-drain output to high-impedance state for data reception, write FFH to SIO0 in advance.

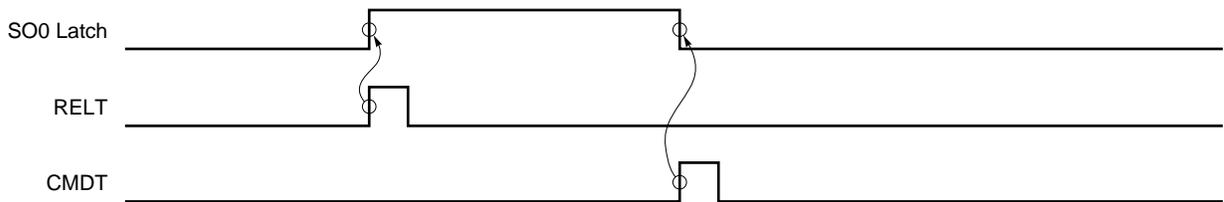
The SB0 (or SB1) pin generates the SO0 latch status and thus the SB0 (or SB1) pin output status can be manipulated by setting bit 0 (RELT) and bit 1 (CMDT) of the serial bus interface control register (SBIC). However, do not carry out this manipulation during serial transfer.

Control the $\overline{\text{SCK0}}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **17.4.5 $\overline{\text{SCK0}}$ /P27 pin output manipulation**).

(3) Other signals

Figure 17-33 shows RELT and CMDT operations.

Figure 17-33. RELT and CMDT Operations



(4) Transfer start

Serial transfer is started by setting transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied.

- Serial interface channel 0 operation control bit (CSIE0) = 1
- Internal serial clock is stopped or $\overline{\text{SCK0}}$ is at high level after 8-bit serial transfer.

Cautions 1. If CSIE0 is set to “1” after data write to SIO0, transfer does not start.

- 2. Because the N-ch transistor must be set N-ch open-drain output to 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 (CSIF0) is set.

(5) Error detection

In the 2-wire serial I/O mode, the serial bus SB0 (SB1) status being transmitted is fetched into the destination device, that is, the serial I/O shift register 0 (SIO0). Thus, transmit error can be detected in the following 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.

17.4.5 $\overline{\text{SCK0/P27}}$ pin output manipulation

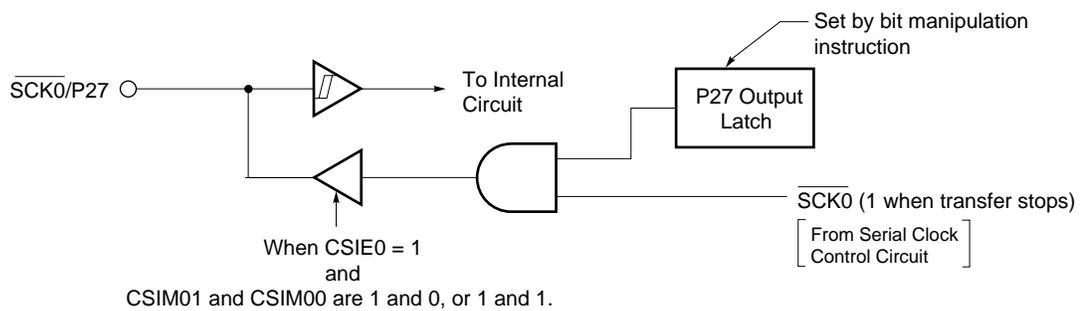
Because the $\overline{\text{SCK0/P27}}$ pin incorporates an output latch, static output is also possible by software in addition to normal serial clock output.

P27 output latch manipulation enables setting any value to $\overline{\text{SCK0}}$ by software (SI0/SB0 and SO0/SB1 pin to be controlled with the RELT and CMDT bits of the serial bus interface control register (SBIC)).

$\overline{\text{SCK0/P27}}$ pin output manipulating procedure is described below.

- (1) Set the serial operating mode register 0 (CSIM0) ($\overline{\text{SCK0}}$ pin enabled for serial operation in the output mode).
 $\overline{\text{SCK0}} = 1$ with serial transfer suspended.
- (2) Manipulate the P27 output latch with a bit manipulation instruction.

Figure 17-34. $\overline{\text{SCK0/P27}}$ Pin Configuration



[MEMO]

CHAPTER 18 SERIAL INTERFACE CHANNEL 0 (μ PD78078Y Subseries)

The μ PD78078Y Subseries incorporates three channels of serial interfaces. Differences between channels 0, 1, and 2 are as follows (Refer to **CHAPTER 19 SERIAL INTERFACE CHANNEL 1** for details of the serial interface channel 1. Refer to **CHAPTER 20 SERIAL INTERFACE CHANNEL 2** for details of the serial interface channel 2).

Table 18-1. Differences between Channels 0, 1, and 2

Serial Transfer Mode		Channel 0	Channel 1	Channel 2
3-wire serial I/O	Clock selection	$f_{xx}/2$, $f_{xx}/2^2$, $f_{xx}/2^3$, $f_{xx}/2^4$, $f_{xx}/2^5$, $f_{xx}/2^6$, $f_{xx}/2^7$, $f_{xx}/2^8$, external clock, TO2 output	$f_{xx}/2$, $f_{xx}/2^2$, $f_{xx}/2^3$, $f_{xx}/2^4$, $f_{xx}/2^5$, $f_{xx}/2^6$, $f_{xx}/2^7$, $f_{xx}/2^8$, external clock, TO2 output	External clock, baud rate generator output
	Transfer method	MSB/LSB switchable as the start bit	MSB/LSB switchable as the start bit Automatic transmit/receive function	MSB/LSB switchable as the start bit
	Transfer end flag	Serial transfer end interrupt request flag (CSIF0)	Serial transfer end interrupt request flag (CSIF1)	Serial transfer end interrupt request flag (SRIF)
I ² C bus (Inter IC Bus)	2-wire serial I/O	Use possible	None	None
UART (Asynchronous serial interface)				

18.1 Serial Interface Channel 0 Functions

Serial interface channel 0 employs the following four modes.

- Operation stop mode
- 3-wire serial I/O mode
- 2-wire serial I/O mode
- I²C (Inter IC) bus mode

★ **Caution** Do not change the operation mode (3-wire serial I/O/2-wire serial I/O/I²C bus) while the operation of the serial interface channel 0 is enabled. Stop the serial operation before changing the operation mode.

(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 ($\overline{\text{SCK0}}$), 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) 2-wire serial I/O mode (MSB-first)

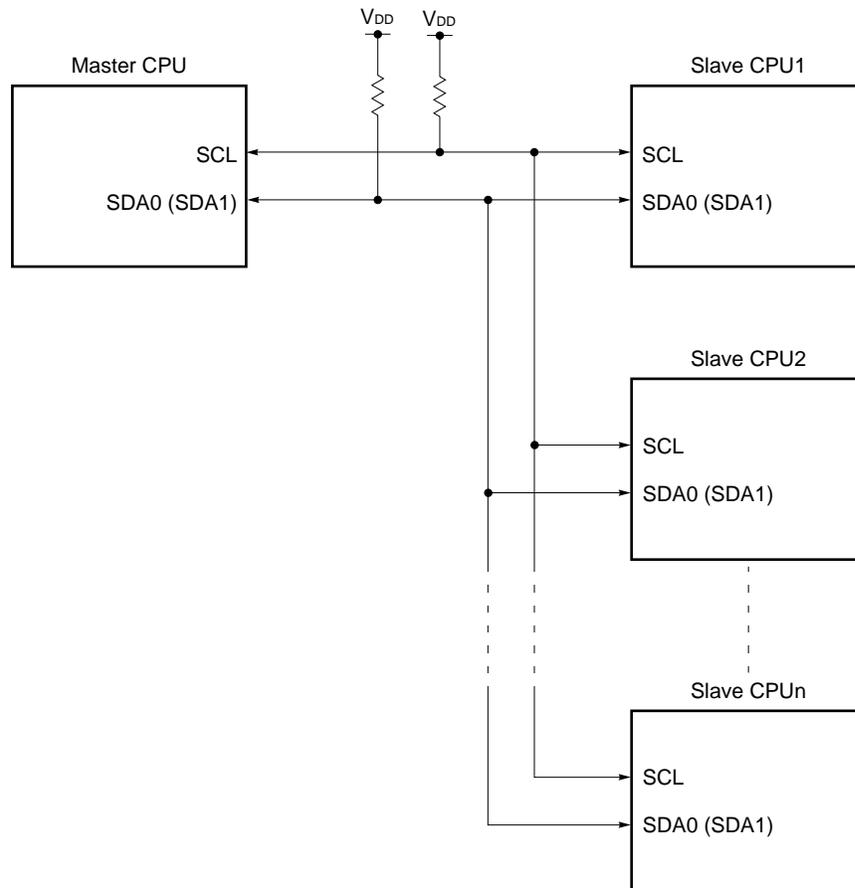
This mode is used for 8-bit data transfer using two lines of serial clock ($\overline{\text{SCK0}}$) and serial data bus (SB0 or SB1).

This mode enables to cope with any one of the possible data transfer formats by controlling the $\overline{\text{SCK0}}$ 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.

(4) I²C (Inter IC) 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 I²C 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 18-1. Serial Bus Configuration Example Using I²C Bus

18.2 Serial Interface Channel 0 Configuration

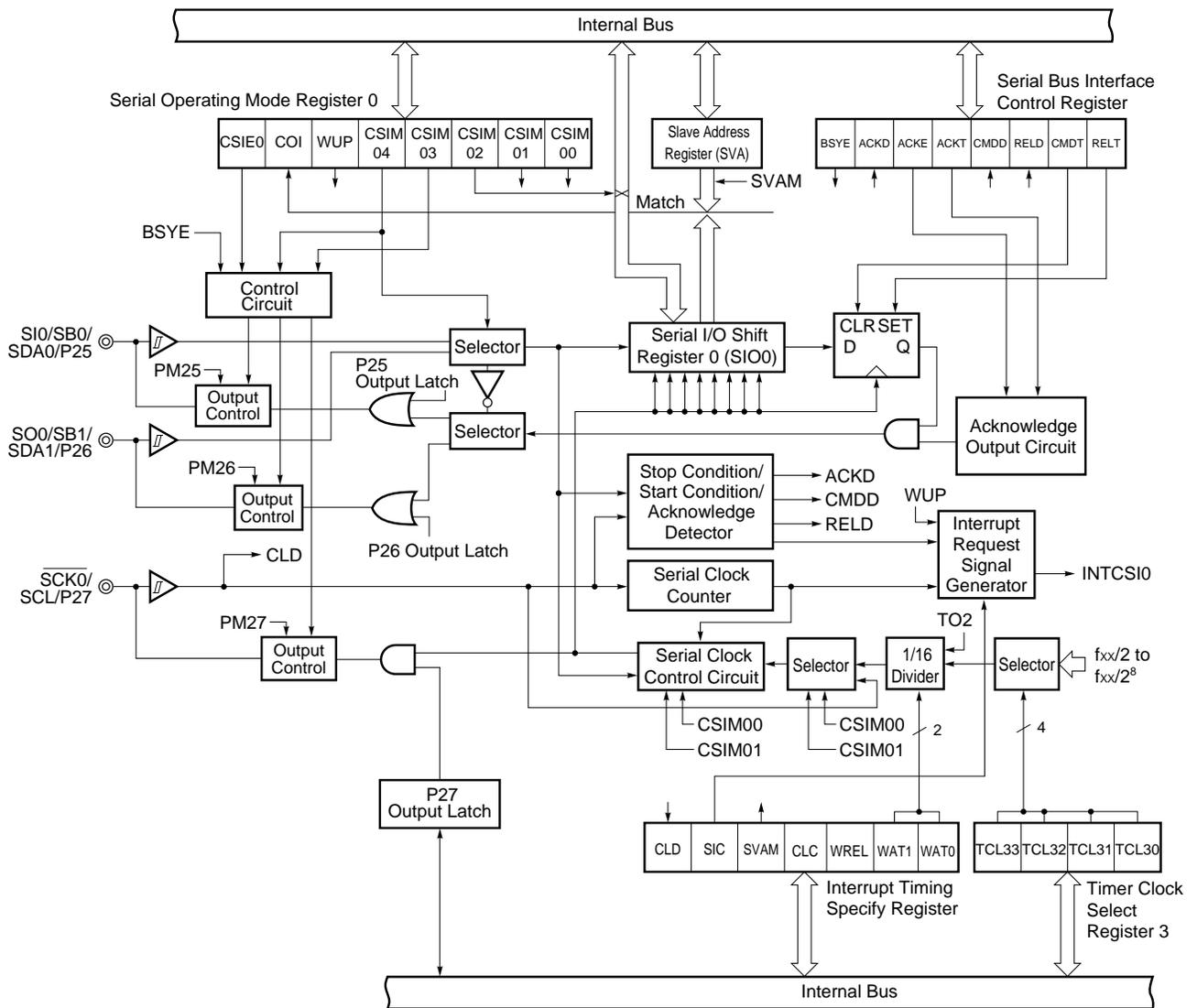
Serial interface channel 0 consists of the following hardware.

Table 18-2. 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 specify register (SINT) Port mode register 2 (PM2) ^{Note}

Note Refer to **Figure 6-7 Block Diagram of P20, P21, P23 to P26** and **Figure 6-8 Block Diagram of P22 and P27**.

Figure 18-2. Serial Interface Channel 0 Block Diagram



Remark Output Control performs selection between CMOS output and N-ch open-drain output.

(1) Serial I/O shift register 0 (SIO0)

This is an 8-bit register to carry out parallel-serial conversion and to carry out serial transmission/reception (shift operation) in synchronization with the serial clock.

SIO0 is set with an 8-bit memory manipulation instruction.

When bit 7 (CSIE0) of serial operating mode register 0 (CSIM0) is 1, writing data to SIO0 starts serial operation.

In transmission, data written to SIO0 is output to the serial output (SO0) or serial data bus (SB0/SB1). In reception, data is read from the serial input (SI0) or SB0/SB1 to SIO0.

Note that, if a bus is driven in the I²C bus mode or 2-wire serial I/O mode, the bus pin must serve for both input and output. Therefore, the transmission N-ch open-drain output of the device which will start reception of data must be set to high-impedance state beforehand. Consequently, write FFH to SIO0 in advance. In the I²C bus mode, set SIO0 to FFH with bit 7 (BSYE) of the serial bus interface control register (SBIC) set to 0.

$\overline{\text{RESET}}$ input makes SIO0 undefined.

Caution In the I²C bus mode, do not execute write instructions to SI0 while WUP (bit 5 of serial operation mode register 0 (CSIM0)) = 1. Data can be received when using the wake-up function (WUP = 1) even if write instruction to SIO0 is not executed. For the wake-up function, refer to 18.4.4 (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. This register is not used in the 3-wire serial I/O mode.

The master device outputs a slave address for selection of a particular slave device to the connected slave device. These two data (the slave address output from the master device and the SVA value) are compared with an address comparator. If they match, the slave device has been selected. In that case, bit 6 (COI) of serial operating mode register 0 (CSIM0) becomes 1.

Address of the data of LSB-masked high-order 7 bits can be compared by setting bit 4 (SVAM) of the interrupt timing specify register (SINT).

If no matching is detected in address reception, bit 2 (RELD) of the serial bus interface control register (SBIC) is cleared to 0. In the I²C bus mode, when bit 5 (WUP) of CSIM0 is set (1), the wake-up function can be used. In this case, the interrupt request signal (INTCSI0) is generated if the slave address output from the master and the SVA value match (the interrupt request signal is generated also when a stop condition is detected). This interrupt request enables to recognize the generation of the communication request from the master device. Set SIC to 1 when using the wake-up function.

Further, when SVA transmits data as master or slave device in the the I²C bus mode or 2-wire serial I/O mode, errors are detected if any.

$\overline{\text{RESET}}$ 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.

(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 $\overline{\text{SCK0}}/\text{SCL}/\text{P27}$ pin.

(6) Interrupt request signal generator

This circuit controls interrupt request signal generation. It generates interrupt request signals according to the settings of interrupt timing specification register (SINT) bits 0 and 1 (WAT0, WAT1) and serial operation mode register 0 (CSIM0) bit 5 (WUP), as shown in Table 18-3.

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

These two circuits output and detect various control signals in the I²C mode.

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

Table 18-3. Serial Interface Channel 0 Interrupt Request Signal Generation

Serial Transfer Mode	BSYE	WUP	WAT1	WAT0	ACKE	Description
3-wire or 2-wire serial I/O mode	0	0	0	0	0	An interrupt request signal is generated each time 8 serial clocks are counted.
	Other than above					Setting prohibited
I ² C 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 ² C 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 request signal 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 and if stop condition is detected, an interrupt request signal is 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)

18.3 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.

$\overline{\text{RESET}}$ input sets TCL3 to 88H.

Figure 18-3. Timer Clock Select Register 3 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33	TCL32	TCL31	TCL30	Serial Interface Channel 0 Serial Clock Selection					
				Serial Clock in I ² C Bus Mode			Serial Clock in 2-Wire or 3-Wire Serial I/O Mode		
				MCS = 1		MCS = 0	MCS = 1		MCS = 0
0	1	1	0	$f_{xx}/2^5$	Setting prohibited	$f_x/2^6$ (78.1 kHz)	$f_{xx}/2$	Setting prohibited	$f_x/2^2$ (1.25 MHz)
0	1	1	1	$f_{xx}/2^6$	$f_x/2^6$ (78.1 kHz)	$f_x/2^7$ (39.1 kHz)	$f_{xx}/2^2$	$f_x/2^2$ (1.25 MHz)	$f_x/2^3$ (625 kHz)
1	0	0	0	$f_{xx}/2^7$	$f_x/2^7$ (39.1 kHz)	$f_x/2^8$ (19.5 kHz)	$f_{xx}/2^3$	$f_x/2^3$ (625 kHz)	$f_x/2^4$ (313 kHz)
1	0	0	1	$f_{xx}/2^8$	$f_x/2^8$ (19.5 kHz)	$f_x/2^9$ (9.77 kHz)	$f_{xx}/2^4$	$f_x/2^4$ (313 kHz)	$f_x/2^5$ (156 kHz)
1	0	1	0	$f_{xx}/2^9$	$f_x/2^9$ (9.77 kHz)	$f_x/2^{10}$ (4.88 kHz)	$f_{xx}/2^5$	$f_x/2^5$ (156 kHz)	$f_x/2^6$ (78.1 kHz)
1	0	1	1	$f_{xx}/2^{10}$	$f_x/2^{10}$ (4.88 kHz)	$f_x/2^{11}$ (2.44 kHz)	$f_{xx}/2^6$	$f_x/2^6$ (78.1 kHz)	$f_x/2^7$ (39.1 kHz)
1	1	0	0	$f_{xx}/2^{11}$	$f_x/2^{11}$ (2.44 kHz)	$f_x/2^{12}$ (1.22 kHz)	$f_{xx}/2^7$	$f_x/2^7$ (39.1 kHz)	$f_x/2^8$ (19.5 kHz)
1	1	0	1	$f_{xx}/2^{12}$	$f_x/2^{12}$ (1.22 kHz)	$f_x/2^{13}$ (0.61 kHz)	$f_{xx}/2^8$	$f_x/2^8$ (19.5 kHz)	$f_x/2^9$ (9.8 kHz)
Other than above				Setting prohibited					

TCL37	TCL36	TCL35	TCL34	Serial Interface Channel 1 Serial Clock Selection		
				MCS = 1		MCS = 0
0	1	1	0	$f_{xx}/2$	Setting prohibited	$f_x/2^2$ (1.25 MHz)
0	1	1	1	$f_{xx}/2^2$	$f_x/2^2$ (1.25 MHz)	$f_x/2^3$ (625 kHz)
1	0	0	0	$f_{xx}/2^3$	$f_x/2^3$ (625 kHz)	$f_x/2^4$ (313 kHz)
1	0	0	1	$f_{xx}/2^4$	$f_x/2^4$ (313 kHz)	$f_x/2^5$ (156 kHz)
1	0	1	0	$f_{xx}/2^5$	$f_x/2^5$ (156 kHz)	$f_x/2^6$ (78.1 kHz)
1	0	1	1	$f_{xx}/2^6$	$f_x/2^6$ (78.1 kHz)	$f_x/2^7$ (39.1 kHz)
1	1	0	0	$f_{xx}/2^7$	$f_x/2^7$ (39.1 kHz)	$f_x/2^8$ (19.5 kHz)
1	1	0	1	$f_{xx}/2^8$	$f_x/2^8$ (19.5 kHz)	$f_x/2^9$ (9.8 kHz)
Other than above				Setting prohibited		

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

- Remarks**
1. f_{xx} : Main system clock frequency (f_x or $f_x/2$)
 2. f_x : Main system clock oscillation frequency
 3. MCS : Bit 0 of oscillation mode selection register (OSMS)
 4. Figures in parentheses apply to operation with $f_x = 5.0$ 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.

$\overline{\text{RESET}}$ input sets CSIM0 to 00H.

- ★ **Caution** Do not change the operation mode (3-wire serial I/O/2-wire serial I/O/I²C bus) while the operation of the serial interface channel 0 is enabled. Stop the serial operation before changing the operation mode.

Figure 18-4. Serial Operating Mode Register 0 Format

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	After Reset	R/W			
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W ^{Note 1}			
R/W	CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection											
	0	x	Input clock to SCK0/SCL pin from off-chip											
	1	0	8-bit timer register 2 (TM2) output ^{Note 2}											
	1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)											
R/W	CSIM04	CSIM03	CSIM02	PM25	P25	PM26	P26	PM27	P27	Operation Mode	Start Bit	SI0/SB0/SDA0/ P25 Pin Function	SO0/SB1/SDA1/ P26 Pin Function	SCK0/SCL/P27 Pin Function
	0	x	0	^{Note 3} 1	^{Note 3} x	0	0	0	1	3-wire serial I/O mode	MSB	SI0 ^{Note 3} (Input)	SO0 (CMOS output)	SCK0 (CMOS input/output)
			1						LSB					
	1	1	0	^{Note 4} x	^{Note 4} x	0	0	0	1	2-wire serial I/O mode or I ² C bus mode	MSB	P25 (CMOS input/output)	SB1/SDA1 (N-ch open-drain input/output)	SCK0/SCL (N-ch open-drain input/output)
			1	0	0	^{Note 4} x	^{Note 4} x	0	1			SB0/SDA0 (N-ch open-drain input/output)	P26 (CMOS input/output)	
R/W	WUP	Wake-up Function Control ^{Note 5}												
	0	Interrupt request signal generation with each serial transfer in any mode												
	1	Interrupt request signal generation when the address received after detecting start condition (when CMDD = 1) matches the slave address register (SVA) data in I ² C bus mode												
R	COI	Slave Address Comparison Result Flag ^{Note 6}												
	0	Slave address register (SVA) not equal to serial I/O shift register 0 (SIO0) data												
	1	Slave address register (SVA) equal to serial I/O shift register 0 (SIO0) 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. I²C bus mode, the clock frequency becomes 1/16 of that output from TO2.
 3. Can be used as P25 (CMOS input/output) when used only for transmission.
 4. Can be used freely as port function.
 5. Set bit 5 (SIC) of the interrupt timing specify register (SINT) to 1 when using the wake-up function (WUP = 1). Do not execute a write instruction to the serial I/O shift register 0 (SIO0) while WUP = 1.
 6. When CSIE0 = 0, COI becomes 0.

Remark x : Don't care
 PMxx : Port mode register
 Pxx : Port output latch

(3) Serial bus interface control register (SBIC)

This register sets serial bus interface operation and displays statuses.

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

$\overline{\text{RESET}}$ input sets SBIC to 00H.

Figure 18-5. Serial Bus Interface Control Register Format (1/2)

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After Reset	R/W																										
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W ^{Note}																										
R/W	<table border="1"> <tr> <td>RELT</td> <td>Used for stop condition signal output. When RELT = 1, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.</td> </tr> </table>											RELT	Used for stop condition signal output. When RELT = 1, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.																								
RELT	Used for stop condition signal output. When RELT = 1, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.																																				
R/W	<table border="1"> <tr> <td>CMDT</td> <td>Used for start condition signal output. When CMDT = 1, SO0 latch is cleared to 0. After SO0 latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.</td> </tr> </table>											CMDT	Used for start condition signal output. When CMDT = 1, SO0 latch is cleared to 0. After SO0 latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.																								
CMDT	Used for start condition signal output. When CMDT = 1, SO0 latch is cleared to 0. After SO0 latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.																																				
R	<table border="1"> <tr> <td>RELD</td> <td>Stop Condition Detection</td> </tr> <tr> <td colspan="5">Clear Conditions (RELD = 0)</td> <td colspan="7">Set Conditions (RELD = 1)</td> </tr> <tr> <td colspan="5"> <ul style="list-style-type: none"> • When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied </td> <td colspan="7"> <ul style="list-style-type: none"> • When stop condition signal is detected </td> </tr> </table>											RELD	Stop Condition Detection	Clear Conditions (RELD = 0)					Set Conditions (RELD = 1)							<ul style="list-style-type: none"> • When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied 					<ul style="list-style-type: none"> • When stop condition signal is detected 						
RELD	Stop Condition Detection																																				
Clear Conditions (RELD = 0)					Set Conditions (RELD = 1)																																
<ul style="list-style-type: none"> • When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied 					<ul style="list-style-type: none"> • When stop condition signal is detected 																																
R	<table border="1"> <tr> <td>CMDD</td> <td>Start Condition Detection</td> </tr> <tr> <td colspan="5">Clear Conditions (CMDD = 0)</td> <td colspan="7">Set Conditions (CMDD = 1)</td> </tr> <tr> <td colspan="5"> <ul style="list-style-type: none"> • When transfer start instruction is executed • When stop condition signal is detected • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied </td> <td colspan="7"> <ul style="list-style-type: none"> • When start condition signal is detected </td> </tr> </table>											CMDD	Start Condition Detection	Clear Conditions (CMDD = 0)					Set Conditions (CMDD = 1)							<ul style="list-style-type: none"> • When transfer start instruction is executed • When stop condition signal is detected • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied 					<ul style="list-style-type: none"> • When start condition signal is detected 						
CMDD	Start Condition Detection																																				
Clear Conditions (CMDD = 0)					Set Conditions (CMDD = 1)																																
<ul style="list-style-type: none"> • When transfer start instruction is executed • When stop condition signal is detected • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied 					<ul style="list-style-type: none"> • When start condition signal is detected 																																
R/W	<table border="1"> <tr> <td>ACKT</td> <td>Used to generate the $\overline{\text{ACK}}$ signal by software when 8-clock wait mode is selected. Keeps SDA0 (SDA1) low from set instruction (ACKT = 1) execution to the next fallig edge of SCL. Also cleared to 0 upon start of serial interface transfer or when CSIE0 = 0.</td> </tr> </table>											ACKT	Used to generate the $\overline{\text{ACK}}$ signal by software when 8-clock wait mode is selected. Keeps SDA0 (SDA1) low from set instruction (ACKT = 1) execution to the next fallig edge of SCL. Also cleared to 0 upon start of serial interface transfer or when CSIE0 = 0.																								
ACKT	Used to generate the $\overline{\text{ACK}}$ signal by software when 8-clock wait mode is selected. Keeps SDA0 (SDA1) low from set instruction (ACKT = 1) execution to the next fallig edge of SCL. Also cleared to 0 upon start of serial interface transfer or when CSIE0 = 0.																																				

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

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

Figure 18-5. Serial Bus Interface Control Register Format (2/2)

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)
		<ul style="list-style-type: none"> • While executing the transfer start instruction • When CSIE0 = 0 • When RESET input is applied 	<ul style="list-style-type: none"> • When acknowledge signal ($\overline{\text{ACK}}$) 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 ² C 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, set BSYE to 1.

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

(4) Interrupt timing specify register (SINT)

This register sets the bus release interrupt and address mask functions and displays the $\overline{\text{SCK0/SCL}}$ pin level status. SINT is set with a 1-bit or 8-bit memory manipulation instruction. $\overline{\text{RESET}}$ input sets SINT to 00H.

Figure 18-6. Interrupt Timing Specify 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 SCK0 clock cycle. (keeping clock output in high impedance)
	0	1	Setting prohibited
	1	0	Used in I ² C 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 ² C 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 State 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 ^{Note 2}
	0	Used in I ² C bus mode. Make output level of SCL pin low unless serial transfer is being performed.
	1	Used in I ² C 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 I²C mode, set CLC to 0.

Figure 18-6. Interrupt Timing Specify 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 Cause Selection ^{Note 1}
	0	CSIF0 is set to 1 upon termination of serial interface channel 0 transfer
	1	CSIF0 is set to 1 upon stop condition detection or termination of serial interface channel 0 transfer

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

Notes 1. When using wake-up function in the I²C mode, set SIC to 0.

2. When CSIE0 = 0, CLD becomes 0.

Remark SVA : Slave address register
 CSIF0 : Interrupt request flag corresponding to INTCSI0
 CSIE0 : Bit 7 of serial operating mode register 0 (CSIM0)

18.4 Serial Interface Channel 0 Operations

The following four operating modes are available to the serial interface channel 0.

- Operation stop mode
- 3-wire serial I/O mode
- 2-wire serial I/O mode
- I²C (Inter IC) bus mode

18.4.1 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/ $\overline{\text{SCK0}}$ /SCL pins can be used as general input/output ports.

(1) Register setting

The operation stop mode is set with the serial operating mode register 0 (CSIM0).

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM0 to 00H.

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W

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

18.4.2 3-wire serial I/O mode operation

The 3-wire serial I/O mode is valid for connection of peripheral I/O units and display controllers which incorporate a conventional synchronous clocked serial interface as is the case with the 75X/XL, 78K, and 17K Series.

Communication is carried out with three lines of serial clock ($\overline{\text{SCK0}}$), serial output (SO0), and serial input (SI0).

(1) Register setting

The 3-wire serial I/O mode is set with the serial operating mode register 0 (CSIM0) and serial bus interface control register (SBIC).

(a) Serial operating mode register 0 (CSIM0)

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM0 to 00H.

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W ^{Note 1}

R/W	CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection
	0	x	Input Clock to $\overline{\text{SCK0}}$ pin from off-chip
	1	0	8-bit timer register 2 (TM2) output
	1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W	CSIM04	CSIM03	CSIM02	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	x	0	Note 2	Note 2	1	x	0	0	0	1	3-wire serial I/O mode	MSB	SI0 ^{Note 2} (Input)	SO0 (CMOS output)	$\overline{\text{SCK0}}$ (CMOS input/output)
			1									LSB				
	1	1	2-wire serial I/O mode (See 18.4.3 2-wire serial I/O mode operation.) or I ² C bus mode (See 18.4.4 I ² C 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 detecting start condition (when CMDD = 1) matches the slave address register (SVA) data in I ² C bus mode

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

Remark x : Don't care
 PMxx : Port mode register
 Pxx : Port output latch

(b) Serial bus interface control register (SBIC)

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

$\overline{\text{RESET}}$ input 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, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.
-----	------	---

R/W	CMDT	When CMDT = 1, SO0 latch is cleared to 0. After SO0 latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.
-----	------	---

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

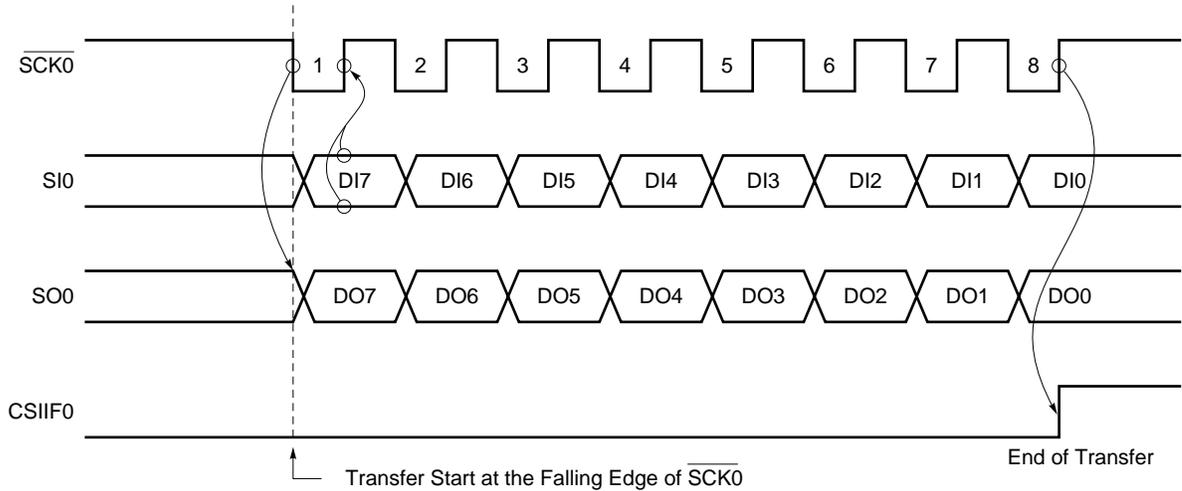
(2) Communication operation

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

Upon termination of 8-bit transfer, SIO0 operation stops automatically and the interrupt request flag (CSIF0) is set.

Figure 18-7. 3-Wire Serial I/O Mode Timings



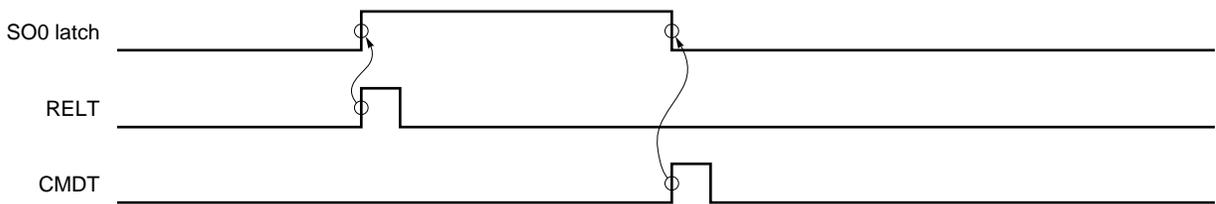
The SO0 pin is a CMOS output pin and outputs current SO0 latch statuses. Thus, the SO0 pin output status can be manipulated by setting the bit 0 (RELT) and bit 1 (CMDT) of the serial bus interface control register (SBIC). However, do not carry out this manipulation during serial transfer.

Control the $\overline{\text{SCK0}}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **18.4.7 $\overline{\text{SCK0}}$ /SCL/P27 pin output manipulation**).

(3) Other signals

Figure 18-8 shows RELT and CMDT operations.

Figure 18-8. RELT and CMDT Operations

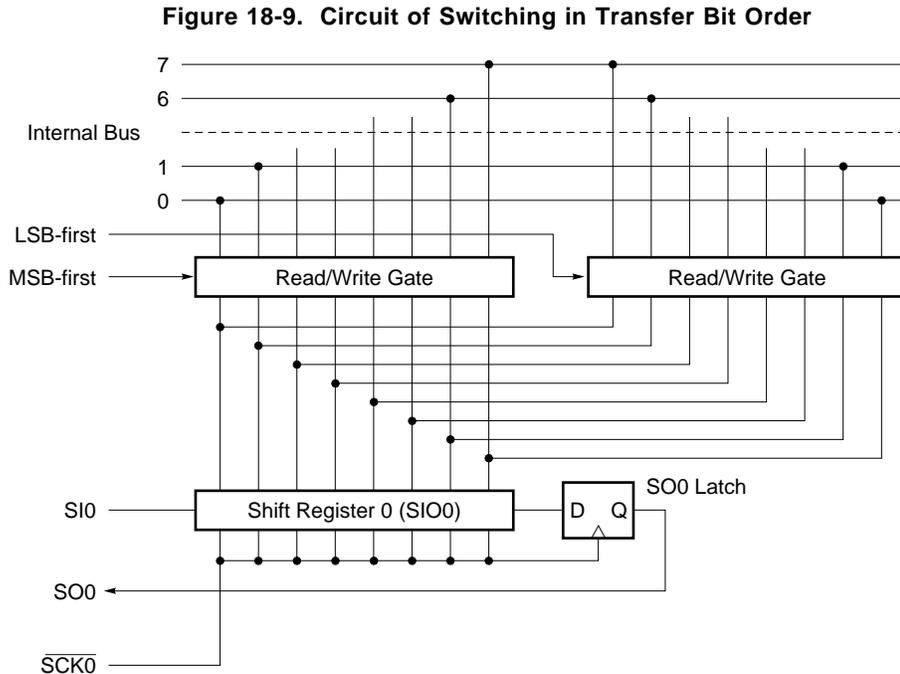


(4) MSB/LSB switching as the start bit

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

Figure 18-9 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).



Start bit switching is realized by switching the bit order for data write to SIO0. The SIO0 shift order remains unchanged.

Thus, MSB-first and LSB-first must be switched before writing data to the shift register.

(5) Transfer start

Serial transfer is started by setting transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied.

- Serial interface channel 0 operation control bit (CSIE0) = 1.
- Internal serial clock is stopped or $\overline{\text{SCK0}}$ is a high level after 8-bit serial transfer.

Caution If CSIE0 is set to “1” after data write to SIO0, transfer does not start.

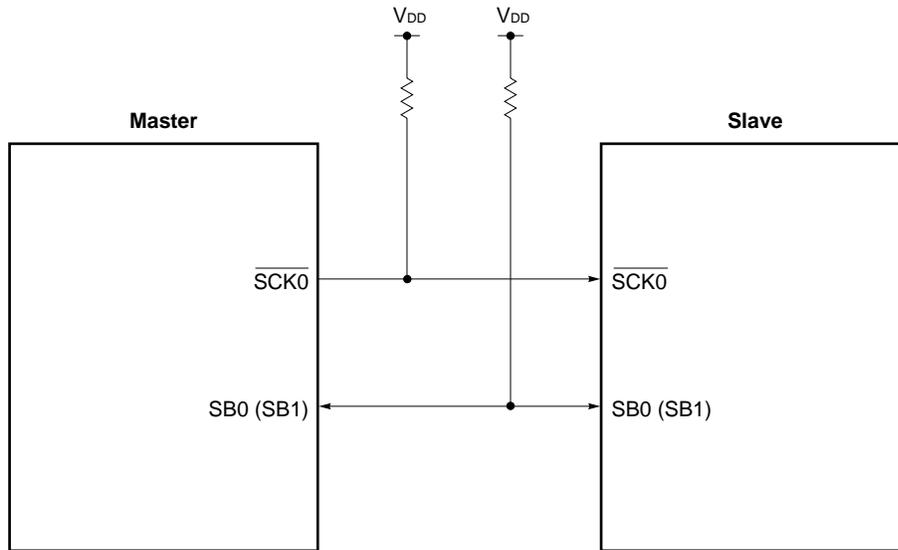
Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (CSIF0) is set.

18.4.3 2-wire serial I/O mode operation

The 2-wire serial I/O mode can cope with any communication format by program.

Communication is basically carried out with two lines of serial clock ($\overline{\text{SCK0}}$) and serial data input/output (SB0 or SB1).

Figure 18-10. 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 00H.

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W ^{Note 1}

R/W	CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection
	0	x	Input Clock to SCK0 pin from off-chip
	1	0	8-bit timer register 2 (TM2) output
	1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W	CSIM04	CSIM03	CSIM02	PM25	PM25	PM26	PM26	PM27	PM27	Operation Mode	Start Bit	SI0/SB0/SDA0 /P25 Pin Function	SO0/SB1/SDA1 /P26 Pin Function	SCK0/SCL/P27 Pin Function	
	0	x	3-wire Serial I/O mode (See 18.4.2 3-wire serial I/O mode operation.)												
	1	1	0	Note 2	Note 2	0	0	0	1	2-wire serial I/O mode or I ² C bus mode	MSB	P25 (CMOS input/output)	SB1/SDA1 (N-ch open-drain input/output)	SCK0/SCL (N-ch open-drain input/output)	
			1	0	Note 2	Note 2	x	x	0			1	SB0/SDA0 (N-ch open-drain input/output)		P26 (CMOS input/output)

R/W	WUP	Wake-up Function Control ^{Note 3}
	0	Interrupt request signal generation with each serial transfer in any mode
	1	Interrupt request signal generation when the address received after detecting start condition (when CMDD = 1) matches the slave address register (SVA) data in I ² C bus mode

R	COI	Slave Address Comparison Result Flag ^{Note 4}
	0	Slave address register (SVA) not equal to serial I/O shift register 0 (SIO0) data
	1	Slave address register (SVA) equal to serial I/O shift register 0 (SIO0) 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. Set WUP to 0 when the 2-wire serial I/O mode is selected.
 4. When CSIE0 = 0, COI becomes 0.

Remark x : Don't care
 PMxxx : Port mode register
 Pxx : Port output latch

(b) Serial bus interface control register (SBIC)

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

$\overline{\text{RESET}}$ input 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, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.
-----	------	--

R/W	CMDT	When CMDT = 1, SO0 latch is cleared to 0. After SO0 latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.
-----	------	--

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

(c) Interrupt timing specify register (SINT)

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

$\overline{\text{RESET}}$ input 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	SIC	INTCSI0 Interrupt Factor Selection
	0	CSIIF0 is set upon termination of serial interface channel 0 transfer
	1	CSIIF0 is set upon bus release detection or termination of serial interface channel 0 transfer

R	CLD	$\overline{\text{SCK0}}$ Pin Level ^{Note 2}
	0	Low level
	1	High level

- Notes**
1. Bit 6 (CLD) is a read-only bit.
 2. When CSIE0 = 0, CLD becomes 0.

Caution Set bits 0 to 3 to 0 when the 2-wire serial I/O mode is selected.

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

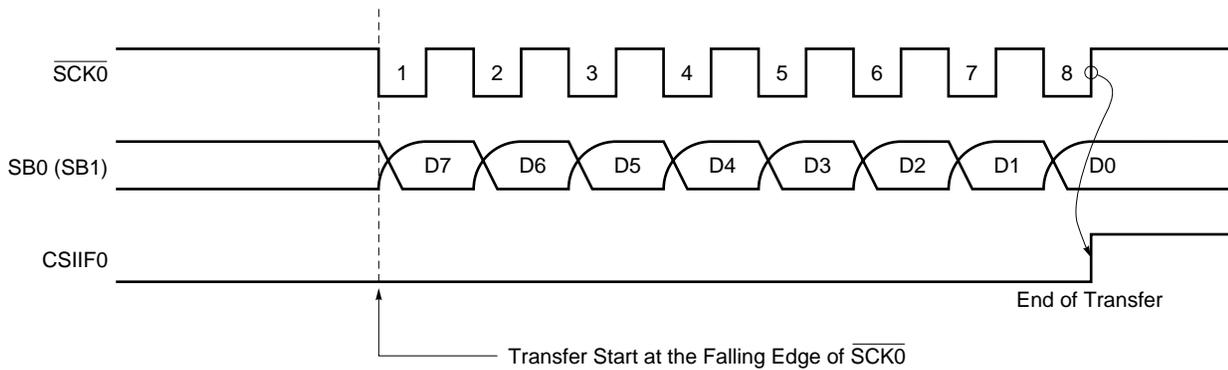
(2) Communication operation

The 2-wire serial I/O mode is used for data transmission/reception in 8-bit units. Data transmission/reception is carried out bit-wise in synchronization with the serial clock.

Shift operation of the serial I/O shift register 0 (SIO0) is carried out in synchronization with the falling edge of the serial clock ($\overline{\text{SCK0}}$). The transmit data is held in the SO0 latch and is output from the SB0/SDA0/P25 (or SB1/SDA1/P26) pin on an MSB-first basis. The receive data input from the SB0 (or SB1) pin is latched into SIO0 at the rising edge of $\overline{\text{SCK0}}$.

Upon termination of 8-bit transfer, the SIO0 operation stops automatically and the interrupt request flag (CSIF0) is set.

Figure 18-11. 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 set N-ch open-drain output to high-impedance state for data reception, write FFH to SIO0 in advance.

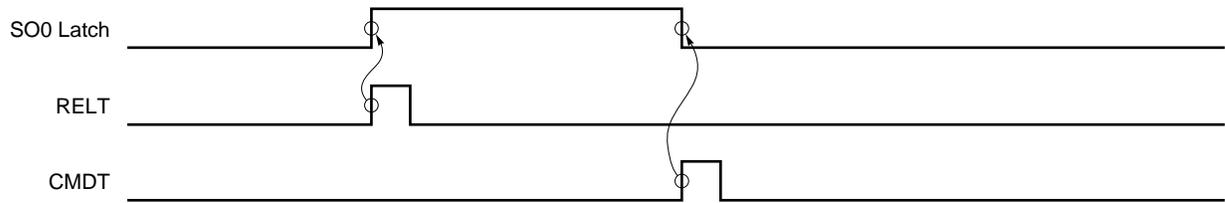
The SB0 (or SB1) pin generates the SO0 latch status and thus the SB0 (or SB1) pin output status can be manipulated by setting the bit 0 (RELT) and bit 1 (CMDT) of the serial bus interface control register (SBIC). However, do not carry out this manipulation during serial transfer.

Control the $\overline{\text{SCK0}}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **18.4.7 $\overline{\text{SCK0/SCL/P27}}$ pin output manipulation**).

(3) Other signals

Figure 18-12 shows RELT and CMDT operations.

Figure 18-12. RELT and CMDT Operations

**(4) Transfer start**

Serial transfer is started by setting transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied.

- Serial interface channel 0 operation control bit (CSIE0) = 1
- Internal serial clock is stopped or $\overline{\text{SCK0}}$ is at high level after 8-bit serial transfer.

Cautions 1. If CSIE0 is set to “1” after data write to SIO0, transfer does not start.

2. Because the N-ch open-drain output must be set to 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 (CSIF0) is set.

(5) Error detection

In the 2-wire serial I/O mode, the serial bus SB0 (SB1) status being transmitted is fetched into the destination device, that is, the serial I/O shift register 0 (SIO0). Thus, transmit error can be detected in the following 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.

18.4.4 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 microcontrollers and peripheral devices, using this configuration results in reduction of the required number of port pins and on-board wires.

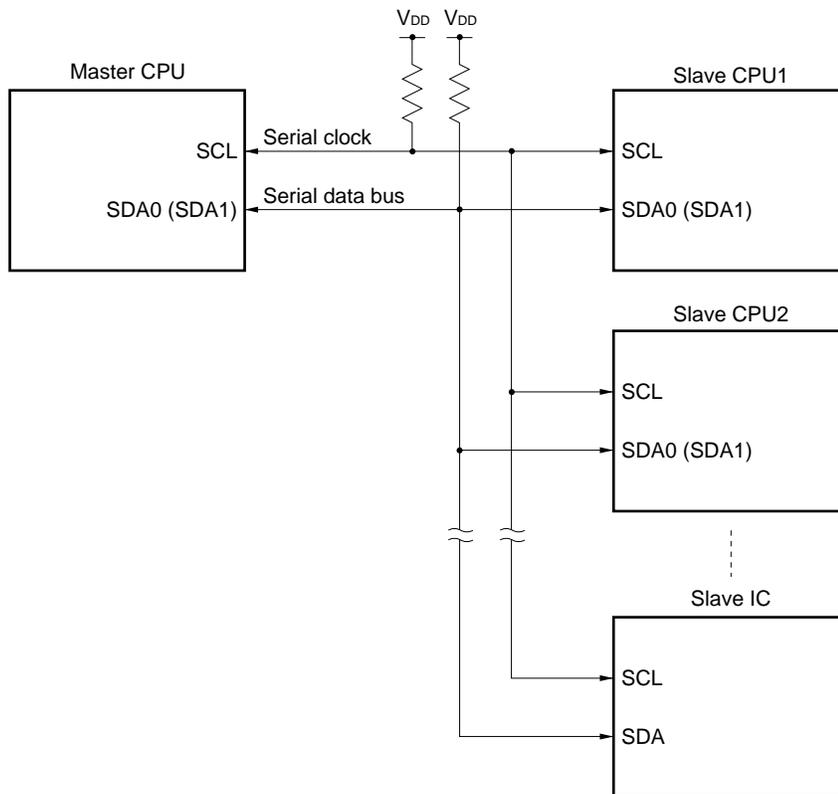
In the I²C bus specification, the master sends start condition, data, and stop condition signals to slave devices through the serial data bus, while slave devices automatically detect and distinguish the type of signals due to the signal detection function incorporated as hardware. This simplifies the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

An example of a serial bus configuration is shown in Figure 18-13. 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 open-drain buffers 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 18-4.

Figure 18-13. Example of Serial Bus Configuration Using I²C Bus



(1) I²C bus mode functions

In the I²C 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 I²C bus and communicate with it by sending in advance the address data corresponding to the destination device.

(c) Wake-up function

When address data is sent from the master device, slave devices compare it with the value registered in their internal slave address registers. If the values in one of the slave devices match, the slave device internally generates an interrupt request signal to terminate the current processing and communicates with the master device (an interrupt request generates also when a stop condition is detected). Therefore, CPUs other than the selected slave device on the I²C bus can operate independently 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

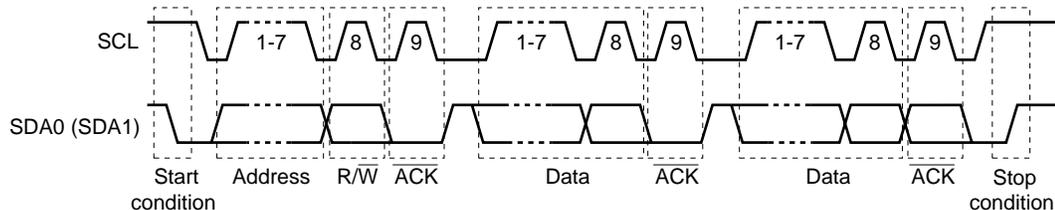
When a slave device is preparing for data transmission or reception and requires more waiting time, the slave device outputs a wait signal on the bus to inform the master device of the wait status.

(2) I²C bus definition

This section describes the format of serial data communications and functions of the signals used in the I²C 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 I²C bus, are shown in Figure 18-14.

Figure 18-14. I²C Bus Serial Data Transfer Timing

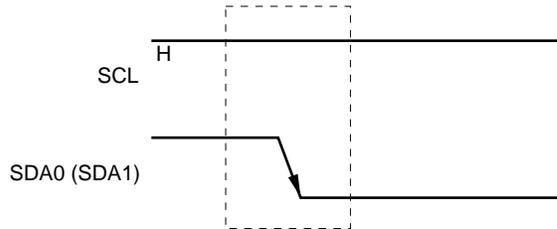


The start condition, slave address, and stop condition signals are output by the master. The acknowledge signal ($\overline{\text{ACK}}$) is output by either the master or the slave device (normally by the device which has received the 8-bit data that was sent). A serial clock (SCL) is continuously supplied 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 **18.4.5 Cautions on use of I²C bus mode**, for details of the start condition output. The start condition signal is detected by hardware incorporated in slave devices.

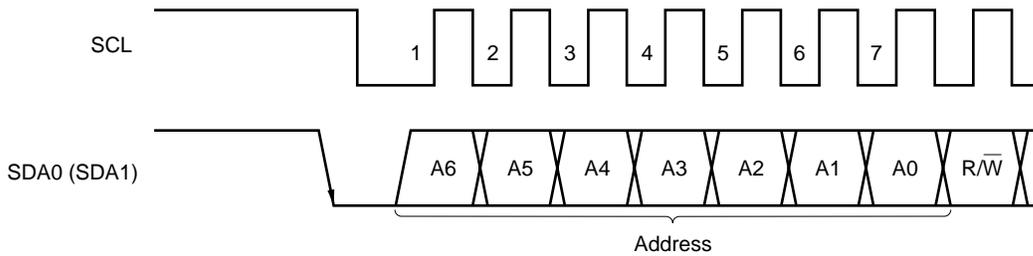
Figure 18-15. Start Condition



(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.

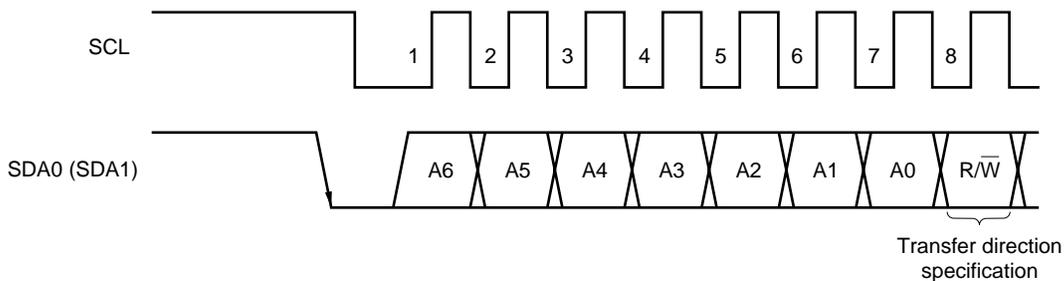
Figure 18-16. Address



(c) Transfer direction specification

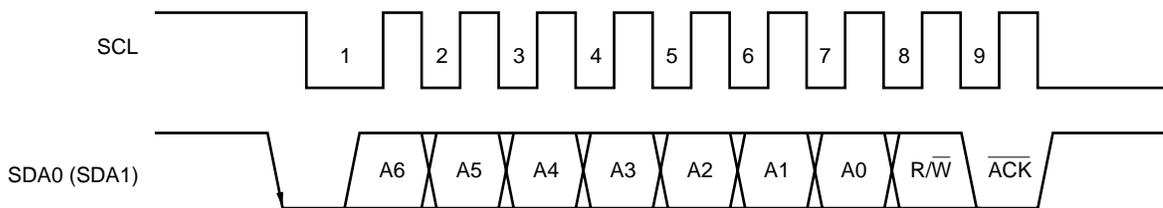
The 1 bit that follows the 7-bit address data will be sent from the master device, and it is defined as the transfer direction specification bit. If this bit is 0, it is the master device which will send data to the slave. If it is 1, it is the slave device which will send data to the master.

Figure 18-17. Transfer Direction Specification



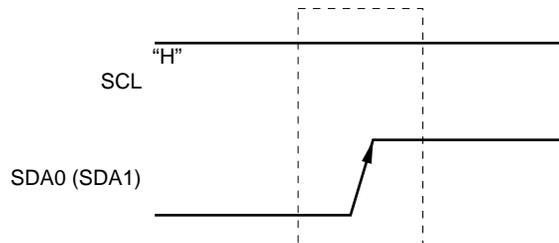
(d) Acknowledge signal ($\overline{\text{ACK}}$)

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 transferred 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 18-18. 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 18-19. Stop Condition

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

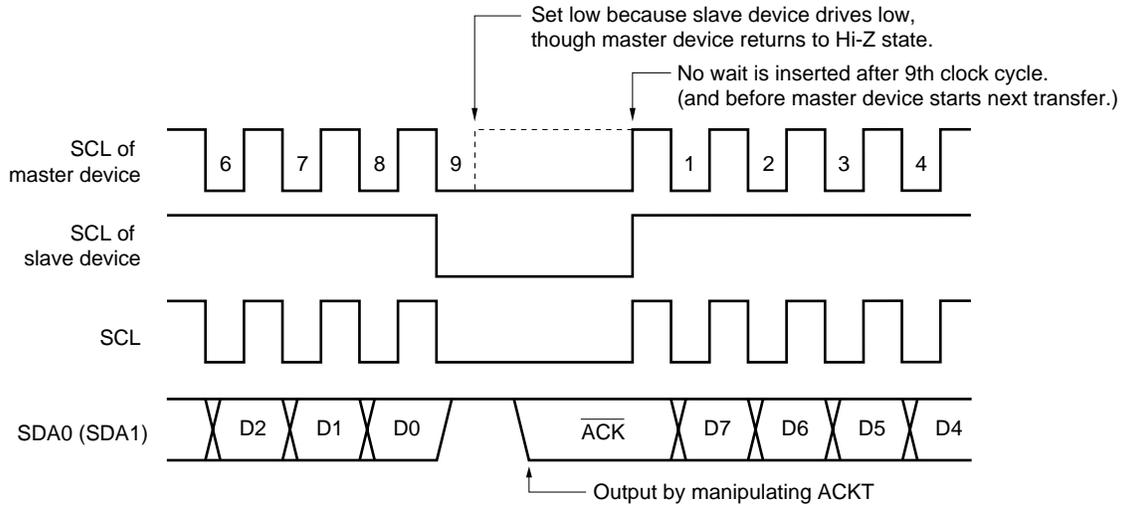
The wait signal is output by a slave device to inform the master device that the slave device is in wait state due to preparing for transmitting or receiving data.

During the wait state, the slave device continues to output the wait signal by keeping the SCL pin low. When the wait state is released, the master device can start the next transfer.

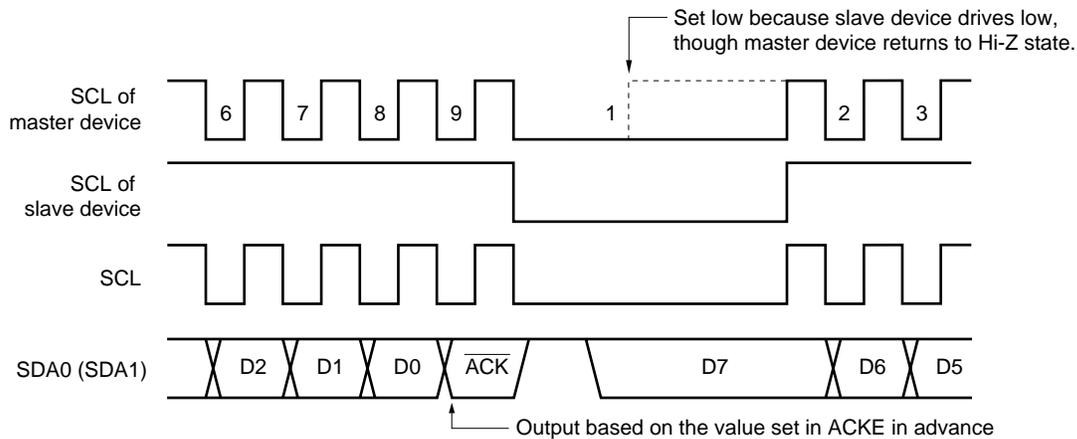
For the releasing operation of slave devices, refer to **18.4.5 Cautions on use of I²C bus mode**.

Figure 18-20. Wait Signal

(a) Wait of 8 Clock Cycles



(b) Wait of 9 Clock Cycles



(3) Register setting

The I²C bus 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 by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets 00H.

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W ^{Note 1}

R/W	CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection								
	0	x	Input clock from off-chip to SCL pin								
	1	0	8-bit timer register 2 (TM2) output ^{Note 2}								
	1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)								

R/W	CSIM04	CSIM03	CSIM02	PM25	P25	PM26	P26	PM27	P27	Operation mode	Start bit	SI0/SB0/SDA0/P25 pin function	SO0/SB1/SDA1/P26 pin function	SCK0/SCL/P27 pin function
	0	x	3-wire serial I/O mode (see 18.4.2 3-wire serial I/O mode operation)											
	1	1	0	x ^{Note 3}	x ^{Note 3}	0	0	0	1	2-wire serial I/O or I ² C bus mode	MSB	P25 (CMOS I/O)	SB1/SDA1 N-ch open-drain I/O	SCK0/SCL N-ch open-drain I/O
	1	1	1	0	0	x	x ^{Note 3}	0	1	2-wire serial I/O or I ² C bus mode	MSB	SB0/SDA0 N-ch open-drain I/O	P26 (CMOS I/O)	SCK0/SCL N-ch open-drain I/O

R/W	WUP	Wake-up Function Control ^{Note 4}												
	0	Interrupt request signal generation with each serial transfer in any mode												
	1	In I ² C 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 (SVA).												

R	COI	Slave Address Comparison Result Flag ^{Note 5}												
	0	Slave address register (SVA) not equal to data in serial I/O shift register 0 (SIO0)												
	1	Slave address register (SVA) equal to data in serial I/O shift register 0 (SIO0)												

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. In the I²C bus mode, the clock frequency is 1/16 of the clock frequency output by TO2.

3. Can be used freely as a port.

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

5. When CSIE0 = 0, COI is 0.

Remark x : Don't care
 PMxxx : Port mode register
 Pxx : Port output latch

(b) Serial bus interface control register (SBIC)

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

$\overline{\text{RESET}}$ 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	RELT	Use for stop condition output. When RELT = 1, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.									
R/W	CMDT	Use for start condition output. When CMDT = 1, SO0 latch is cleared to 0. After clearing SO0 latch, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.									
R	RELD	Stop Condition Detection									
	0	Clear Conditions <ul style="list-style-type: none"> • When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied 									
	1	Setting Condition <ul style="list-style-type: none"> • When stop condition is detected 									
R	CMDD	Start Condition Detection									
	0	Clear Conditions <ul style="list-style-type: none"> • When transfer start instruction is executed • When stop condition is detected • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied 									
	1	Setting Condition <ul style="list-style-type: none"> • When start condition is detected 									
R/W	ACKT	SDA0 (SDA1) is set to low after the Set instruction execution (ACKT = 1) before the next SCL falling edge. Used for generating an $\overline{\text{ACK}}$ 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 ACKT = 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 <ul style="list-style-type: none"> • When transfer start instruction is executed • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied 									
	1	Set Conditions <ul style="list-style-type: none"> • When acknowledge signal is detected at the rising edge of SCL clock after completion of transfer 									
R/W	BSYE	Control of N-ch Open-Drain Output for Transmission in I ² C Bus Mode ^{Note 5}									
	Note 4										
	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, set BSYE to 1.

Remark CSIE0 : Bit 7 of 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.

$\overline{\text{RESET}}$ 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 $\overline{\text{SCK0}}$ clock cycle (clock output is high impedance).
	0	1	Setting prohibited
	1	0	Used in I ² C 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.)
R/W	1	1	Used in I ² C 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.)
	WREL	Wait release control	
R/W	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.	
	CLC	Clock level control	
R/W	0	Used in I ² C bus mode. In cases other than serial transfer, SCL pin output is driven low.	
	1	Used in I ² C 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	SCL pin level ^{Note 4}	
	0	Low level	
	1	High level	

- Notes**
1. Bit 6 (CLD) is read-only.
 2. When the I²C bus mode is used, set 1 and 0, or 1 and 1 in WAT0 and WAT1, respectively.
 3. When using the wake-up function in I²C mode, 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 serial operating mode register 0 (CSIM0)

(4) Various signals

A list of signals in the I²C bus mode is given in Table 18-4.

Table 18-4. Signals in I²C 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 (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 : See Note 2 below.
	Affected flag(s) : CSIF0. Also see Note 3 below.
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 : See Note 2 below.
	Affected flag(s) : CSIF0. Also see Note 3 below.
Transfer direction (R/ \bar{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 : See Note 2 below.
	Affected flag(s) : CSIF0. Also see Note 3 below.
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 : See Note 2 below.
	Affected flag(s) : CSIF0. Also see Note 3 below.

- Notes**
1. The level of the serial clock can be controlled by CLC of the interrupt timing specify 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, CSIF0 is set at the rising edge of the 8th clock cycle of SCL. If the 9-clock wait is selected when WUP = 0, CSIF0 is set at the rising edge of the 9th clock cycle of SCL. If WUP = 1, CSIF0 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 pins 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

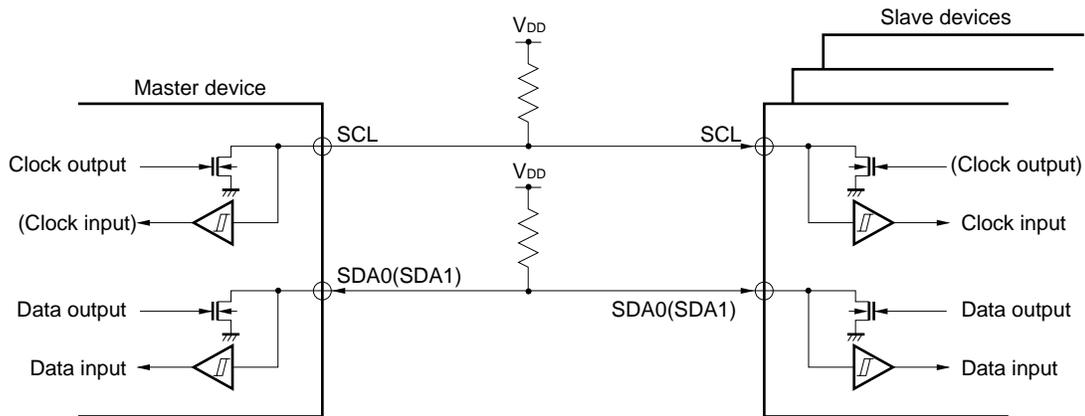
(b) SDA0 (SDA1)

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 I²C bus.

Figure 18-21. Pin Configuration



Caution Because the N-ch open-drain output must be in the high-impedance state during data reception, set bit 7 (BSYE) of the serial bus interface control register (SBIC) to 1 before writing FFH to the serial I/O shift register 0 (SIO0). However, do not write FFH to the SIO0 during data reception when using the wake-up function (when bit 5 (WUP) of the serial operation mode register 0 (CSIM0)). N-ch open-drain always enters the high-impedance state even if FFH is not written to SIO0.

(6) Address match detection method

In the I²C 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 function specification (WUP) bit is then 1, CSIF0 is set (also when a stop condition is detected).

When using the wake-up function, set SIC to 1.

Caution Whether a slave is selected or not is detected by the matching of the data (address) received after start condition.

The matching detection interrupt request (INTCSI0) of the address generated in the WUP = 1 state is normally used for detecting the matching of the data. Therefore, detection of whether a slave is selected or not using slave address must be performed in the WUP = 1 state.

(7) Error detection

In the I²C bus mode, transmission error can be detected 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) 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 I²C 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/\overline{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 18-22 and 18-23.

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 SIO0 in synchronization with the rising edge of SCL.

Figure 18-22. Example of Communication from Master to Slave (with 9-Clock Wait Selected for Both Master and Slave) (1/3)

(a) Start condition - address

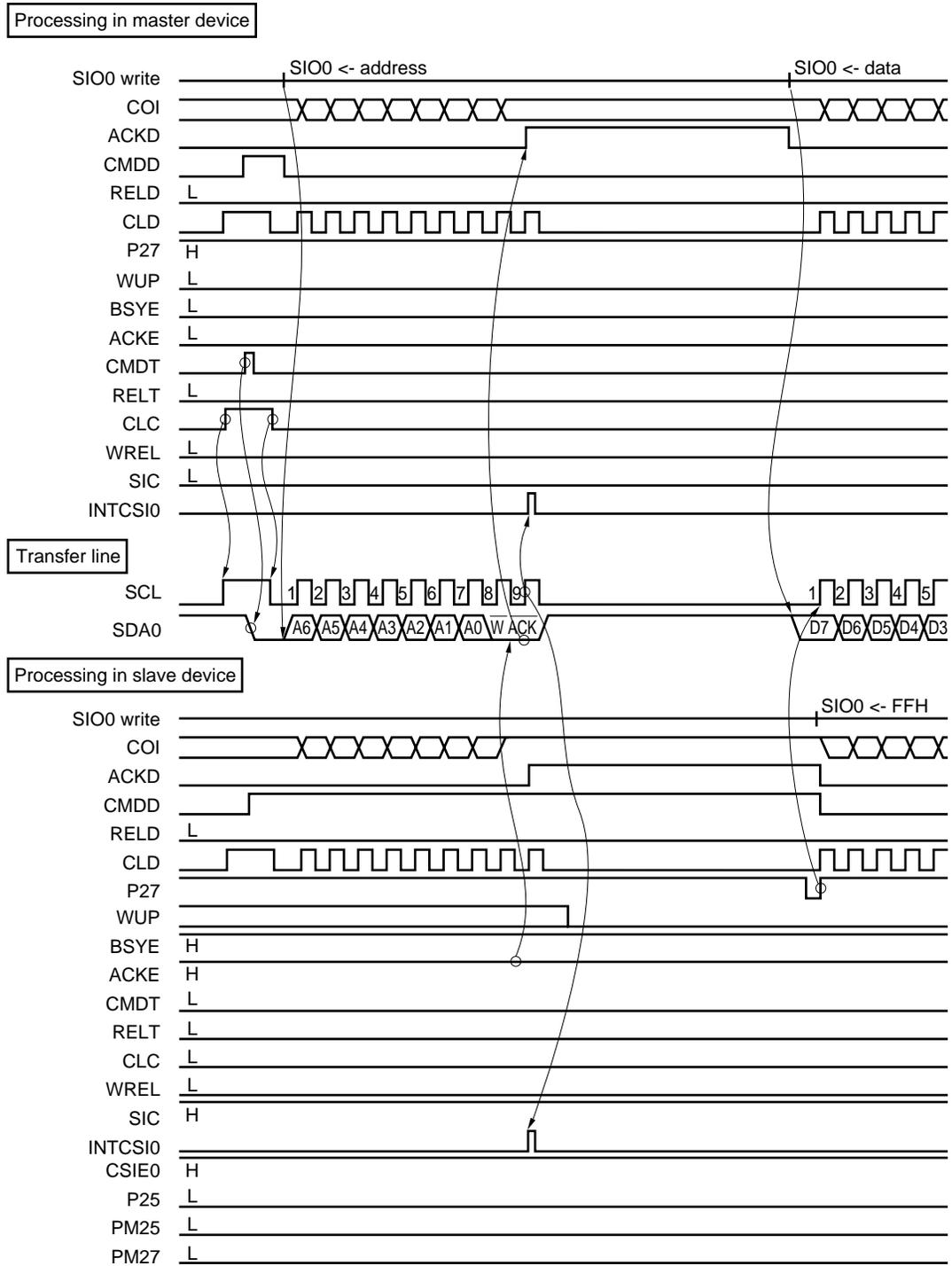


Figure 18-22. Example of Communication from Master to Slave (with 9-Clock Wait Selected for Both Master and Slave) (2/3)

(b) Data

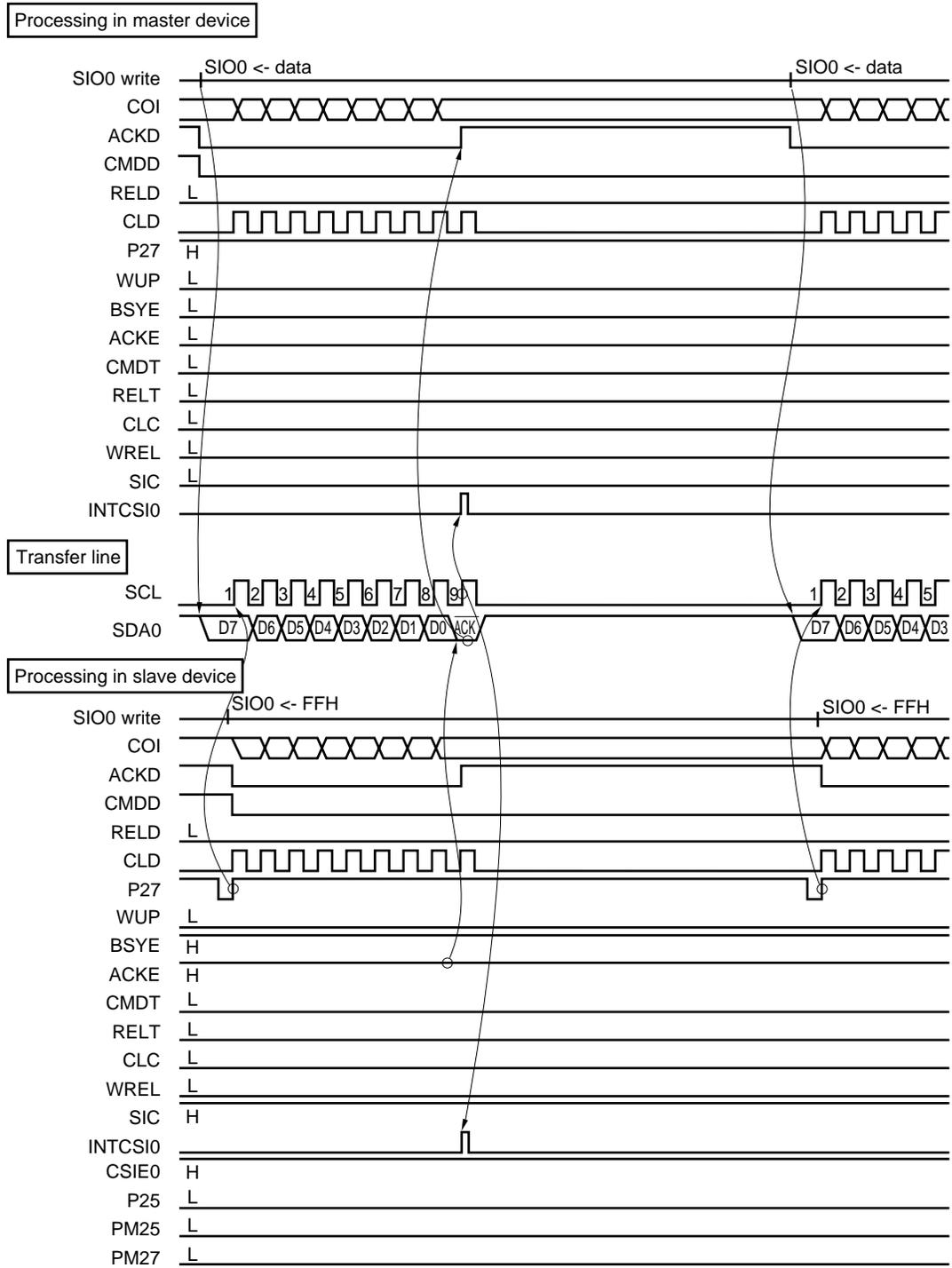


Figure 18-22. Example of Communication from Master to Slave (with 9-Clock Wait Selected for Both Master and Slave) (3/3)

(c) Stop condition

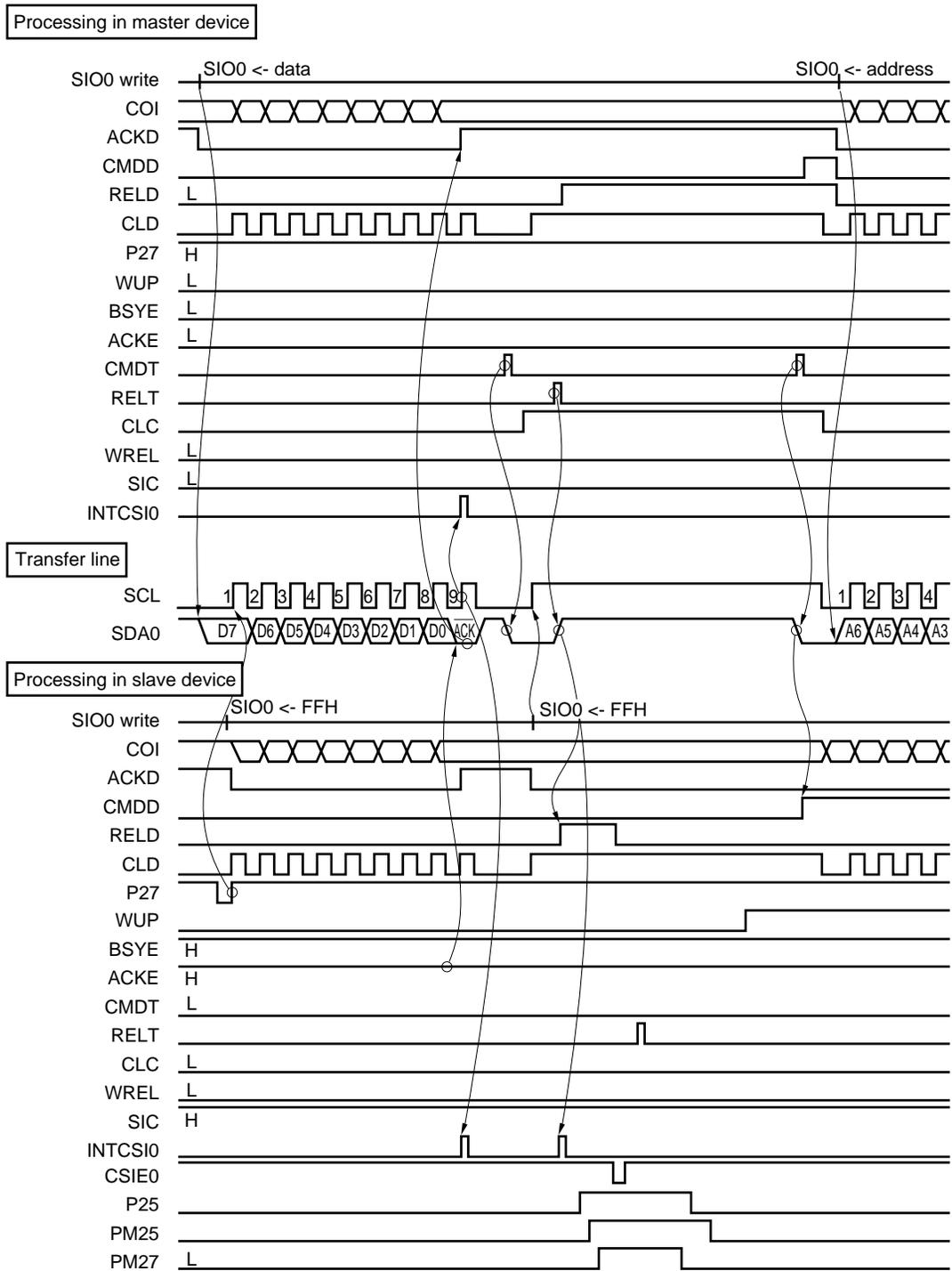


Figure 18-23. Example of Communication from Slave to Master (with 9-Clock Wait Selected for Both Master and Slave) (1/3)

(a) Start condition - address

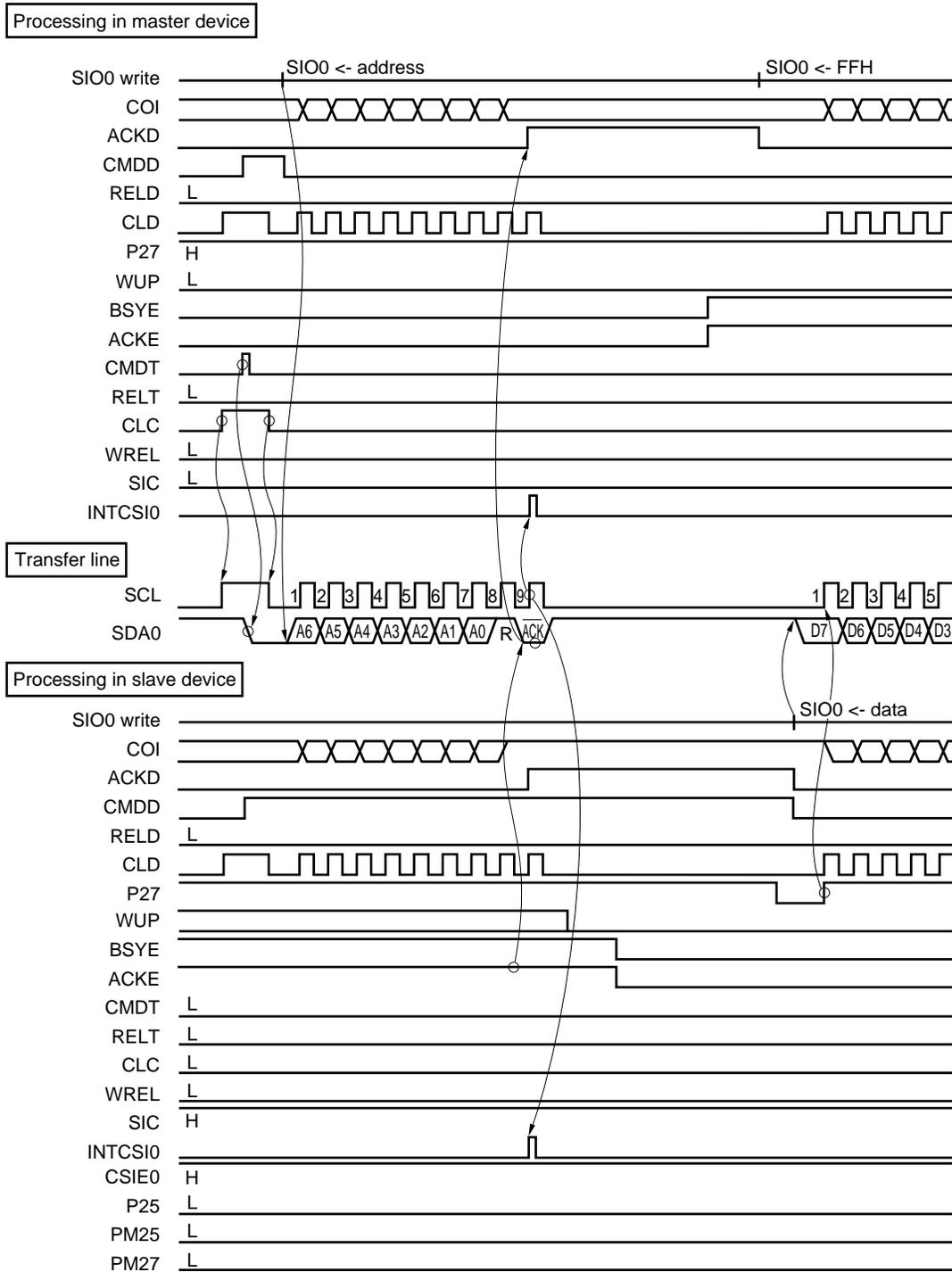


Figure 18-23. Example of Communication from Slave to Master (with 9-Clock Wait Selected for Both Master and Slave) (2/3)

(b) Data

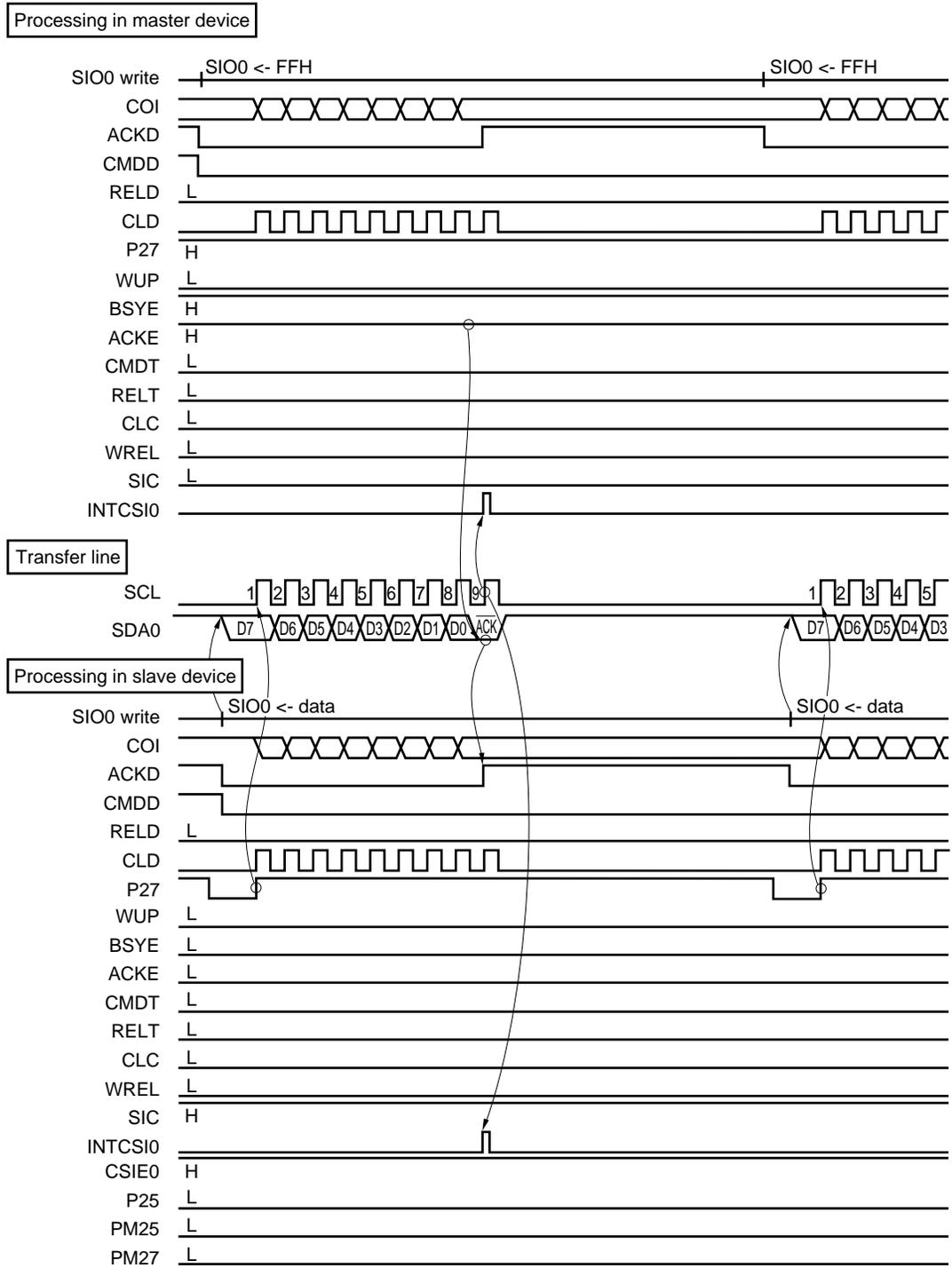
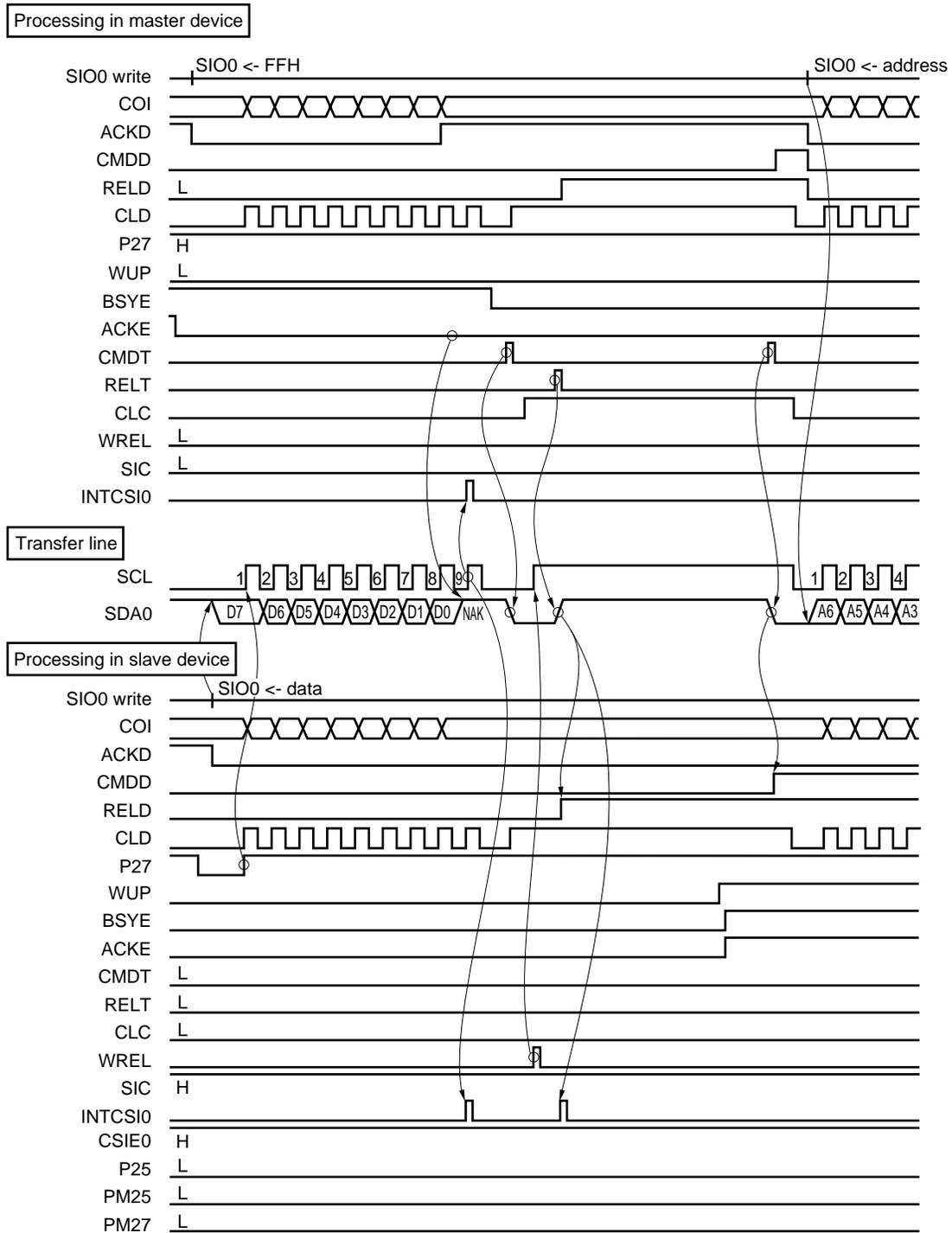


Figure 18-23. Example of Communication from Slave to Master (with 9-Clock Wait Selected for Both Master and Slave) (3/3)

(c) Stop condition



(9) Start of transfer

A serial transfer is started by setting transfer data in the 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. **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 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.**

However, do not write FFH to the SIO0 during data reception when using the wake-up function (setting the bit 5 (WUP) of the serial operation mode register 0 (CSIM0)). N-ch open-drain output always enters the high-impedance state even when 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 (CSIF0) is set.

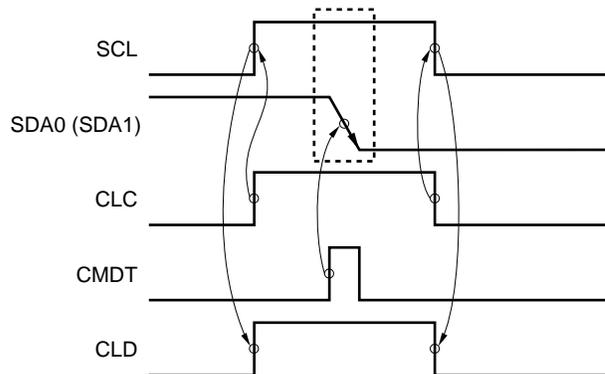
18.4.5 Cautions on use of I²C bus mode

(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 the bit 3 (CLC) of the interrupt timing specify 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 18-24. Start Condition Output



(2) Slave wait release (slave transmission)

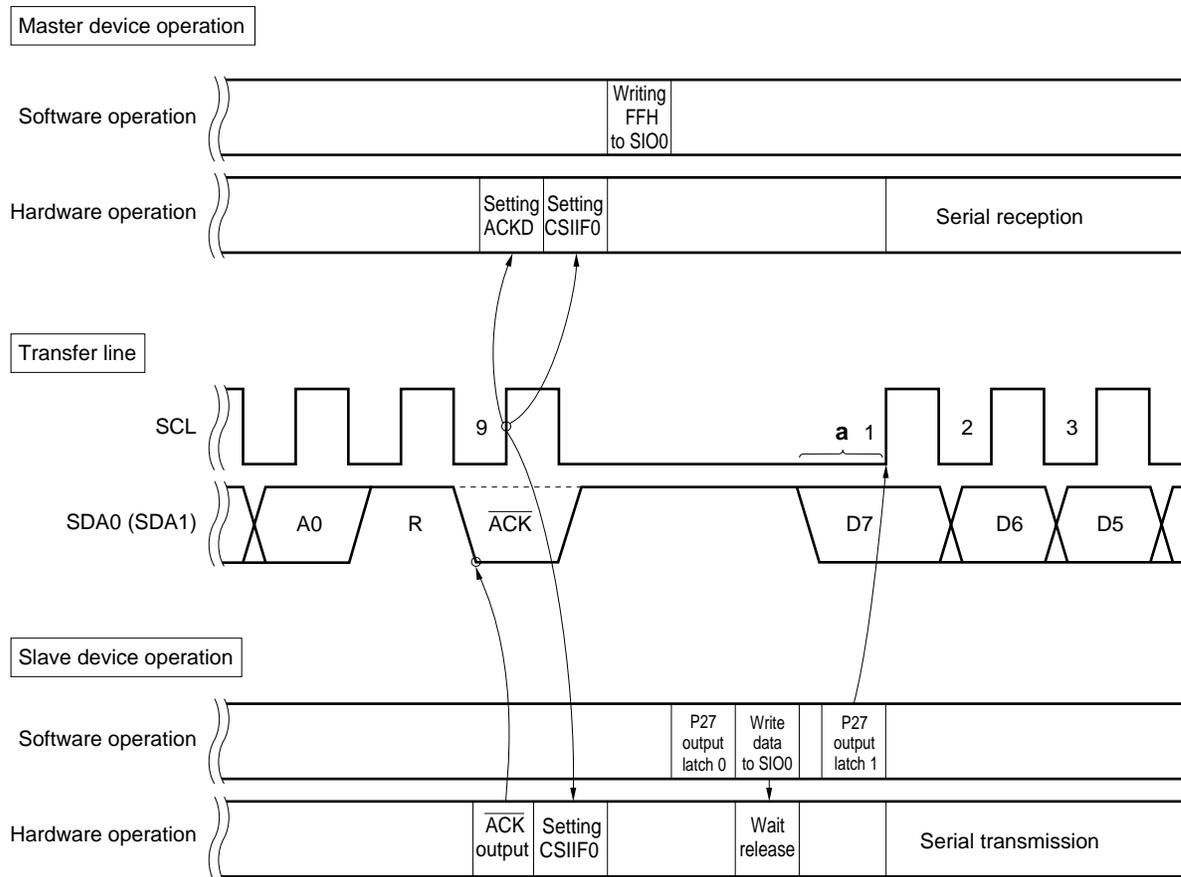
The wait status of a slave is released by setting the WREL flag, which is bit 2 of the interrupt timing specify register (SINT), or by executing a serial I/O shift register 0 (SIO0) write instruction.

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, manipulate the P27 output latch through the program as shown in Figure 18-25 to transmit data correctly. At this time, control the low-level width ("a" in Figure 18-25) 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.

For these timings, see Figure 18-23.

Figure 18-25. Slave Wait Release (Transmission)



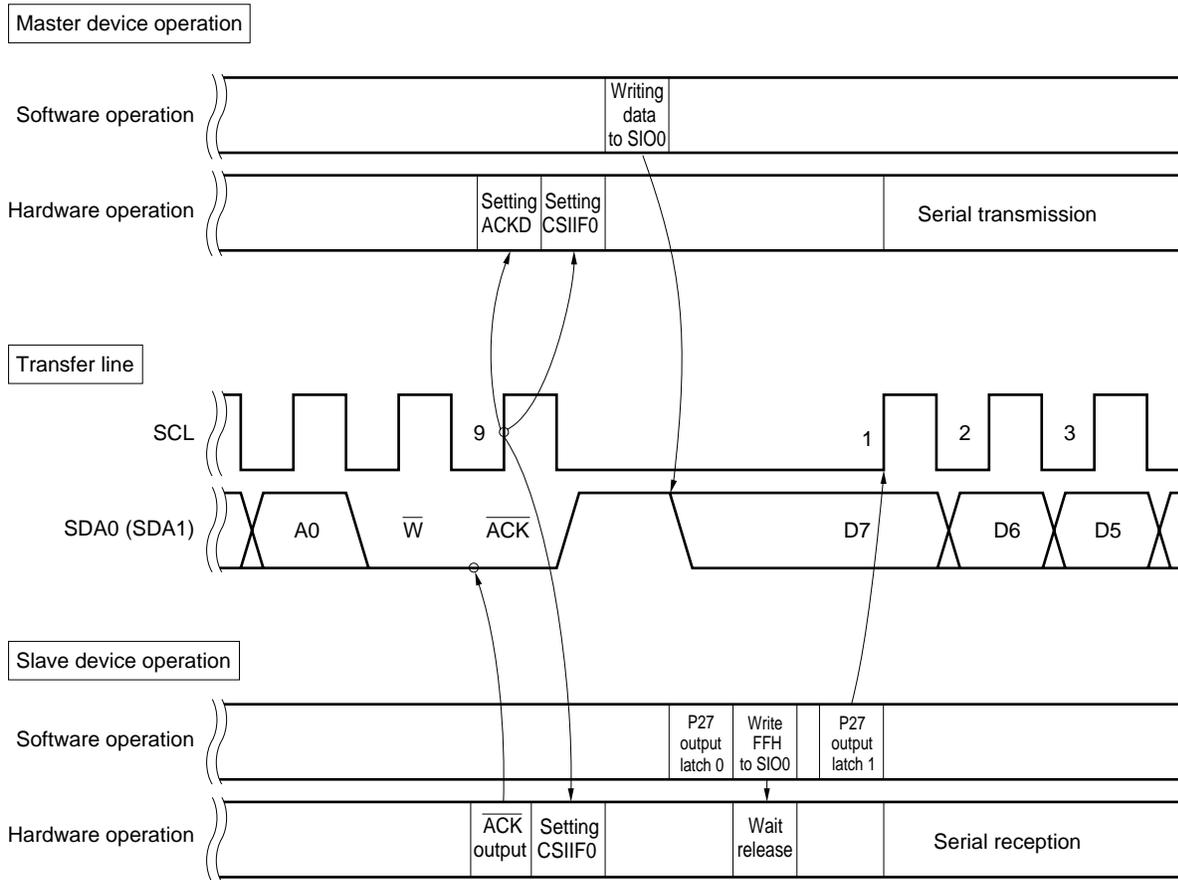
(3) Slave wait release (slave reception)

The wait status of a slave is released by setting the WREL flag, which is bit 2 of the interrupt timing specify register (SINT), or by executing a serial I/O shift register 0 (SIO0) write instruction.

When a slave receives data, if the SCL line immediately enters a high-impedance state due to a write to SIO0, the slave may not receive the first bit of the data sent from the master. This is because SIO0 cannot start operation if the SCL line is in a high-impedance state during execution of a write instruction to SIO0 (until the next instruction execution is started). Therefore, manipulate the P27 output latch through the program as shown in Figure 18-26 to receive data correctly.

For these timings, see **Figure 18-22**.

Figure 18-26. Slave Wait Release (Reception)



(4) 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.

18.4.6 Restrictions in I²C bus mode

The following restrictions apply to the μ PD78078Y Subseries.

- **Restrictions when used as slave device in I²C bus mode**

Applicable models μ PD78076Y, 78078Y, 78P078Y, and IE-78078-R-EM

Description When the wake-up function is executed (by setting the WUP flag (bit 5 of the serial operation mode register 0 (CSIM0)) in the serial transfer status^{Note}, data between the other slaves and master will be judged as an address. If this data happens to coincide with the slave address of the μ PD78078Y Subseries, the μ PD78078Y Subseries will initiate communication, destroying the communication data.

Note The serial transfer status is the status in which the interrupt request flag (CSIF0) is set because of the end of serial transfer after the serial I/O shift register 0 (SIO0) has been written.

Preventive measure This restriction can be avoided by modifying the program. Before executing the wake-up function, execute the following program that releases serial transfer status. To execute the wake-up function, do not execute an instruction that writes SIO0. Even if such an instruction is not executed, data can be received when the wake-up function is executed.

This program releases the serial transfer status. To release the serial transfer status, the serial interface channel 0 must be set once in the operation stop status (by clearing the CSIE0 flag (bit 7 of the serial operation mode register (CSIM0) to 0). However, if the serial interface channel 0 is set in the operation stop status in the I²C bus mode, the SCL pin output a high level and the SDA0 (SDA1) pin outputs a low level, affecting communication of the I²C bus. Therefore, this program places the SCL and SDA0 (SDA1) pins in the high-impedance state to prevent the I²C bus from being affected.

In the example below, SDA0 (/P25) is used as a serial data input/output pin. When SDA1 (/P26) is used as the serial data input/output pin, take P2.5 and PM2.5 in the program below as P2.6 and PM2.6, respectively

For the timing of each signal when this program is executed, refer to **Figure 18-22**.

- **Example of program releasing serial transfer status**

```
SET1 P2.5    ; <1>
SET1 PM2.5   ; <2>
SET1 PM2.7   ; <3>
CLR1 CSIE0   ; <4>
SET1 CSIE0   ; <5>
SET1 RELT    ; <6>
CLR1 PM2.7   ; <7>
CLR1 P2.5    ; <8>
CLR1 PM2.5   ; <9>
```

<1> Prevents the SDA0 pin from outputting a low level when the I²C bus mode is restored by the instruction in <5>. The output of the SDA0 pin goes into a high-impedance state.

<2> Sets the P25 (/SDA0) pin in the input mode to prevent the SDA0 line from being affected when the port mode is set by the instruction in <4>. The P25 pin is set in the input mode when the instruction in <2> is executed.

<3> Sets the P27 (/SCL) pin in the input mode to prevent the SCL line from being affected when the port mode is set by the instruction in <4>. The P27 pin is set in the input mode when the instruction in <3> is executed.

<4> Changes the mode from the I²C bus mode to port mode.

<5> Restores the mode from the port mode to the I²C bus mode.

<6> Prevents the instruction in <8> from causing the SDA0 pin to output a low level.

<7> Sets the P27 pin in the output mode because the P27 pin must be in the output mode in the I²C bus mode.

<8> Clears the output latch of the P25 pin to 0 because the output latch of the P25 pin must be cleared to 0 in the I²C bus mode.

<9> Sets the P25 pin in the output mode because the P25 pin must be in the output mode in the I²C bus mode.

Remark RELT: bit 0 of serial bus interface control register (SBIC)

18.4.7 $\overline{\text{SCK0/SCL/P27}}$ pin output manipulation

The $\overline{\text{SCK0/SCL/P27}}$ pin enables static output by manipulating software in addition to normal serial clock output.

The value of serial clocks can be set by software (SI0/SB0/SDA0 and SO0/SB1/SDA1 pins are controlled with the RELT and CMDT bits of serial bus interface control register (SBIC)).

The $\overline{\text{SCK0/SCL/P27}}$ pin output should be manipulated as described below.

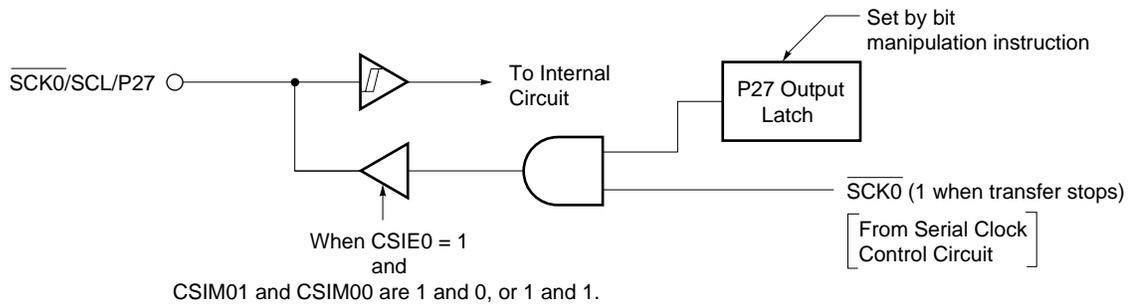
(1) In 3-wire serial I/O mode and 2-wire serial I/O mode

The $\overline{\text{SCK0/SCL/P27}}$ pin output level is manipulated by the P27 output latch.

<1> Set serial operating mode register 0 (CSIM0) ($\overline{\text{SCK0}}$ pin is set in the output mode and serial operation is enabled). While serial transfer is suspended, $\overline{\text{SCK0}}$ is set to 1.

<2> Manipulate the content of the P27 output latch by executing the bit manipulation instruction.

Figure 18-27. $\overline{\text{SCK0/SCL/P27}}$ Pin Configuration



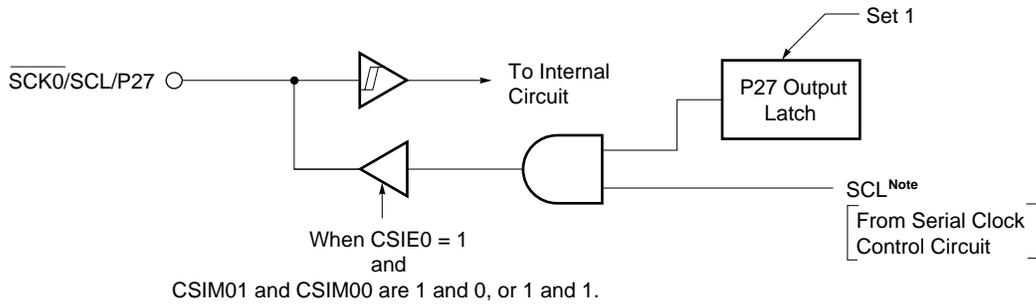
(2) In I²C bus mode

The $\overline{\text{SCK0/SCL/P27}}$ pin output level is manipulated by the CLC bit of interrupt timing specify register (SINT).

<1> Set serial operating mode register 0 (CSIM0) (SCL pin is set in the output mode and serial operation is enabled). Set 1 to the P27 output latch. While serial transfer is suspended, SCL is set to 0.

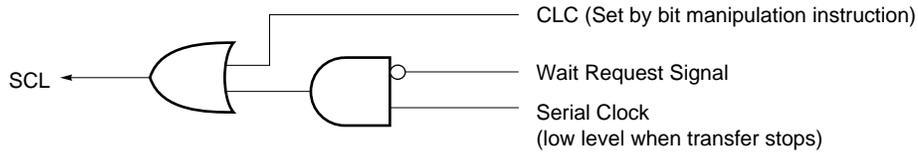
<2> Manipulate the content of the CLC bit of SINT by executing the bit manipulation instruction.

Figure 18-28. $\overline{\text{SCK0/SCL/P27}}$ Pin Configuration



Note The level of SCL signal follows the contents of logic circuit shown in Figure 18-29.

Figure 18-29. Logic Circuit of SCL Signal



- Remarks**
1. This figure shows the relationship of each signal, and does not show the internal circuit.
 2. CLC: Bit 3 of interrupt timing specify register (SINT)

CHAPTER 19 SERIAL INTERFACE CHANNEL 1

19.1 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 the same functions as those of the 3-wire serial I/O mode with automatic transmit/receive function added.

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.

19.2 Serial Interface Channel 1 Configuration

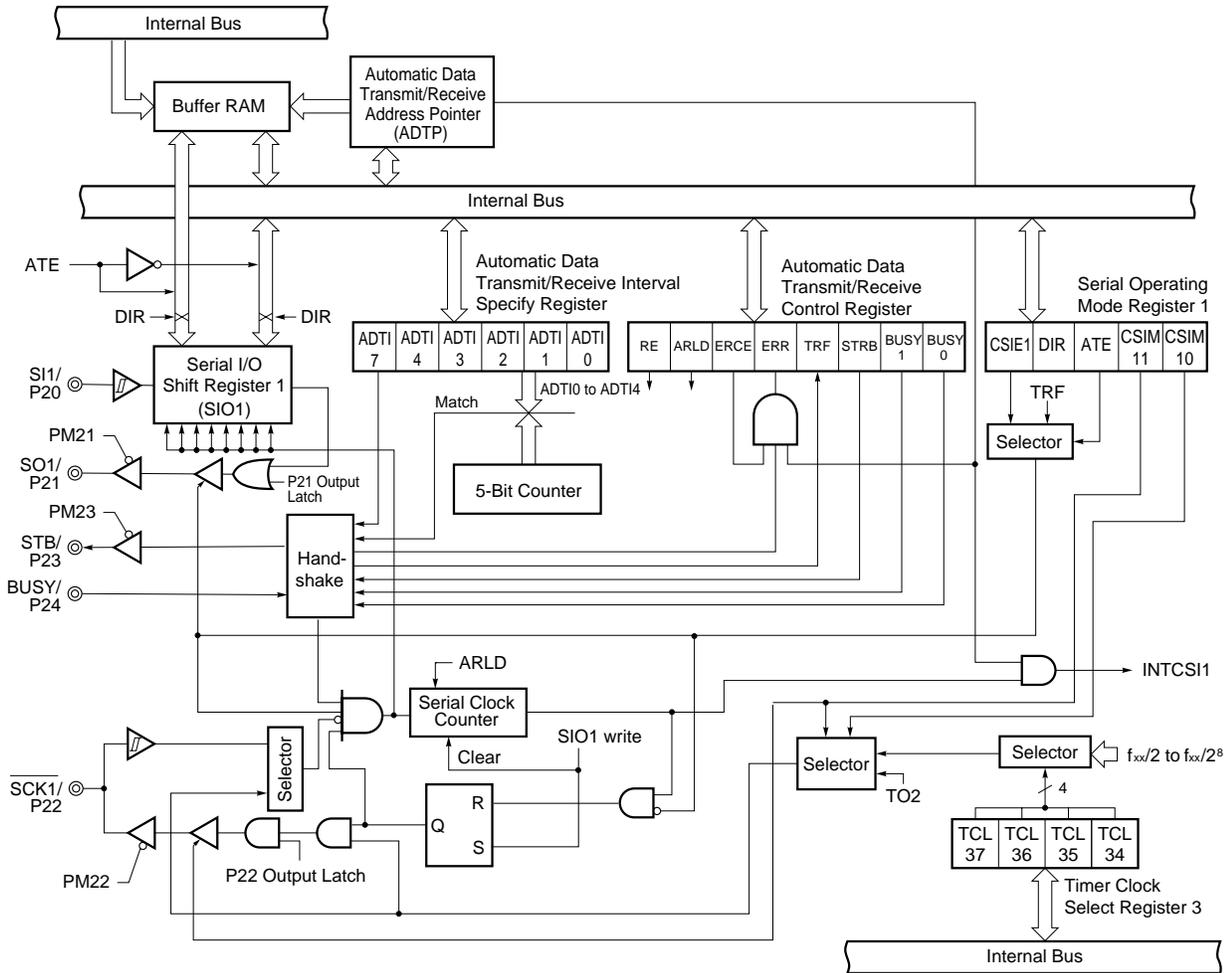
Serial interface channel 1 consists of the following hardware.

Table 19-1. 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 specify register (ADTI) Port mode register 2 (PM2) ^{Note}

Note Refer to **Figures 6-5 and 6-7 Block Diagram of P20, P21, P23 to 26** and **Figures 6-6 and 6-8 Block Diagram of P22 and P27**.

Figure 19-1. Serial Interface Channel 1 Block Diagram



(1) Serial I/O shift register 1 (SIO1)

This is an 8-bit register to carry out parallel/serial conversion and to carry out serial transmission/reception (shift operation) in synchronization with the serial clock.

SIO1 is set with an 8-bit memory manipulation instruction.

When value in bit 7 (CSIE1) of serial operating mode register 1 (CSIM1) is 1, writing data to SIO1 starts serial operation.

In transmission, data written to SIO1 is output to the serial output (SO1). In reception, data is read from the serial input (SI1) to SIO1.

$\overline{\text{RESET}}$ input makes SIO1 undefined.

Caution Do not write data to SIO1 while the automatic transmit/receive function is activated.

(2) Automatic data transmit/receive address pointer (ADTP)

This register stores the value (number of transmit data bytes – 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.

$\overline{\text{RESET}}$ 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.

19.3 Serial Interface Channel 1 Control Registers

The following four types of registers are used to control serial interface channel 1.

- Timer clock select register 3 (TCL3)
- Serial operating mode register 1 (CSIM1)
- Automatic data transmit/receive control register (ADTC)
- Automatic data transmit/receive interval specify register (ADTI)

(1) Timer clock select register 3 (TCL3)

This register sets the serial clock of serial interface channel 1.

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

$\overline{\text{RESET}}$ input sets TCL3 to 88H.

Remark Besides setting the serial clock of serial interface channel 1, TCL3 sets the serial clock of serial interface channel 0.

Figure 19-2. 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

				Serial Interface Channel 1 Serial Clock Selection		
TCL37	TCL36	TCL35	TCL34		MCS = 1	MCS = 0
0	1	1	0	$f_{xx}/2$	Setting prohibited	$f_x/2^2$ (1.25 MHz)
0	1	1	1	$f_{xx}/2^2$	$f_x/2^2$ (1.25 MHz)	$f_x/2^3$ (625 kHz)
1	0	0	0	$f_{xx}/2^3$	$f_x/2^3$ (625 kHz)	$f_x/2^4$ (313 kHz)
1	0	0	1	$f_{xx}/2^4$	$f_x/2^4$ (313 kHz)	$f_x/2^5$ (156 kHz)
1	0	1	0	$f_{xx}/2^5$	$f_x/2^5$ (156 kHz)	$f_x/2^6$ (78.1 kHz)
1	0	1	1	$f_{xx}/2^6$	$f_x/2^6$ (78.1 kHz)	$f_x/2^7$ (39.1 kHz)
1	1	0	0	$f_{xx}/2^7$	$f_x/2^7$ (39.1 kHz)	$f_x/2^8$ (19.5 kHz)
1	1	0	1	$f_{xx}/2^8$	$f_x/2^8$ (19.5 kHz)	$f_x/2^9$ (9.8 kHz)
Other than above				Setting prohibited		

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

- Remarks**
1. f_{xx} : Main system clock frequency (f_x or $f_x/2$)
 2. f_x : Main system clock oscillation frequency
 3. MCS : Bit 0 of oscillation mode selection register (OSMS)
 4. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

(2) Serial operating mode register 1 (CSIM1)

This register sets serial interface channel 1 serial clock, operating mode, operation enable/stop, and automatic transmit/receive operation enable/stop.

CSIM1 is set with a 1-bit or 8-bit memory manipulation instruction.

RESET input sets CSIM1 to 00H.

Figure 19-3. Serial Operation 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	CSIM10	FF68H	00H	R/W

CSIM11	CSIM10	Serial Interface Channel 1 Clock Selection
0	x	Clock externally input to $\overline{\text{SCK1}}$ pin ^{Note 1}
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 4 to 7 of timer clock select register 3 (TCL3)

ATE	Serial Interface Channel 1 Operating Mode Selection
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

DIR	Start Bit	SI1 Pin Function	SO1 Pin Function
0	MSB	SI1/P20 (Input)	SO1 (CMOS output)
1	LSB		

CSIE	CSIM	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	x	Note 2	Operation stop	Clear	P20 (CMOS input/output)	P21 (CMOS input/output)	P22 (CMOS input/output)					
1	0	Note 3	Note 3	0	0	1	x	Operation enable	Count operation	SI1 ^{Note 3} (input)	SO1 (CMOS output)	$\overline{\text{SCK1}}$ (Input)
	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. Can be used as P20 (CMOS input/output) when only transmitter is used (set bit 7 (RE) of ADTC to 0).

Remark x : Don't care
 PMxx: Port mode register
 Pxx : Port output latch

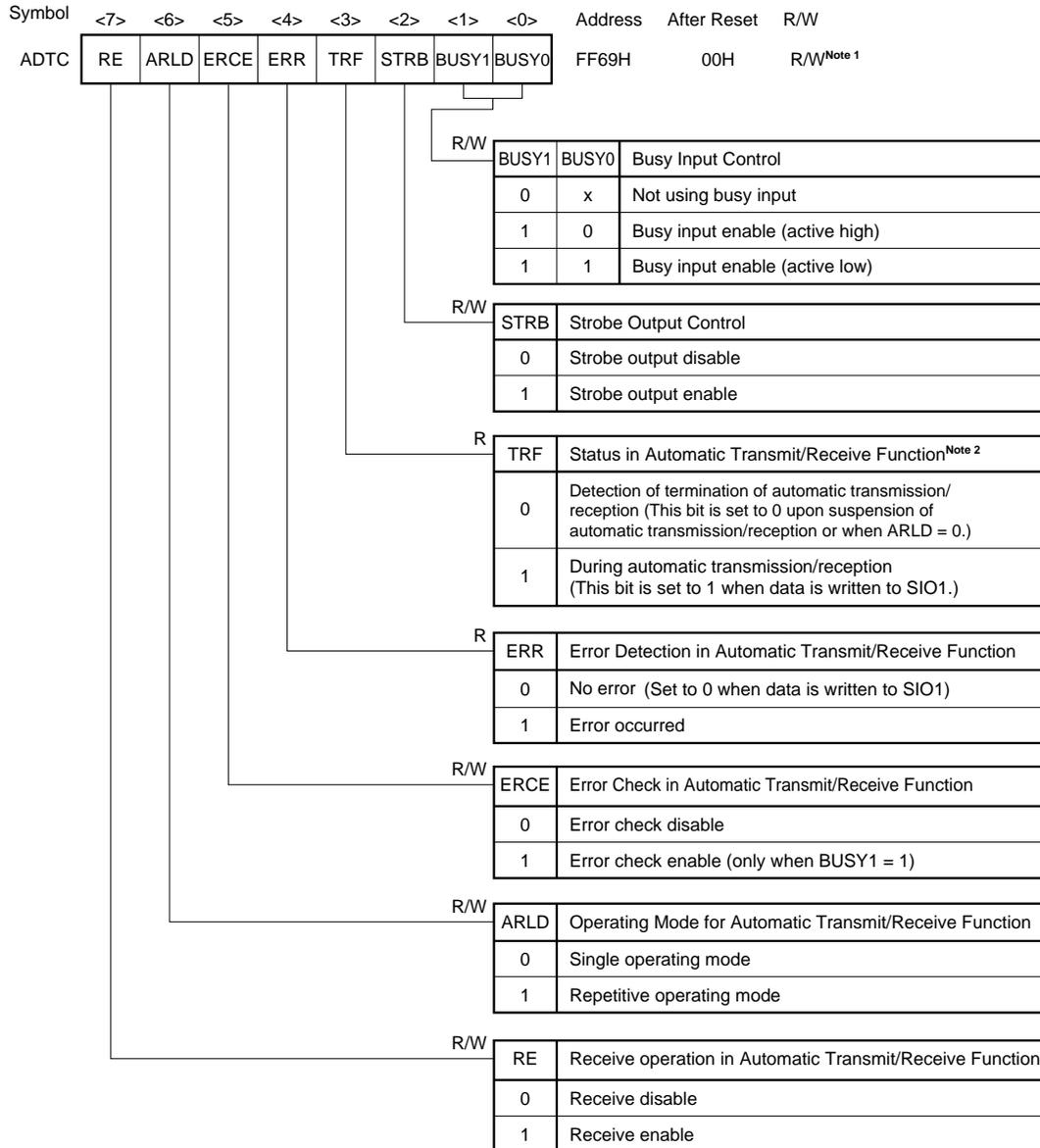
(3) Automatic data transmit/receive control register (ADTC)

This register sets automatic receive enable/disable, the operating mode, strobe output enable/disable, busy input enable/disable, and error check enable/disable, and displays automatic transmit/receive execution and error detection.

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

RESET input sets ADTC to 00H.

Figure 19-4. Automatic Data Transmit/Receive Control Register Format



Notes 1. Bits 3 and 4 (TRF and ERR) are read-only bits.

2. The termination of automatic transmission/reception should be judged by using TRF, not CSIF1 (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 x: Don't care

(4) Automatic data transmit/receive interval specify register (ADTI)

This register sets the automatic data transmit/receive function data transfer interval.

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

$\overline{\text{RESET}}$ input sets ADTI to 00H.

Figure 19-5. Automatic Data Transmit/Receive Interval Specify Register Format (1/4)

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} = 5.0-MHz Operation)	
					Minimum ^{Note 2}	Maximum ^{Note 2}
0	0	0	0	0	18.4 μs + 0.5/f _{sck}	20.0 μs + 1.5/f _{sck}
0	0	0	0	1	31.2 μs + 0.5/f _{sck}	32.8 μs + 1.5/f _{sck}
0	0	0	1	0	44.0 μs + 0.5/f _{sck}	45.6 μs + 1.5/f _{sck}
0	0	0	1	1	56.8 μs + 0.5/f _{sck}	58.4 μs + 1.5/f _{sck}
0	0	1	0	0	69.6 μs + 0.5/f _{sck}	71.2 μs + 1.5/f _{sck}
0	0	1	0	1	82.4 μs + 0.5/f _{sck}	84.0 μs + 1.5/f _{sck}
0	0	1	1	0	95.2 μs + 0.5/f _{sck}	96.8 μs + 1.5/f _{sck}
0	0	1	1	1	108.0 μs + 0.5/f _{sck}	109.6 μs + 1.5/f _{sck}
0	1	0	0	0	120.8 μs + 0.5/f _{sck}	122.4 μs + 1.5/f _{sck}
0	1	0	0	1	133.6 μs + 0.5/f _{sck}	135.2 μs + 1.5/f _{sck}
0	1	0	1	0	146.4 μs + 0.5/f _{sck}	148.0 μs + 1.5/f _{sck}
0	1	0	1	1	159.2 μs + 0.5/f _{sck}	160.8 μs + 1.5/f _{sck}
0	1	1	0	0	172.0 μs + 0.5/f _{sck}	173.6 μs + 1.5/f _{sck}
0	1	1	0	1	184.8 μs + 0.5/f _{sck}	186.4 μs + 1.5/f _{sck}
0	1	1	1	0	197.6 μs + 0.5/f _{sck}	199.2 μs + 1.5/f _{sck}
0	1	1	1	1	210.4 μs + 0.5/f _{sck}	212.0 μs + 1.5/f _{sck}

Notes 1. The interval 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}.

$$\text{Minimum} = (n + 1) \times \frac{2^6}{f_{xx}} + \frac{28}{f_{xx}} + \frac{0.5}{f_{sck}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^6}{f_{xx}} + \frac{36}{f_{xx}} + \frac{1.5}{f_{sck}}$$

Cautions 1. Do not write to ADTI during operation of the automatic data transmit/receive function.

2. Bits 5 and 6 must be set to 0.

3. When controlling the data transfer interval by automatic transmit/receive using ADTI, busy control (refer to 19.4.3 (4) (a) Busy control option) becomes invalid.

Remark f_{xx} : Main system clock frequency (fx or fx/2)

fx : Main system clock oscillation frequency

f_{sck} : Serial clock frequency

Figure 19-5. Automatic Data Transmit/Receive Interval Specify Register Format (2/4)

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} = 5.0-MHz Operation)	
					Minimum ^{Note}	Maximum ^{Note}
1	0	0	0	0	223.2 μs + 0.5/f _{sck}	224.8 μs + 1.5/f _{sck}
1	0	0	0	1	236.0 μs + 0.5/f _{sck}	237.6 μs + 1.5/f _{sck}
1	0	0	1	0	248.8 μs + 0.5/f _{sck}	250.4 μs + 1.5/f _{sck}
1	0	0	1	1	261.6 μs + 0.5/f _{sck}	263.2 μs + 1.5/f _{sck}
1	0	1	0	0	274.4 μs + 0.5/f _{sck}	276.0 μs + 1.5/f _{sck}
1	0	1	0	1	287.2 μs + 0.5/f _{sck}	288.8 μs + 1.5/f _{sck}
1	0	1	1	0	300.0 μs + 0.5/f _{sck}	301.6 μs + 1.5/f _{sck}
1	0	1	1	1	312.8 μs + 0.5/f _{sck}	314.4 μs + 1.5/f _{sck}
1	1	0	0	0	325.6 μs + 0.5/f _{sck}	327.2 μs + 1.5/f _{sck}
1	1	0	0	1	338.4 μs + 0.5/f _{sck}	340.0 μs + 1.5/f _{sck}
1	1	0	1	0	351.2 μs + 0.5/f _{sck}	352.8 μs + 1.5/f _{sck}
1	1	0	1	1	364.0 μs + 0.5/f _{sck}	365.6 μs + 1.5/f _{sck}
1	1	1	0	0	376.8 μs + 0.5/f _{sck}	378.4 μs + 1.5/f _{sck}
1	1	1	0	1	389.6 μs + 0.5/f _{sck}	391.2 μs + 1.5/f _{sck}
1	1	1	1	0	402.4 μs + 0.5/f _{sck}	404.0 μs + 1.5/f _{sck}
1	1	1	1	1	415.2 μs + 0.5/f _{sck}	416.8 μs + 1.5/f _{sck}

Note The 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 to ADTI during operation of the automatic data transmit/receive function.
 2. Bits 5 and 6 must be set to 0.
 3. When controlling the data transfer interval by automatic transmit/receive using ADTI, busy control (refer to 19.4.3 (4) (a) Busy control option) becomes invalid.

Remark f_{xx} : Main system clock frequency (fx or fx/2)
 fx : Main system clock oscillation frequency
 f_{sck} : Serial clock frequency

Figure 19-5. Automatic Data Transmit/Receive Interval Specify Register Format (3/4)

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} = 2.5-MHz Operation)	
					Minimum ^{Note 2}	Maximum ^{Note 2}
0	0	0	0	0	36.8 μs + 0.5/f _{sck}	40.0 μs + 1.5/f _{sck}
0	0	0	0	1	62.4 μs + 0.5/f _{sck}	65.6 μs + 1.5/f _{sck}
0	0	0	1	0	88.0 μs + 0.5/f _{sck}	91.2 μs + 1.5/f _{sck}
0	0	0	1	1	113.6 μs + 0.5/f _{sck}	116.8 μs + 1.5/f _{sck}
0	0	1	0	0	139.2 μs + 0.5/f _{sck}	142.4 μs + 1.5/f _{sck}
0	0	1	0	1	164.8 μs + 0.5/f _{sck}	168.0 μs + 1.5/f _{sck}
0	0	1	1	0	190.4 μs + 0.5/f _{sck}	193.6 μs + 1.5/f _{sck}
0	0	1	1	1	216.0 μs + 0.5/f _{sck}	219.2 μs + 1.5/f _{sck}
0	1	0	0	0	241.6 μs + 0.5/f _{sck}	244.8 μs + 1.5/f _{sck}
0	1	0	0	1	267.2 μs + 0.5/f _{sck}	270.4 μs + 1.5/f _{sck}
0	1	0	1	0	292.8 μs + 0.5/f _{sck}	296.0 μs + 1.5/f _{sck}
0	1	0	1	1	318.4 μs + 0.5/f _{sck}	321.6 μs + 1.5/f _{sck}
0	1	1	0	0	344.0 μs + 0.5/f _{sck}	347.2 μs + 1.5/f _{sck}
0	1	1	0	1	369.6 μs + 0.5/f _{sck}	372.8 μs + 1.5/f _{sck}
0	1	1	1	0	395.2 μs + 0.5/f _{sck}	398.4 μs + 1.5/f _{sck}
0	1	1	1	1	420.8 μs + 0.5/f _{sck}	424.0 μs + 1.5/f _{sck}

- Notes**
1. The interval is dependent only on CPU processing.
 2. The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than 2/f_{sck}, the minimum interval time is 2/f_{sck}.

$$\text{Minimum} = (n + 1) \times \frac{2^6}{f_{xx}} + \frac{28}{f_{xx}} + \frac{0.5}{f_{sck}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^6}{f_{xx}} + \frac{36}{f_{xx}} + \frac{1.5}{f_{sck}}$$

- Cautions**
1. Do not write to ADTI during operation of the automatic data transmit/receive function.
 2. Bits 5 and 6 must be set to 0.
 3. When controlling the data transfer interval by automatic transmit/receive using ADTI, busy control (refer to 19.4.3 (4) (a) Busy control option) becomes invalid.

Remark f_{xx} : Main system clock frequency (fx or fx/2)
 fx : Main system clock oscillation frequency
 f_{sck} : Serial clock frequency

Figure 19-5. Automatic Data Transmit/Receive Interval Specify Register Format (4/4)

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} = 2.5-MHz Operation)	
					Minimum ^{Note}	Maximum ^{Note}
1	0	0	0	0	446.4 μs + 0.5/f _{sck}	449.6 μs + 1.5/f _{sck}
1	0	0	0	1	472.0 μs + 0.5/f _{sck}	475.2 μs + 1.5/f _{sck}
1	0	0	1	0	497.6 μs + 0.5/f _{sck}	500.8 μs + 1.5/f _{sck}
1	0	0	1	1	523.2 μs + 0.5/f _{sck}	526.4 μs + 1.5/f _{sck}
1	0	1	0	0	548.8 μs + 0.5/f _{sck}	552.0 μs + 1.5/f _{sck}
1	0	1	0	1	574.4 μs + 0.5/f _{sck}	577.6 μs + 1.5/f _{sck}
1	0	1	1	0	600.0 μs + 0.5/f _{sck}	603.2 μs + 1.5/f _{sck}
1	0	1	1	1	625.6 μs + 0.5/f _{sck}	628.8 μs + 1.5/f _{sck}
1	1	0	0	0	651.2 μs + 0.5/f _{sck}	654.4 μs + 1.5/f _{sck}
1	1	0	0	1	676.8 μs + 0.5/f _{sck}	680.0 μs + 1.5/f _{sck}
1	1	0	1	0	702.4 μs + 0.5/f _{sck}	705.6 μs + 1.5/f _{sck}
1	1	0	1	1	728.0 μs + 0.5/f _{sck}	731.2 μs + 1.5/f _{sck}
1	1	1	0	0	753.6 μs + 0.5/f _{sck}	756.8 μs + 1.5/f _{sck}
1	1	1	0	1	779.2 μs + 0.5/f _{sck}	782.4 μs + 1.5/f _{sck}
1	1	1	1	0	804.8 μs + 0.5/f _{sck}	808.0 μs + 1.5/f _{sck}
1	1	1	1	1	830.4 μs + 0.5/f _{sck}	833.6 μs + 1.5/f _{sck}

Note The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than 2/f_{sck}, the minimum interval time is 2/f_{sck}.

$$\text{Minimum} = (n + 1) \times \frac{2^6}{f_{xx}} + \frac{28}{f_{xx}} + \frac{0.5}{f_{sck}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^6}{f_{xx}} + \frac{36}{f_{xx}} + \frac{1.5}{f_{sck}}$$

- Cautions**
1. Do not write to ADTI during operation of the automatic data transmit/receive function.
 2. Bits 5 and 6 must be set to 0.
 3. When controlling the data transfer interval by automatic transmit/receive using ADTI, busy control (refer to 19.4.3 (4) (a) Busy control option) becomes invalid.

Remark f_{xx} : Main system clock frequency (fx or fx/2)
 fx : Main system clock oscillation frequency
 f_{sck} : Serial clock frequency

19.4 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

19.4.1 Operation stop mode

Serial transfer is not carried out in the operation stop mode. Thus, power consumption can be reduced. The serial I/O shift register 1 (SIO1) does not carry out shift operation either, and thus it can be used as an ordinary 8-bit register.

In the operation stop mode, the P20/SI1, P21/SO1, P22/ $\overline{\text{SCK1}}$, P23/STB, and P24/BUSY pins can be used as ordinary input/output ports.

(1) Register setting

The operation stop mode is set with the serial operating mode register 1 (CSIM1).

CSIM1 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM1 to 00H.

Symbol	<7>	6	<5>	4	3	2	1	0	Address	After Reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM11	CSIM10	FF68H	00H	R/W

CSIE	CSIM	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	x	Note 1 x	Operation stop	Clear	P20 (CMOS input/output)	P21 (CMOS input/output)	P22 (CMOS input/output)					
1	0	Note 2 1	Note 2 x	0	0	1	x	Operation enable	Count operation	SI1>Note 2 (Input)	SO1 (CMOS output)	$\overline{\text{SCK1}}$ (Input)
	0					1	$\overline{\text{SCK1}}$ (CMOS output)					

Notes 1. Can be used freely as port function.

2. Can be used as P20 (CMOS input/output) when only transmitter is used (set bit 7 (RE) of the automatic data transmit/receive control register (ADTC) to 0).

Remark x : Don't care
 PMxx : Port mode register
 Pxx : Port output latch

19.4.2 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.

\overline{RESET} input sets CSIM1 to 00H.

Symbol	<7>	6	<5>	4	3	2	1	0	Address	After Reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM11	CSIM10	FF68H	00H	R/W

CSIM11	CSIM10	Serial Interface Channel 1 Clock Selection
0	x	Clock externally input to $\overline{SCK1}$ pin ^{Note}
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 4 to 7 of timer clock select register 3 (TCL3)

ATE	Serial Interface Channel 1 Operating Mode Selection
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

DIR	Start Bit	SI1 Pin Function	SO1 Pin Function
0	MSB	SI1/P20 (Input)	SO1 (CMOS output)
1	LSB		

CSIE	CSIM	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
1	11							Operation stop	Clear	P20 (CMOS input/output)	P21 (CMOS input/output)	P22 (CMOS input/output)
1	0	Note 2 1	Note 2 x	0	0	1	x	Operation enable	Count operation	SI1 ^{Note 2} (Input)	SO1 (CMOS output)	$\overline{SCK1}$ (Input)
	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. Can be used as P20 (CMOS input/output) when only transmitter is used (set bit 7 (RE) of ADTC to 0).

Remark x : Don't care
 PMxx : Port mode register
 Pxx : Port output latch

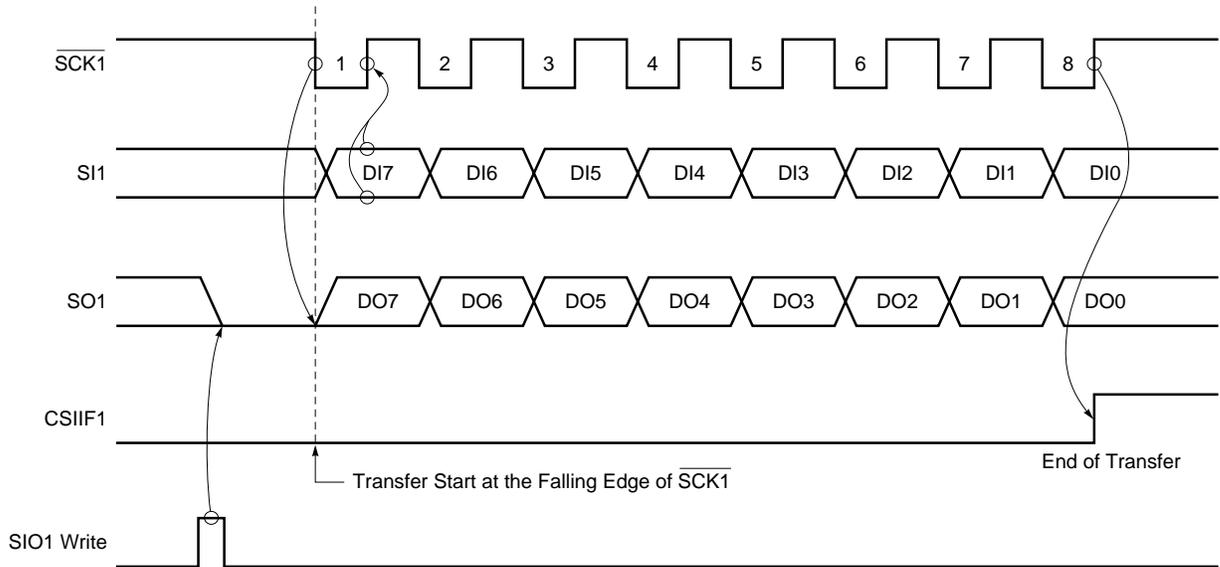
(2) Communication operation

The 3-wire serial I/O mode is used for data transmission/reception in 8-bit units. Bit-wise data transmission/reception is carried out in synchronization with the serial clock.

Shift operation of the serial I/O shift register 1 (SIO1) is carried out at the falling edge of the serial clock $\overline{\text{SCK1}}$. The transmit data is held in the SO1 latch and is output from the SO1 pin. The receive data input to the SI1 pin is latched into SIO1 at the rising edge of $\overline{\text{SCK1}}$.

Upon termination of 8-bit transfer, the SIO1 operation stops automatically and the interrupt request flag (CSIF1) is set.

Figure 19-6. 3-Wire Serial I/O Mode Timings



Caution SO1 pin becomes low level by SIO1 write.

★

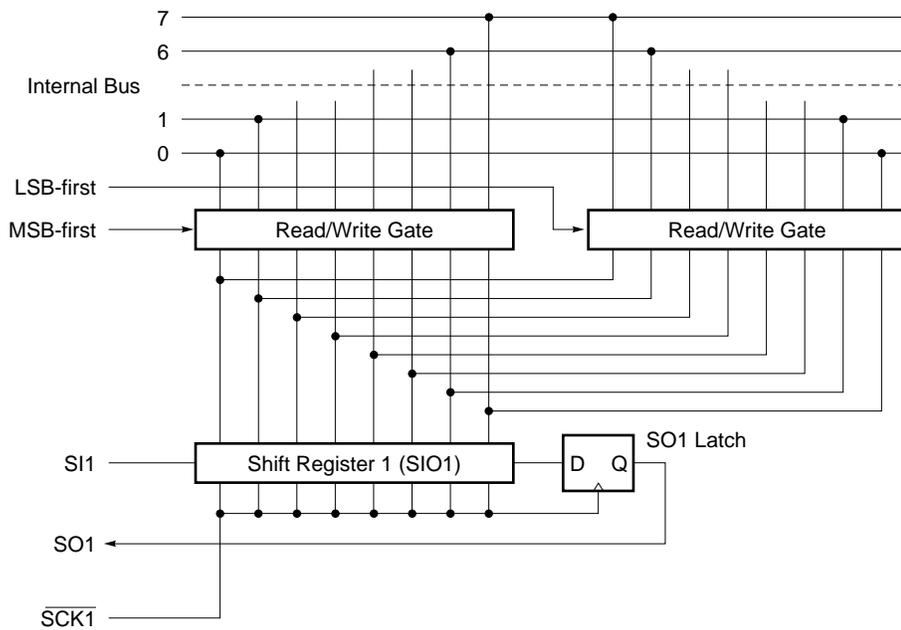
(3) MSB/LSB switching as the start bit

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

Figure 19-7 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 19-7. Circuit of Switching in Transfer Bit Order



Start bit switching is realized by switching the bit order write to SIO1. The SIO1 shift order remains unchanged. Thus, MSB-first and LSB-first must be switched before writing data to the shift register.

(4) Transfer start

Serial transfer is started by setting transfer data to the serial I/O shift register 1 (SIO1) when the following two conditions are satisfied.

- Serial interface channel 1 operation control bit (CSIE1) = 1
- Internal serial clock is stopped or $\overline{\text{SCK1}}$ is a high level after 8-bit serial transfer.

Caution If CSIE1 is set to “1” after data write to SIO1, transfer does not start.

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (CSIF1) is set.

19.4.3 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 specify 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 00H.

Symbol	<7>	6	<5>	4	3	2	1	0	Address	After Reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM11	CSIM10	FF68H	00H	R/W

CSIM11	CSIM10	Serial Interface Channel 1 Clock Selection
0	x	Clock externally input to SCK1 pin ^{Note 1}
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 4 to 7 of timer clock select register 3 (TCL3)

ATE	Serial Interface Channel 1 Operating Mode Selection
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

DIR	Start Bit	SI1 Pin Function	SO1 Pin Function
0	MSB	SI1/P20 (Input)	SO1 (CMOS output)
1	LSB		

CSIE	CSIM	PM20	P20	PM21	P21	PM22	P22	Shift Register 1 Operation	Serial Clock Counter Operation Control	SI1/P20 Pin Function	SO1/P21 Pin Function	SCK1/P22 Pin Function
1	11							Operation	Clear	P20 (CMOS input/output)	P21 (CMOS input/output)	P22 (CMOS input/output)
0	x	Note 2	Operation stop									
1	0	Note 3	Note 3	0	0	1	x	Operation enable	Count operation	SI1 ^{Note 3} (Input)	SO1 (CMOS output)	SCK1 (Input)
	1					0	1					SCK1 (CMOS output)

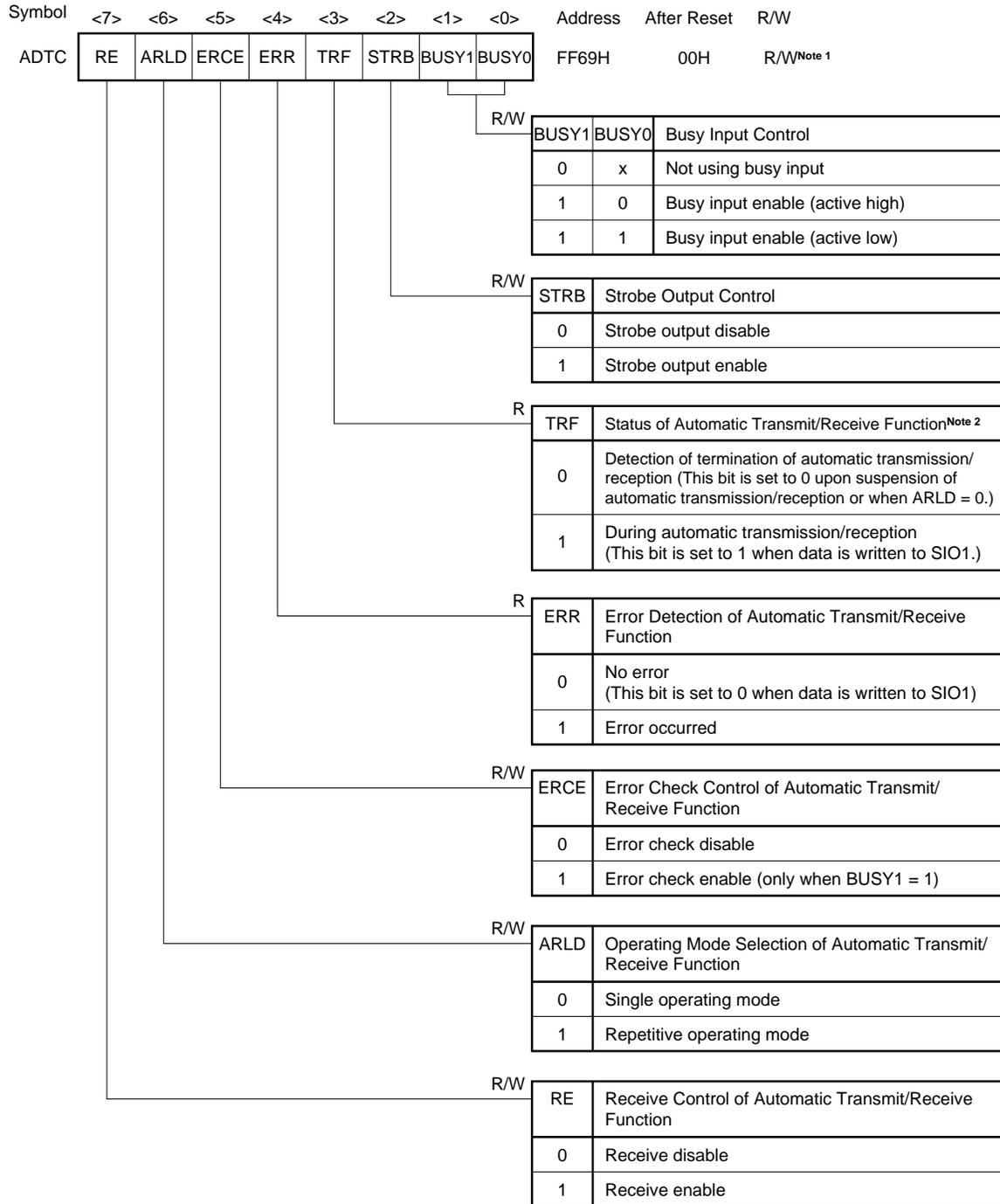
- Notes**
1. If the external clock input has been selected with CSIM11 set to 0, set bit 1 (BUSY1) and bit 2 (STRB) of the automatic data transmit/receive control register (ADTC) to 0, 0.
 2. Can be used freely as port function.
 3. Can be used as P20 (CMOS input/output) when only transmitter is used (set bit 7 (RE) of ADTC to 0).

Remark x : Don't care
 PMxx : Port mode register
 Pxx : Port output latch

(b) Automatic data transmit/receive control register (ADTC)

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

RESET input sets ADTC to 00H.



Notes 1. Bits 3 and 4 (TRF and ERR) are read-only bits.

2. Judge the termination of automatic transmission/reception by using TRF, not CSIF1 (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, handshake control cannot be carried out).

Remark x: Don't care

(c) Automatic data transmit/receive interval specify 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.

$\overline{\text{RESET}}$ 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} = 5.0-MHz Operation)	
					Minimum ^{Note 2}	Maximum ^{Note 2}
0	0	0	0	0	18.4 μs + 0.5/f _{sck}	20.0 μs + 1.5/f _{sck}
0	0	0	0	1	31.2 μs + 0.5/f _{sck}	32.8 μs + 1.5/f _{sck}
0	0	0	1	0	44.0 μs + 0.5/f _{sck}	45.6 μs + 1.5/f _{sck}
0	0	0	1	1	56.8 μs + 0.5/f _{sck}	58.4 μs + 1.5/f _{sck}
0	0	1	0	0	69.6 μs + 0.5/f _{sck}	71.2 μs + 1.5/f _{sck}
0	0	1	0	1	82.4 μs + 0.5/f _{sck}	84.0 μs + 1.5/f _{sck}
0	0	1	1	0	95.2 μs + 0.5/f _{sck}	96.8 μs + 1.5/f _{sck}
0	0	1	1	1	108.0 μs + 0.5/f _{sck}	109.6 μs + 1.5/f _{sck}
0	1	0	0	0	120.8 μs + 0.5/f _{sck}	122.4 μs + 1.5/f _{sck}
0	1	0	0	1	133.6 μs + 0.5/f _{sck}	135.2 μs + 1.5/f _{sck}
0	1	0	1	0	146.4 μs + 0.5/f _{sck}	148.0 μs + 1.5/f _{sck}
0	1	0	1	1	159.2 μs + 0.5/f _{sck}	160.8 μs + 1.5/f _{sck}
0	1	1	0	0	172.0 μs + 0.5/f _{sck}	173.6 μs + 1.5/f _{sck}
0	1	1	0	1	184.8 μs + 0.5/f _{sck}	186.4 μs + 1.5/f _{sck}
0	1	1	1	0	197.6 μs + 0.5/f _{sck}	199.2 μs + 1.5/f _{sck}
0	1	1	1	1	210.4 μs + 0.5/f _{sck}	212.0 μs + 1.5/f _{sck}

Notes 1. The interval 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}.

$$\text{Minimum} = (n + 1) \times \frac{2^6}{f_{xx}} + \frac{28}{f_{xx}} + \frac{0.5}{f_{sck}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^6}{f_{xx}} + \frac{36}{f_{xx}} + \frac{1.5}{f_{sck}}$$

Cautions 1. Do not write to ADTI during operation of the automatic data transmit/receive function.

2. Bits 5 and 6 must be set to 0.

3. When controlling the data transfer interval by automatic transmit/receive using ADTI, busy control (refer to 19.4.3 (4) (a) Busy control option) becomes invalid.

Remark f_{xx} : Main system clock frequency (f_x or f_x/2)
 f_x : Main system clock oscillation frequency
 f_{sck} : Serial clock frequency

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data Transfer Interval Specification (f _{xx} = 5.0-MHz Operation)	
					Minimum ^{Note}	Maximum ^{Note}
1	0	0	0	0	223.2 μs + 0.5/f _{sck}	224.8 μs + 1.5/f _{sck}
1	0	0	0	1	236.0 μs + 0.5/f _{sck}	237.6 μs + 1.5/f _{sck}
1	0	0	1	0	248.8 μs + 0.5/f _{sck}	250.4 μs + 1.5/f _{sck}
1	0	0	1	1	261.6 μs + 0.5/f _{sck}	263.2 μs + 1.5/f _{sck}
1	0	1	0	0	274.4 μs + 0.5/f _{sck}	276.0 μs + 1.5/f _{sck}
1	0	1	0	1	287.2 μs + 0.5/f _{sck}	288.8 μs + 1.5/f _{sck}
1	0	1	1	0	300.0 μs + 0.5/f _{sck}	301.6 μs + 1.5/f _{sck}
1	0	1	1	1	312.8 μs + 0.5/f _{sck}	314.4 μs + 1.5/f _{sck}
1	1	0	0	0	325.6 μs + 0.5/f _{sck}	327.2 μs + 1.5/f _{sck}
1	1	0	0	1	338.4 μs + 0.5/f _{sck}	340.0 μs + 1.5/f _{sck}
1	1	0	1	0	351.2 μs + 0.5/f _{sck}	352.8 μs + 1.5/f _{sck}
1	1	0	1	1	364.0 μs + 0.5/f _{sck}	365.6 μs + 1.5/f _{sck}
1	1	1	0	0	376.8 μs + 0.5/f _{sck}	378.4 μs + 1.5/f _{sck}
1	1	1	0	1	389.6 μs + 0.5/f _{sck}	391.2 μs + 1.5/f _{sck}
1	1	1	1	0	402.4 μs + 0.5/f _{sck}	404.0 μs + 1.5/f _{sck}
1	1	1	1	1	415.2 μs + 0.5/f _{sck}	416.8 μs + 1.5/f _{sck}

Note The 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 to ADTI during operation of the automatic data transmit/receive function.
 2. Bits 5 and 6 must be set to 0.
 3. When controlling the data transfer interval by automatic transmit/receive using ADTI, busy control (refer to 19.4.3 (4) (a) Busy control option) becomes invalid.

Remark f_{xx} : Main system clock frequency (fx or fx/2)
 fx : Main system clock oscillation frequency
 f_{sck} : Serial clock frequency

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

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} = 2.5-MHz Operation)	
					Minimum ^{Note 2}	Maximum ^{Note 2}
0	0	0	0	0	36.8 μs + 0.5/f _{sck}	40.0 μs + 1.5/f _{sck}
0	0	0	0	1	62.4 μs + 0.5/f _{sck}	65.6 μs + 1.5/f _{sck}
0	0	0	1	0	88.0 μs + 0.5/f _{sck}	91.2 μs + 1.5/f _{sck}
0	0	0	1	1	113.6 μs + 0.5/f _{sck}	116.8 μs + 1.5/f _{sck}
0	0	1	0	0	139.2 μs + 0.5/f _{sck}	142.4 μs + 1.5/f _{sck}
0	0	1	0	1	164.8 μs + 0.5/f _{sck}	168.0 μs + 1.5/f _{sck}
0	0	1	1	0	190.4 μs + 0.5/f _{sck}	193.6 μs + 1.5/f _{sck}
0	0	1	1	1	216.0 μs + 0.5/f _{sck}	219.2 μs + 1.5/f _{sck}
0	1	0	0	0	241.6 μs + 0.5/f _{sck}	244.8 μs + 1.5/f _{sck}
0	1	0	0	1	267.2 μs + 0.5/f _{sck}	270.4 μs + 1.5/f _{sck}
0	1	0	1	0	292.8 μs + 0.5/f _{sck}	296.0 μs + 1.5/f _{sck}
0	1	0	1	1	318.4 μs + 0.5/f _{sck}	321.6 μs + 1.5/f _{sck}
0	1	1	0	0	344.0 μs + 0.5/f _{sck}	347.2 μs + 1.5/f _{sck}
0	1	1	0	1	369.6 μs + 0.5/f _{sck}	372.8 μs + 1.5/f _{sck}
0	1	1	1	0	395.2 μs + 0.5/f _{sck}	398.4 μs + 1.5/f _{sck}
0	1	1	1	1	420.8 μs + 0.5/f _{sck}	424.0 μs + 1.5/f _{sck}

Notes 1. The interval is dependent only on CPU processing.

2. The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than 2/f_{sck}, the minimum interval time is 2/f_{sck}.

$$\text{Minimum} = (n + 1) \times \frac{2^6}{f_{xx}} + \frac{28}{f_{xx}} + \frac{0.5}{f_{sck}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^6}{f_{xx}} + \frac{36}{f_{xx}} + \frac{1.5}{f_{sck}}$$

Cautions 1. Do not write to ADTI during operation of the automatic data transmit/receive function.

2. Bits 5 and 6 must be set to 0.

3. When controlling the data transfer interval by automatic transmit/receive using ADTI, busy control (refer to 19.4.3 (4) (a) Busy control option) becomes invalid.

Remark f_{xx} : Main system clock frequency (fx or fx/2)
 fx : Main system clock oscillation frequency
 f_{sck} : Serial clock frequency

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

					Data Transfer Interval Specification (f _{xx} = 2.5-MHz Operation)	
ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Minimum ^{Note}	Maximum ^{Note}
1	0	0	0	0	446.4 μs + 0.5/f _{sck}	449.6 μs + 1.5/f _{sck}
1	0	0	0	1	472.0 μs + 0.5/f _{sck}	475.2 μs + 1.5/f _{sck}
1	0	0	1	0	497.6 μs + 0.5/f _{sck}	500.8 μs + 1.5/f _{sck}
1	0	0	1	1	523.2 μs + 0.5/f _{sck}	526.4 μs + 1.5/f _{sck}
1	0	1	0	0	548.8 μs + 0.5/f _{sck}	552.0 μs + 1.5/f _{sck}
1	0	1	0	1	574.4 μs + 0.5/f _{sck}	577.6 μs + 1.5/f _{sck}
1	0	1	1	0	600.0 μs + 0.5/f _{sck}	603.2 μs + 1.5/f _{sck}
1	0	1	1	1	625.6 μs + 0.5/f _{sck}	628.8 μs + 1.5/f _{sck}
1	1	0	0	0	651.2 μs + 0.5/f _{sck}	654.4 μs + 1.5/f _{sck}
1	1	0	0	1	676.8 μs + 0.5/f _{sck}	680.0 μs + 1.5/f _{sck}
1	1	0	1	0	702.4 μs + 0.5/f _{sck}	705.6 μs + 1.5/f _{sck}
1	1	0	1	1	728.0 μs + 0.5/f _{sck}	731.2 μs + 1.5/f _{sck}
1	1	1	0	0	753.6 μs + 0.5/f _{sck}	756.8 μs + 1.5/f _{sck}
1	1	1	0	1	779.2 μs + 0.5/f _{sck}	782.4 μs + 1.5/f _{sck}
1	1	1	1	0	804.8 μs + 0.5/f _{sck}	808.0 μs + 1.5/f _{sck}
1	1	1	1	1	830.4 μs + 0.5/f _{sck}	833.6 μs + 1.5/f _{sck}

Note The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than 2/f_{sck}, the minimum interval time is 2/f_{sck}.

$$\text{Minimum} = (n + 1) \times \frac{2^6}{f_{xx}} + \frac{28}{f_{xx}} + \frac{0.5}{f_{sck}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^6}{f_{xx}} + \frac{36}{f_{xx}} + \frac{1.5}{f_{sck}}$$

- Cautions**
1. Do not write to ADTI during operation of the automatic data transmit/receive function.
 2. Bits 5 and 6 must be set to 0.
 3. When controlling the data transfer interval by automatic transmit/receive using ADTI, busy control (refer to 19.4.3 (4) (a) Busy control option) becomes invalid.

Remark f_{xx} : Main system clock frequency (fx or fx/2)
 fx : Main 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 CSIE1 and ATE of the serial operating mode register 1 (CSIM1) to 1.
- <2> Set RE of the automatic data transmit/receive control register (ADTC) to 1.
- <3> Set a data transmit/receive interval in the automatic data transmit/receive interval specify register (ADTI).
- <4> Write any value to the serial I/O shift register 1 (SIO1) (transfer start trigger).

Caution Writing any value to SIO1 orders the start of automatic transmit/receive operation and the written value has no meaning.

The following operations are automatically carried out when (a) and (b) are carried out.

- After the buffer RAM data specified with ADTP is transferred to SIO1, transmission is carried out (start of automatic transmission/reception).
- The received data is written to the buffer RAM address specified with ADTP.
- ADTP is decremented and the next data transmission/reception is carried out. Data transmission/reception continues until the ADTP decremental output becomes 00H and address FAC0H data is output (end of automatic transmission/reception).
- When automatic transmission/reception is terminated, 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.

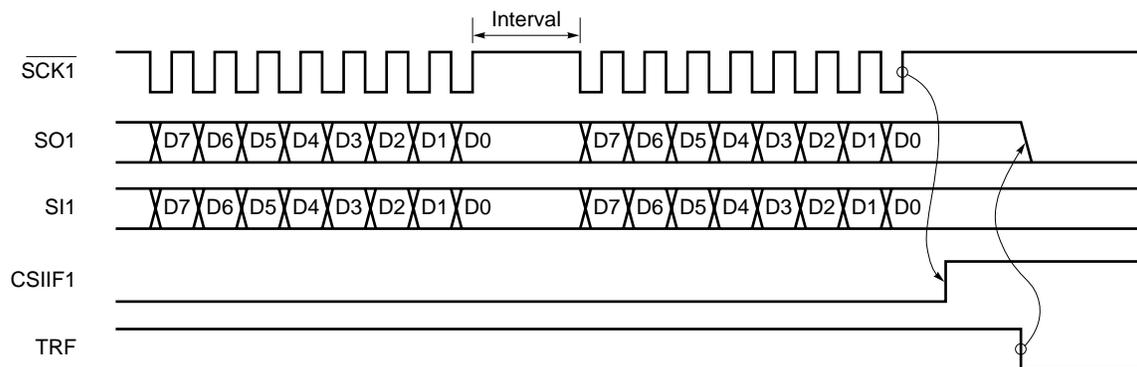
Serial transfer is started when any data is written to the serial I/O shift register 1 (SIO1) while bit 7 (CSIE1) of the serial operating mode register 1 (CSIM1) is set to 1.

Upon completion of transmission of the last byte, the interrupt request flag (CSIF1) is set. However, determine whether the automatic transmission/reception is completed, not with CSIF1 but with bit 3 (TRF) of the automatic data transmission/reception control register (ADTC).

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 19-8 shows the basic transmission/reception mode operation timings, and Figure 19-9 shows the operation flowchart. Figure 19-10 shows an example of the buffer RAM operation in 6-byte transmission/reception.

Figure 19-8. Basic Transmission/Reception Mode Operation Timings



Cautions

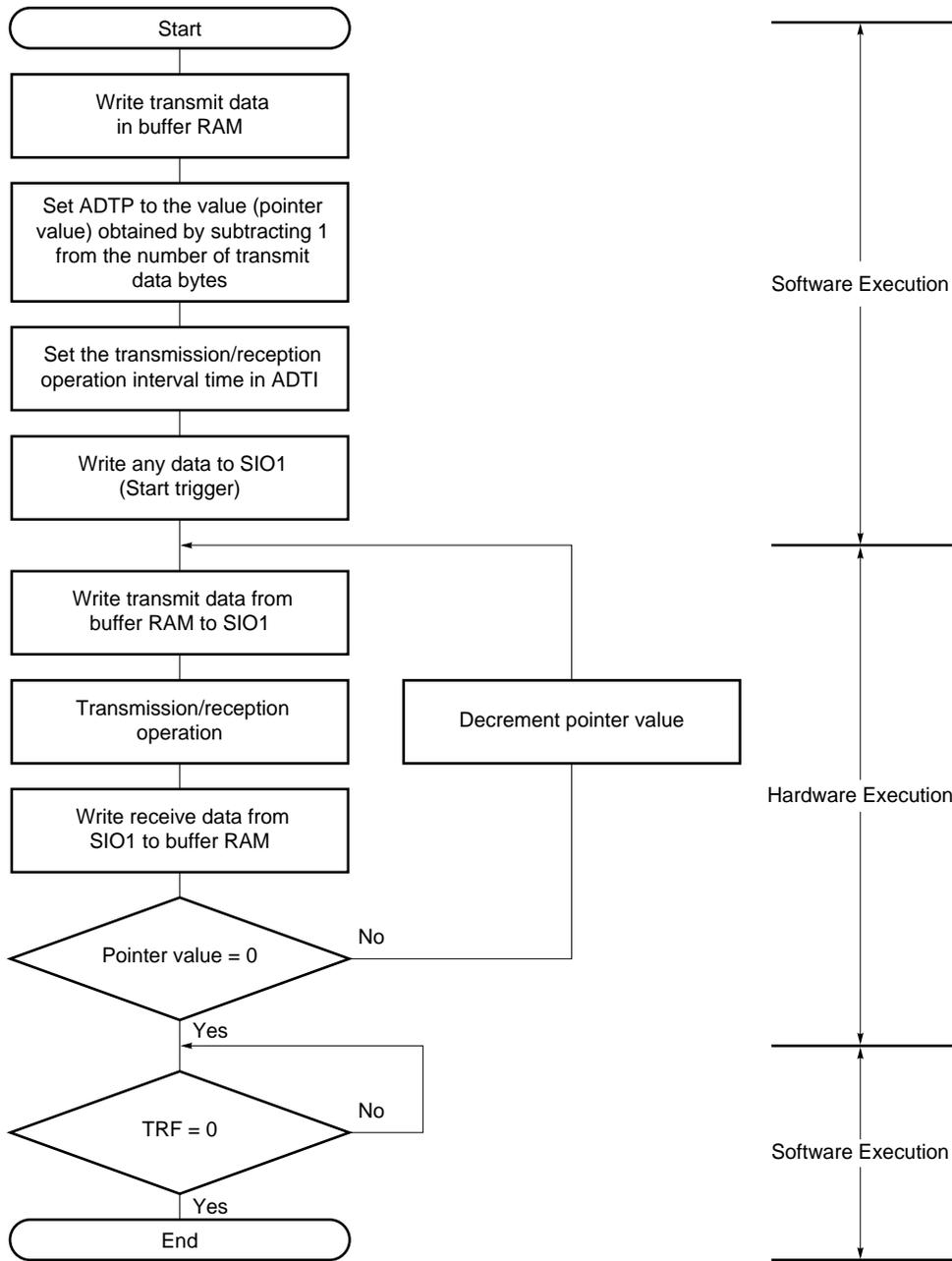
1. Because, in the basic transmission/reception mode, the automatic transmit/receive function writes/reads data to/from the buffer RAM after 1-byte transmission/reception, an interval is inserted till the next transmission/reception. As the buffer RAM write/read is performed at the same time as CPU processing, the maximum interval is dependent upon CPU processing and the value of the automatic data transmit/receive interval specify register (ADTI) (see (5) Automatic transmit/receive interval time).

2. When TRF is cleared, the SO1 pin becomes low level.

Remark CSIF0 : Interrupt request flag

TRF : Bit 3 of automatic data transmit/receive control register (ADTC)

Figure 19-9. Basic Transmission/Reception Mode Flowchart



ADTP: Automatic data transmit/receive address pointer
 ADTI : Automatic data transmit/receive interval specify 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 19-10 (a))

After any data has been written to the serial I/O shift register 1 (SIO1) (start trigger: this data is not transferred), transmit data 1 (T1) is transferred from the buffer RAM to SIO1. When transmission of the first byte is completed, the receive data 1 (R1) is transferred from SIO1 to the buffer RAM, and automatic data transmit/receive address pointer (ADTP) is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1.

(ii) 4th byte transmission/reception point (refer to Figure 19-10 (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 19-10 (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 (CSIF1) is set (INTCS11 generation).

Figure 19-10. Buffer RAM Operation in 6-byte Transmission/Reception (in Basic Transmit/Receive Mode) (1/2)

(a) Before transmission/reception

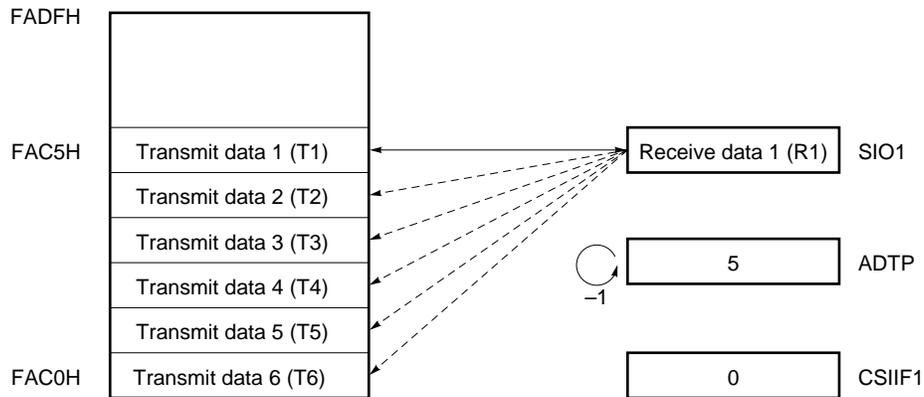
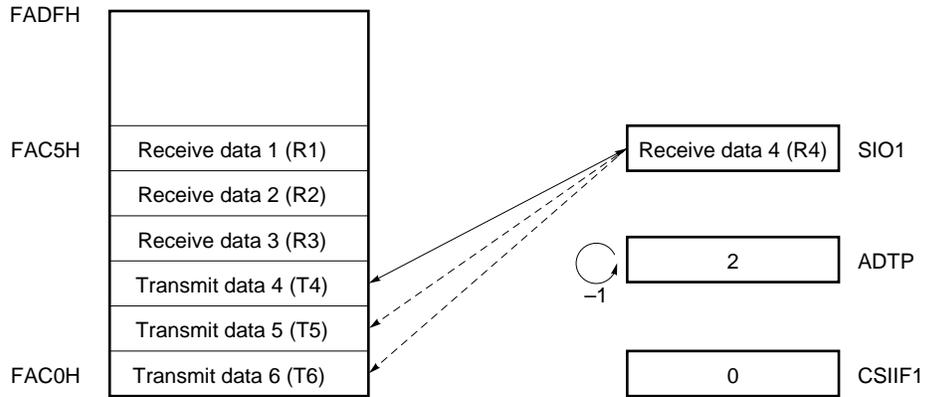
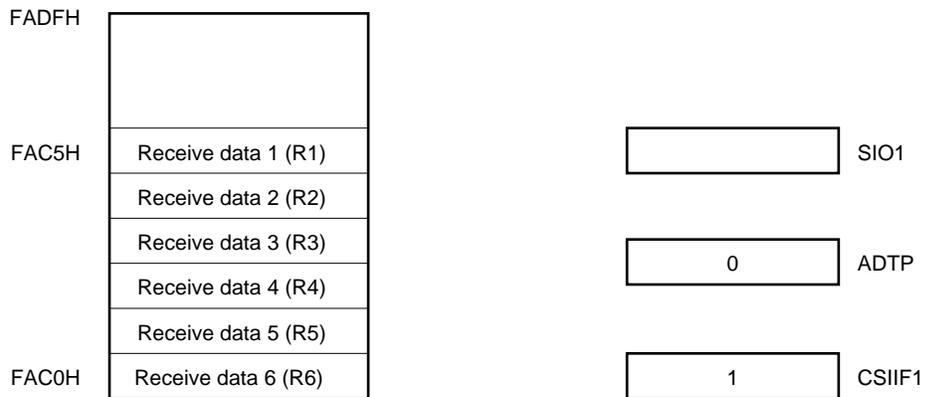


Figure 19-10. Buffer RAM Operation in 6-byte Transmission/Reception (in Basic Transmit/Receive Mode) (2/2)

(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.

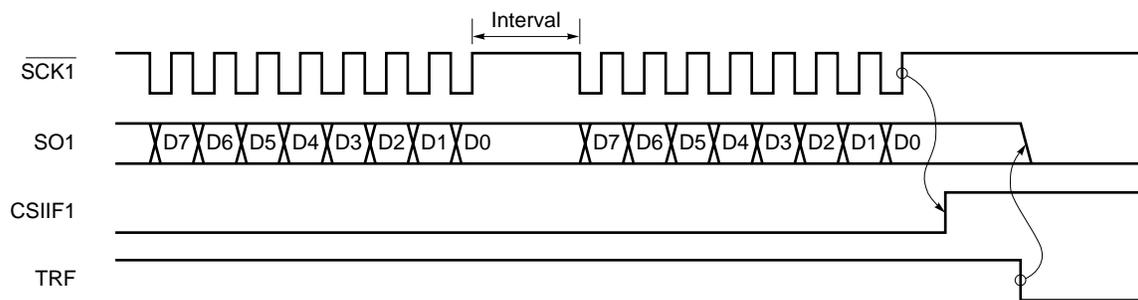
Serial transfer is started when any data is written to the serial I/O shift register 1 (SIO1) while bit 7 (CSIE1) of the serial operating mode register 1 (CSIM1) is set to 1.

Upon completion of transmission of the last byte, the interrupt request flag (CSIF1) is set. However, determine whether the automatic transmission/reception is completed, not with CSIF1 but with the bit 3 (TRF) of automatic data transmission/reception control register (ADTC).

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 19-11 shows the basic transmission mode operation timings, and Figure 19-12 shows the operation flowchart. Figure 19-13 shows an example of the buffer RAM operation in 6-byte transmission.

Figure 19-11. Basic Transmission Mode Operation Timings



Cautions

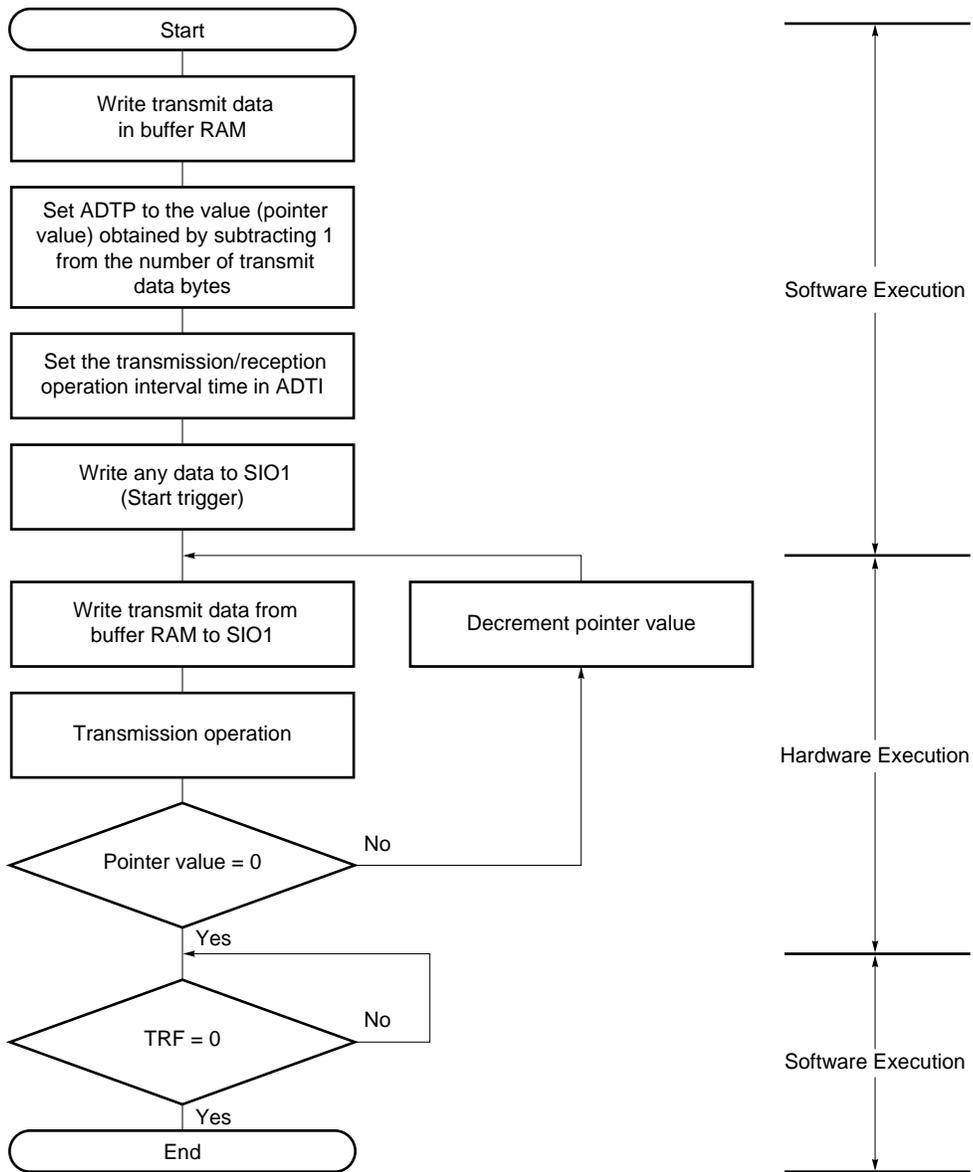
1. Because, in the basic transmission mode, the automatic transmit/receive function reads data from the buffer RAM after 1-byte transmission, an interval is inserted till the next transmission. As the buffer RAM read is performed at the same time as CPU processing, the maximum interval is dependent upon CPU processing and the value of the automatic data transmit/receive interval specify register (ADTI) (see (5) Automatic transmit/receive interval time).

2. When TRF is cleared, the SO1 pin becomes low level.

Remark CSIF1: Interrupt request flag

TRF : Bit 3 of automatic data transmit/receive control register (ADTC)

Figure 19-12. Basic Transmission Mode Flowchart



ADTP : Automatic data transmit/receive address pointer
 ADTI : Automatic data transmit/receive interval specify 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 Figure 19-13 (a))

After any data has been written to the serial I/O shift register 1 (SIO1) (start trigger: this data is not transferred), transmit data 1 (T1) is transferred from the buffer RAM to SIO1. When transmission of the first byte is completed, the automatic data transmit/receive address pointer (ADTP) is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1.

(ii) 4th byte transmission point (refer to Figure 19-13 (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 19-13 (c))

When transmission of the sixth byte is completed, the interrupt request flag (CSIF1) is set (INTCSI1 generation).

Figure 19-13. Buffer RAM Operation in 6-byte Transmission (in Basic Transmit Mode) (1/2)

(a) Before transmission

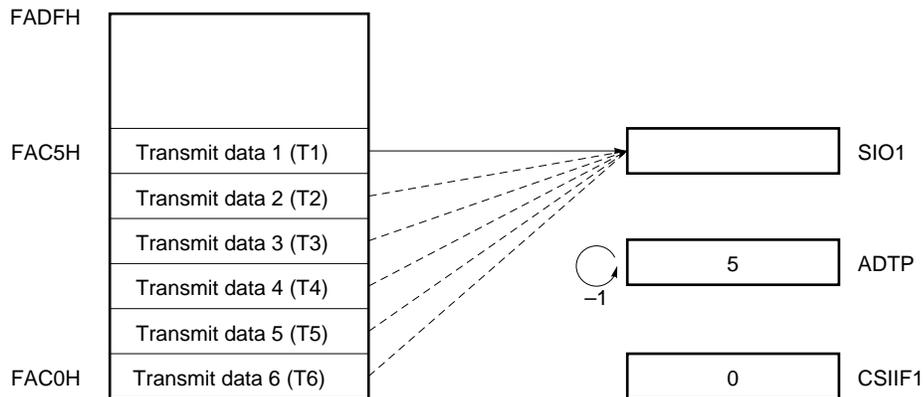
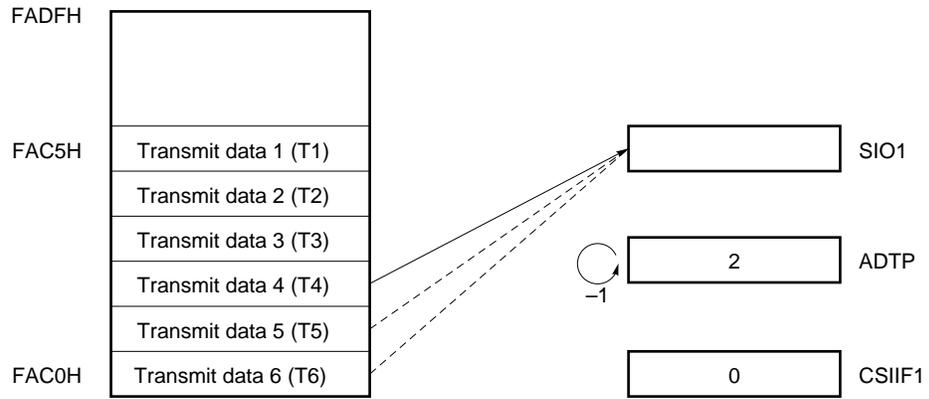
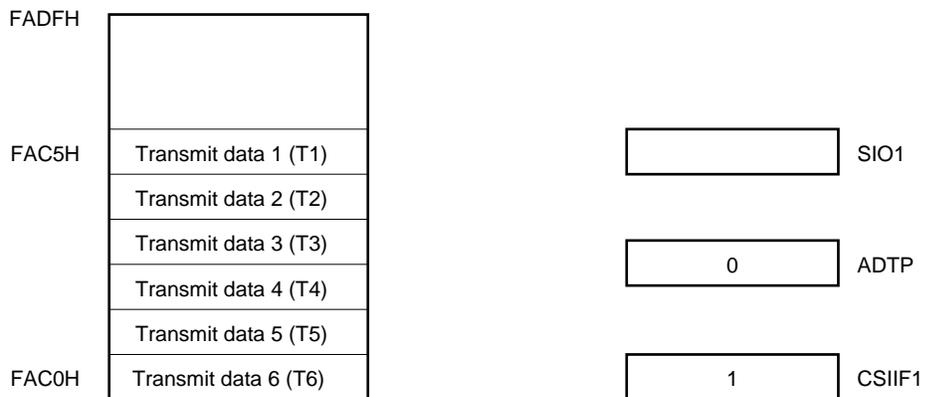


Figure 19-13. Buffer RAM Operation in 6-byte Transmission (in Basic Transmit Mode) (2/2)

(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.

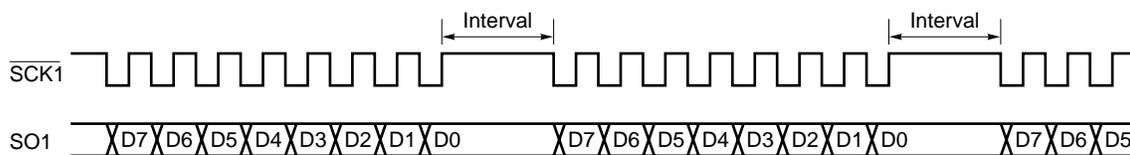
Serial transfer is started by writing any data to serial I/O shift register 1 (SIO1) when 1 is set in bit 7 (CSIE1) of the serial operating mode register 1 (CSIM1).

Unlike the basic transmission mode, after the last byte (data in address FAC0H) has been transmitted, the interrupt request flag (CSIF1) 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.

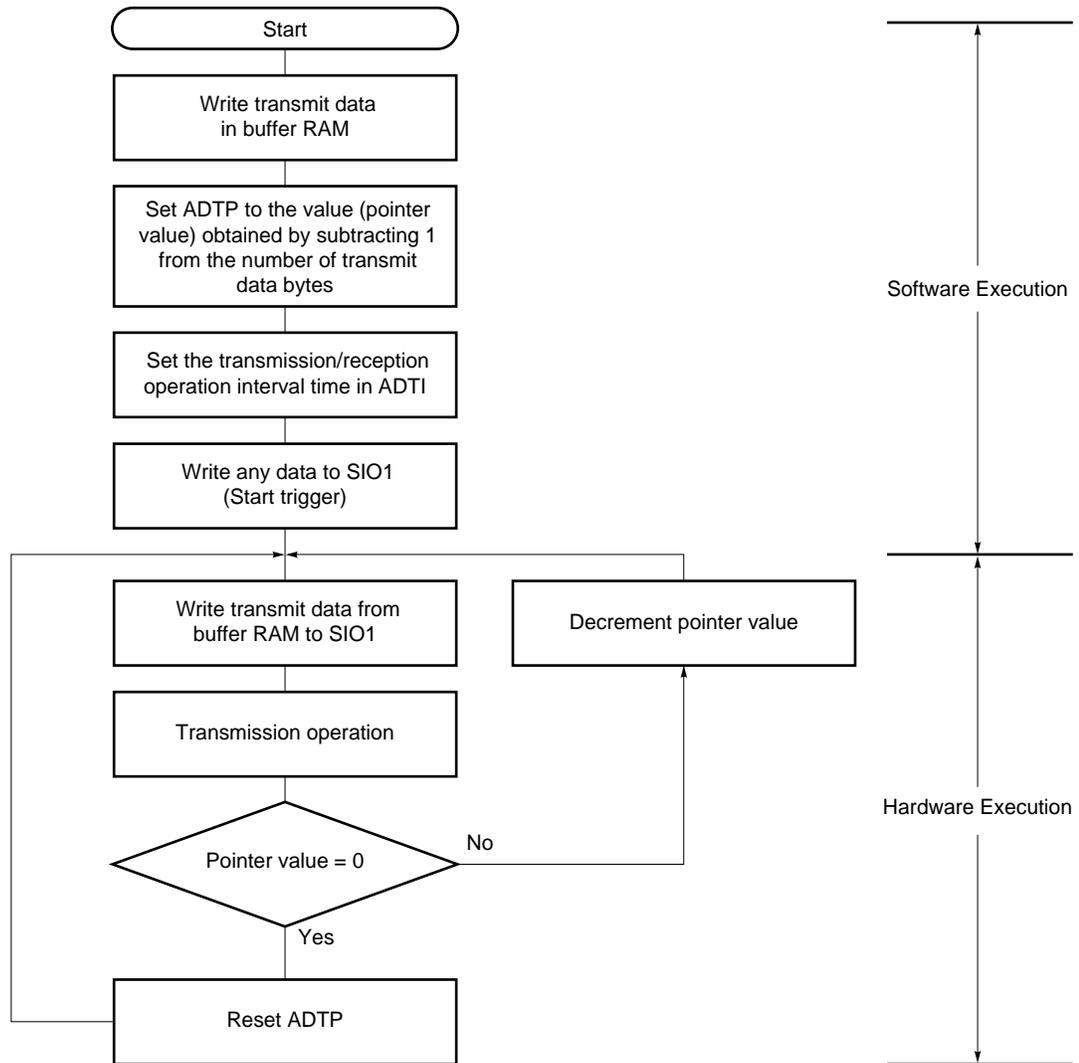
The repeat transmission mode operation timing is shown in Figure 19-14, and the operation flowchart in Figure 19-15. Figure 19-16 shows an example of the buffer RAM operation in 6-byte repeat transmission.

Figure 19-14. 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 specify register (ADTI) (see (5) Automatic transmit/receive interval time).

Figure 19-15. Repeat Transmission Mode Flowchart



ADTP : Automatic data transmit/receive address pointer
 ADTI : Automatic data transmit/receive interval specify 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 19-16 (a))

After any data has been written to the serial I/O shift register 1 (SIO1) (start trigger: this data is not transferred), transmit data 1 (T1) is transferred from the buffer RAM to SIO1. When transmission of the first byte is completed, the automatic data transmit/receive address pointer (ADTP) is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1.

(ii) Upon completion of transmission of 6 bytes (refer to Figure 19-16 (b))

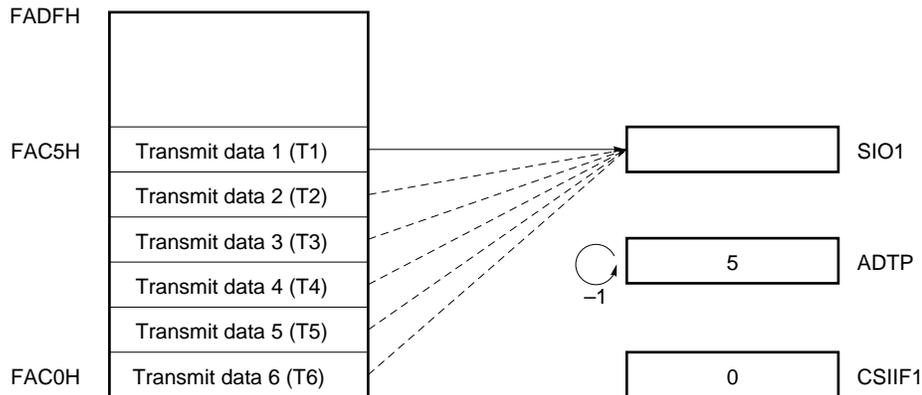
When transmission of the sixth byte is completed, the interrupt request flag (CSIIF1) is not set. The first pointer value is set to ADTP again.

(iii) 7th byte transmission point (refer to Figure 19-16 (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.

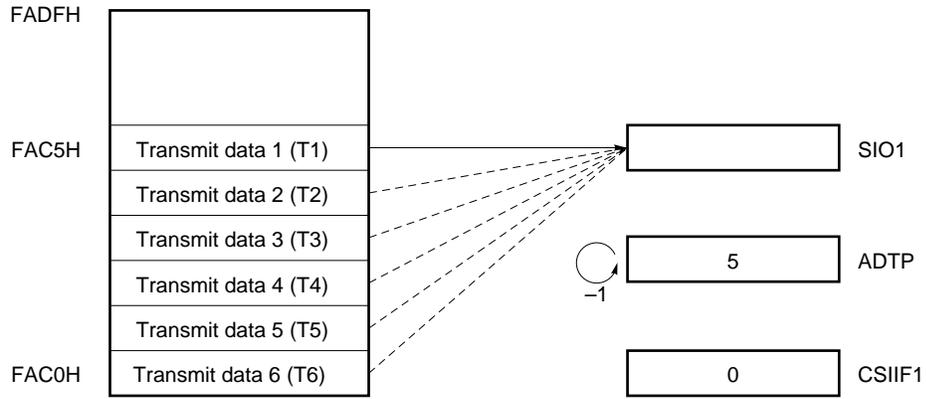
**Figure 19-16. Buffer RAM Operation in 6-byte Transmission
(in Repeat Transmit Mode) (1/2)**

(a) Before transmission

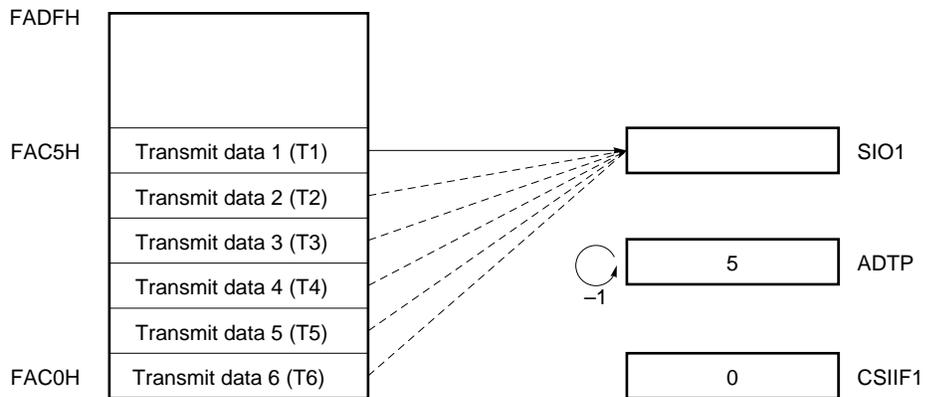


**Figure 19-16. Buffer RAM Operation in 6-byte Transmission
(in Repeat Transmit Mode) (2/2)**

(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.

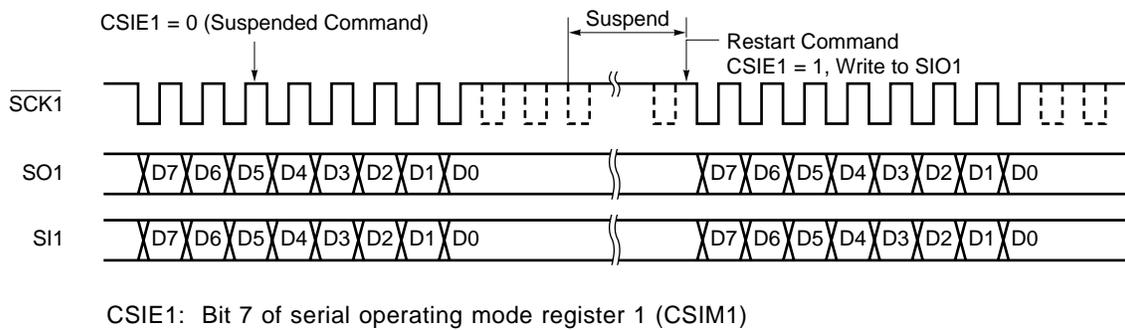
When suspended, bit 3 (TRF) of the automatic data transmit/receive control register (ADTC) is set to 0 after transfer of the 8th bit, and all the port pins used with the serial interface pins for dual function (P20/SI1, P21/SO1, P22/ $\overline{\text{SCK1}}$, P23/STB, and P24/BUSY) are set to the port mode.

For restart of automatic 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 19-17. Automatic Transmission/Reception Suspension and Restart



- ★ **(4) Synchronization control**
 Busy control and strobe control are functions for synchronizing sending and receiving between the master device and slave device.
 By using these functions, it is possible to detect bit slippage during sending and receiving.

(a) Busy control option

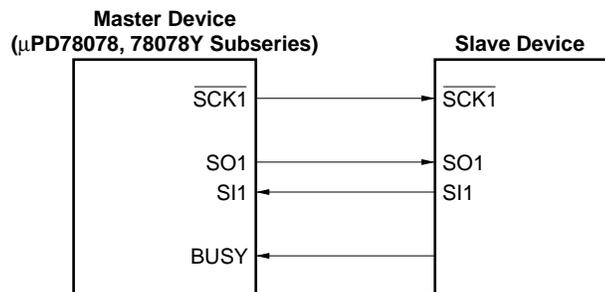
Busy control is a function which causes the master device's serial transmission to wait when the slave device outputs a busy signal to the master device, and maintain the wait state while that busy signal is active.

When the busy control option is used, the conditions shown below are necessary.

- Bit 5 (ATE) of serial operation mode register 1 (CSIM1) should be set at (1).
- Bit 1 (BUSY1) of the automatic data transmit/receive control register (ADTC) should be set at (1).

The system configuration between the master device and slave device in cases where the busy control option is used is shown in Figure 19-18.

Figure 19-18. System Configuration when the Busy Control Option is Used



The master device inputs the busy signal output by the slave device to pin BUSY/P24. In sync with the fall of the serial clock, the master device samples the input busy signal. Even if the busy signal becomes active during sending or receiving of 8 bit data, the wait does not apply. If the busy signal becomes active at the rise of the serial clock 2 clock cycles after sending or receiving of 8 bit data ends, the busy input first becomes effective at that point, and thereafter, sending or receiving of data waits during the period that the busy signal is active.

The busy signal's active level is set in bit 0 (BUSY0) of ADTC.

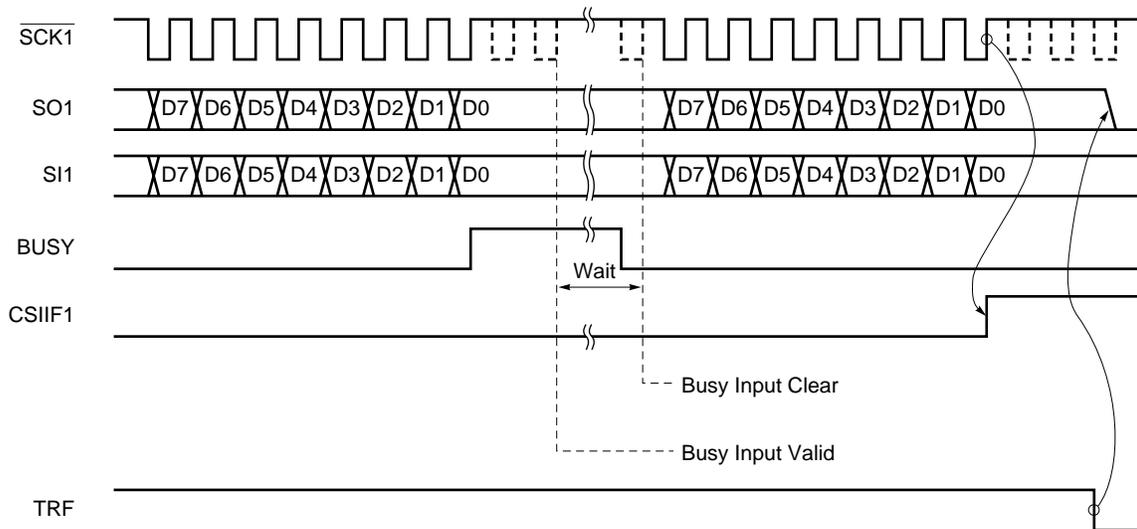
BUSY0 = 0: Active high

BUSY0 = 1: Active low

Furthermore, in the case that the busy control option is used, select the internal clock for the serial clock. The busy signal cannot be controlled with an external clock.

The operation timing when the busy control option is used is shown in Figure 19-19.

Caution Busy control cannot be used at the same time as interval timing control using the automatic data transmit/receive interval specify register (ADTI). If both are used simultaneously, busy control becomes invalid.

Figure 19-19. Operation Timings when Using Busy Control Option (BUSY0 = 0)

Caution When TRF is cleared, the SO1 pin becomes low level.

Remark CSIF1 : Interrupt request flag

TRF : Bit 3 of the automatic data transmit/receive control register (ADTC)

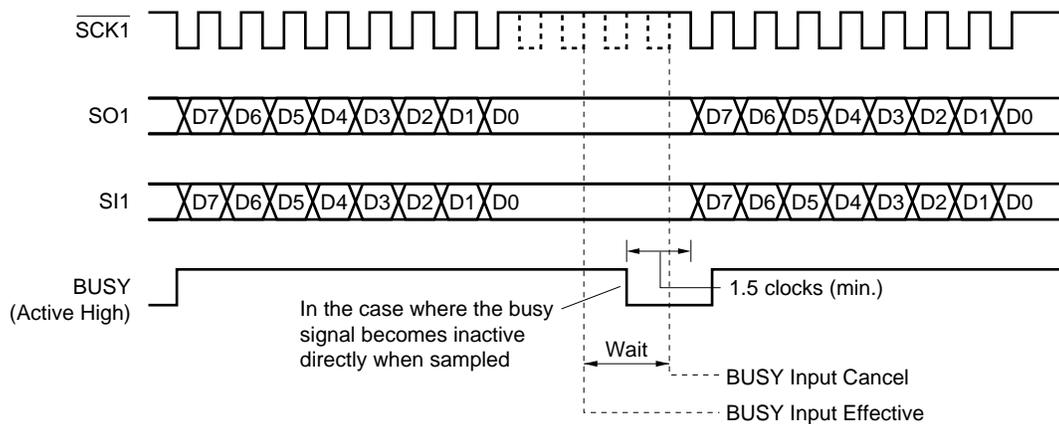
If the busy signal becomes inactive, the wait is canceled. If the sampled busy signal is inactive, sending or receiving of the next 8 bit data begins from the fall of the next serial clock cycle.

Furthermore, the busy signal is asynchronous with the serial clock, so even if the slave side inactivates the busy signal, it takes nearly 1 clock cycle at the most until it is sampled again. Also, it takes another 0.5 clock cycle after sampling until data transmission resumes.

Therefore, in order to definitely cancel a wait state, it is necessary for the slave side to keep the busy signal for at least 1.5 clock cycles.

Figure 19-20 shows the timing of the busy signal and wait cancel. In this figure, an example of the case where the busy signal becomes active when sending or receiving starts is shown.

Figure 19-20. Busy Signal and Wait Cancel (BUSY0 = 0)

**(b) Busy & strobe control option**

Strobe control is a function for synchronizing the sending and receiving of data between a master device and slave device. When sending or receiving of 8 bit data ends, the strobe signal is output by the master device from pin STB/P23. Through this means, the slave device can know the timing of the end of master data transmission. Therefore, even if there is noise in the serial clock and bit slippage occurs, synchronization is maintained and bit slippage has no effect on transmission of the next byte. In the case that the strobe control option is used, the conditions shown below are necessary.

- Set bit 5 (ATE) of serial operation mode register 1 (CSIM1) at (1).
- Set bit 2 (STRB) of the automatic data transmit/receive control register (ADTC) at (1).

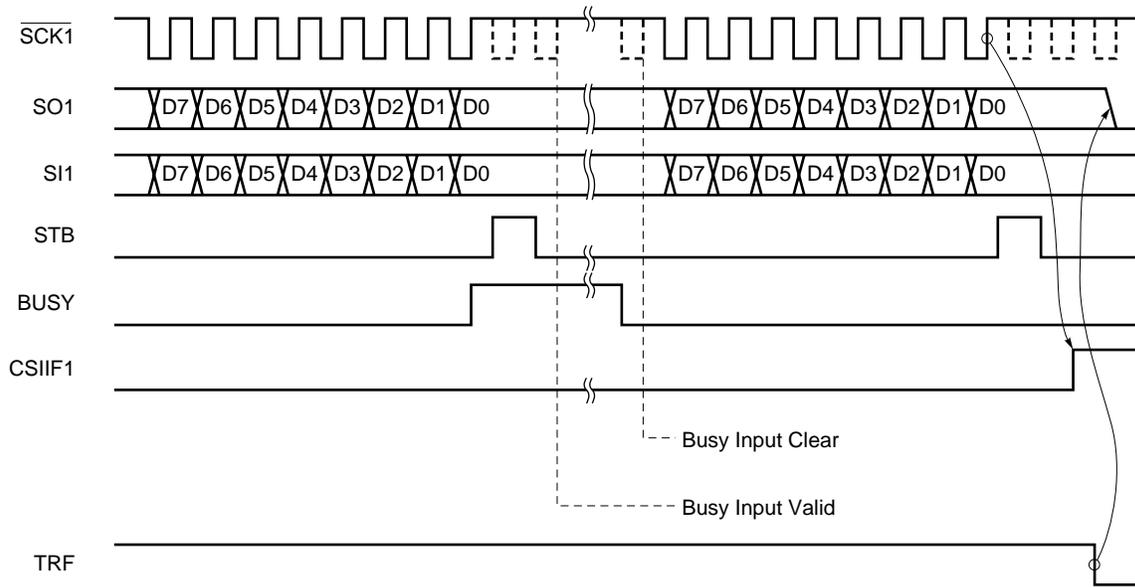
Normally, busy control and strobe control are used simultaneously as handshake signals. In this case, together with output of the strobe signal from pin STB/P23, pin BUSY/P24 can be sampled and sending or receiving can wait while the busy signal is being input.

If strobe control is not carried out, pin P23/STB can be used as a normal I/O port.

Operation timing when busy and strobe control are used is shown in Figure 19-21.

Furthermore, if strobe control is used, the interrupt request flag (CSIF1), set when sending or receiving ends, is set after the strobe signal is output.

Figure 19-21. Operation Timings when Using Busy & Strobe Control Option (BUSY0 = 0)



Caution When TRF is cleared, the SO1 pin becomes low level.

Remark CSIIIF1: Interrupt request flag

TRF : Bit 3 of the automatic data transmit/receive control register (ADTC)

(c) Bit slippage detection function through the busy signal

During an automatic transmit/receive operation, noise occur in the serial clock signal output by the master device and bit slippage may occur in the slave device side serial clock. At this time, if the strobe control option is not used, this bit slippage will have an effect on sending of the next byte. In such a case, the busy control option can be used on the master device side and, by checking the busy signal during sending, bit slippage can be detected.

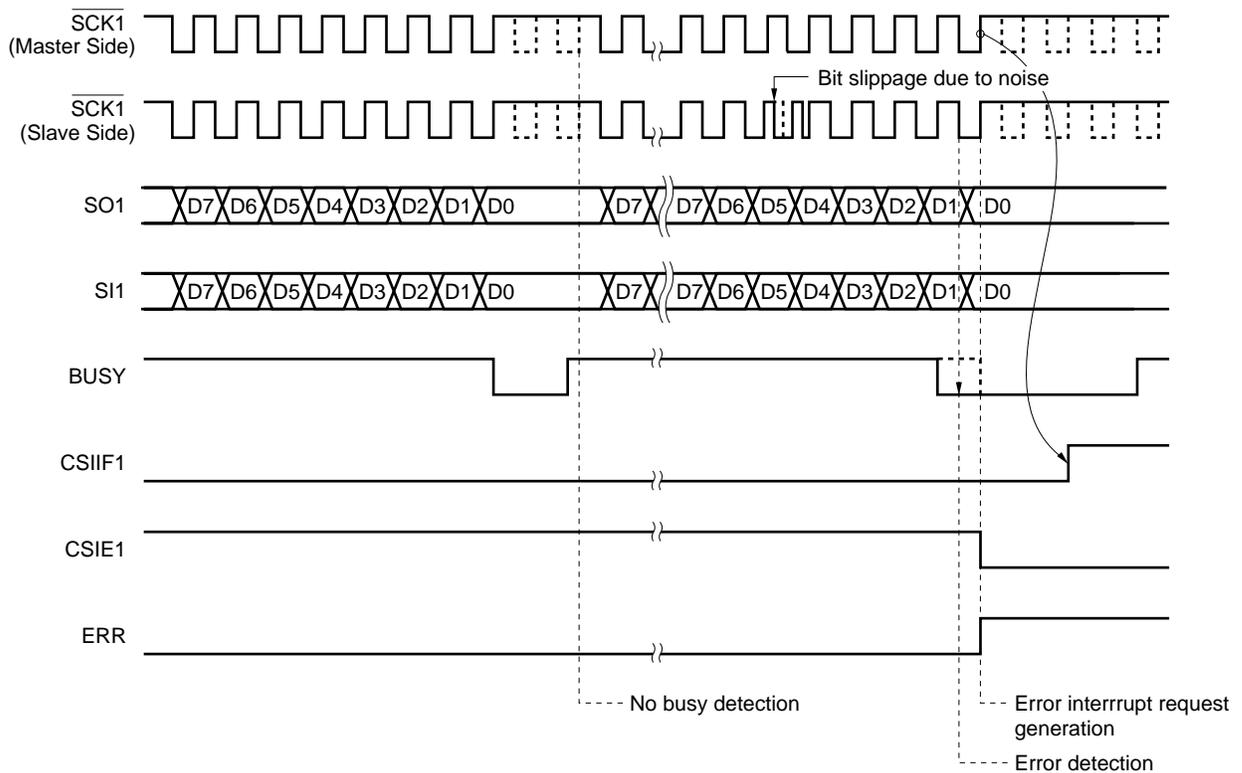
Bit slippage detection through the busy signal is accomplished as follows.

The slave side outputs a busy signal after the serial clock rises on the 8th cycle of data sending or receiving (at this time, if application of the wait state by the busy signal is not desired, the busy signal is made inactive within 2 clock cycles).

The master device side samples the busy signal in sync with the fall of the serial clock's front side. If no bit slippage is occurring, the busy signal will be inactive in sampling for 8 clock cycles. If the busy signal is found to be active in sampling, it is regarded as an occurrence of bit slippage error processing is executed (bit 4 (ERR) of the automatic data transmit/receive control register (ADTC) is set at (1)).

The operation timing of the bit slippage detection function through the busy signal is shown in Figure 19-22.

Figure 19-22. Operation Timing of the Bit Slippage Detection Function through the Busy Signal (BUSY0 = 1)



CSIF1: Interrupt request flag

CSIE1 : Bit 7 of serial operating mode register 1 (CSIM1)

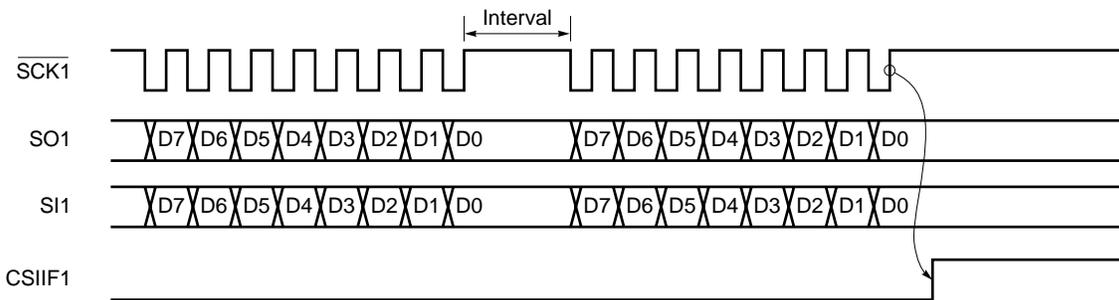
ERR : Bit 4 of the automatic data transmit/receive control register (ADTC)

(5) Automatic transmit/receive interval time

When using the automatic transmit/receive function, the read/write operations from/to the buffer RAM are performed after transmitting/receiving one byte. Therefore, an interval is inserted before the next transmit/receive.

Since the read/write operations from/to the buffer RAM are performed in parallel with the CPU processing when using the automatic transmit/receive function by the internal clock, the interval depends on the value which is set in the automatic transmit/receive interval specification register (ADTI) and the CPU processing at the rising edge of the eighth serial clock. Whether it depends on the ADTI or not can be selected by the setting of its bit 7 (ADTI7). When it is set to 0, the interval depends only on the CPU processing. When it is set to 1, the interval depends on the contents of the ADTI or CPU processing, whichever is greater. When the automatic transmit/receive function is used by an external clock, it must be selected so that the interval may be longer than the value indicated by paragraph (b).

Figure 19-23. Automatic Data Transmit/Receive Interval



CSIF1: Interrupt request flag

(a) When the automatic transmit/receive function is used by the internal clock

If bit 1 (CSIM11) of serial operation mode register 1 (CSIM1) is set at (1), the internal clock operates. If the automatic transmit/receive function is operated by the internal clock, interval timing by CPU processing is as follows.

When bit 7 (ADTI7) of automatic data transmit/receive interval specify register (ADTI) is set to 0, the interval depends on the CPU processing. When ADTI7 is set to 1, it depends on the contents of the ADTI or CPU processing, whichever is greater.

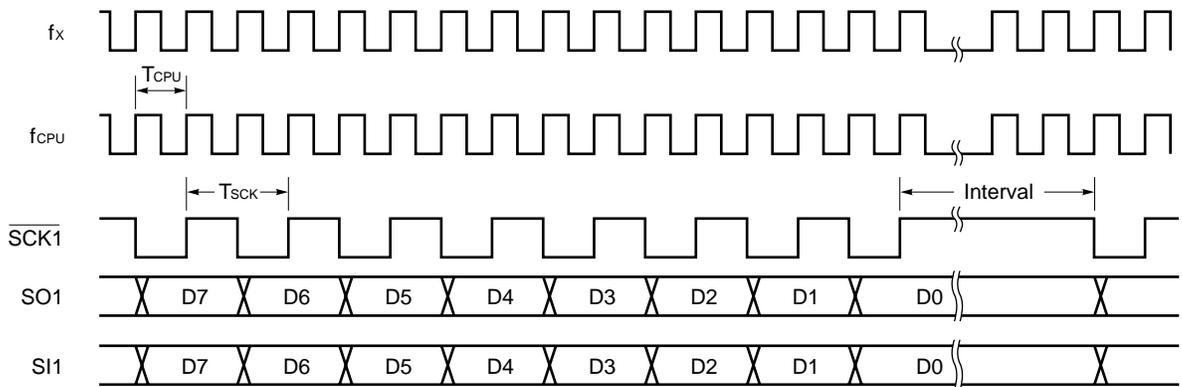
Refer to **Figure 19-5. Automatic Data Transmit/Receive Interval Specify Register Format** for the intervals which are set by the ADTI.

Table 19-2. Interval Timing through CPU Processing (when the Internal Clock is Operating)

CPU Processing	Interval Time
When using multiplication instruction	Max. (2.5T _{SCK} , 13T _{CPU})
When using division instruction	Max. (2.5T _{SCK} , 20T _{CPU})
External access 1 wait mode	Max. (2.5T _{SCK} , 9T _{CPU})
Other than above	Max. (2.5T _{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 to 2 (PCC0 to PCC2) of the processor clock control register (PCC) and bit 0 (MCS) of the oscillation mode selection register (OSMS))
- MAX. (a, b) : a or b, whichever is greater

Figure 19-24. Operation Timing with Automatic Data Transmit/Receive Function Performed by Internal Clock



- f_x : Main system clock oscillation frequency
- f_{CPU} : CPU clock (set by bits 0 to 2 (PCC0 to PCC2) of the processor clock control register (PCC) and bit 0 (MCS) of the oscillation mode selection register (OSMS)).
- T_{CPU} : 1/f_{CPU}
- T_{SCK} : 1/f_{SCK}
- f_{SCK} : Serial clock frequency

(b) When the automatic transmit/receive function is used by the external clock

If bit 1 (CSIM11) of serial operating mode register 1 (CSIM1) is cleared to 0, external clock operation is set.

When the automatic transmit/receive function is used by the external clock, it must be selected so that the interval may be longer than the values shown as follows.

Table 19-3. Interval Timing through CPU Processing (when the External Clock is Operating)

CPU Processing	Interval Time
When using multiplication instruction	$13T_{\text{CPU}}$ or longer
When using division instruction	$20T_{\text{CPU}}$ or longer
External access 1 wait mode	$9T_{\text{CPU}}$ or longer
Other than above	$7T_{\text{CPU}}$ or longer

T_{CPU} : $1/f_{\text{CPU}}$

f_{CPU} : CPU clock (set by the bits 0 to 2 (PCC0 to PCC2) of the processor clock control register (PCC) and bit 0 (MCS) of the oscillation mode selection register (OSMS))

20.1 Serial Interface Channel 2 Functions

Serial interface channel 2 has the following three modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode
- 3-wire serial I/O mode

(1) Operation stop mode

This mode is used when serial transfer is not carried out to reduce power consumption.

(2) Asynchronous serial interface (UART) mode

In this mode, one byte of data is transmitted/received following the start bit, and full-duplex operation is possible.

A dedicated UART baud rate generator is incorporated, allowing communication over a wide range of baud rates. In addition, the baud rate can be defined by scaling the input clock to the ASCK pin.

The MIDI standard baud rate (31.25 kbps) can be used by employing the dedicated UART baud rate generator.

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

In this mode, 8-bit data transfer is performed using three lines: the serial clock ($\overline{\text{SCK2}}$), and serial data lines (SI2, SO2).

In the 3-wire serial I/O mode, simultaneous transmission and reception is possible, increasing the data transfer processing speed.

Either the MSB or LSB can be specified as the start bit for an 8-bit data serial transfer, allowing connection to devices using either as the start bit.

The 3-wire serial I/O mode is useful for connection to peripheral I/Os and display controllers, etc., which incorporate a conventional synchronous clocked serial interface, such as the 75X/XL Series, 78K Series, 17K Series, etc.

20.2 Serial Interface Channel 2 Configuration

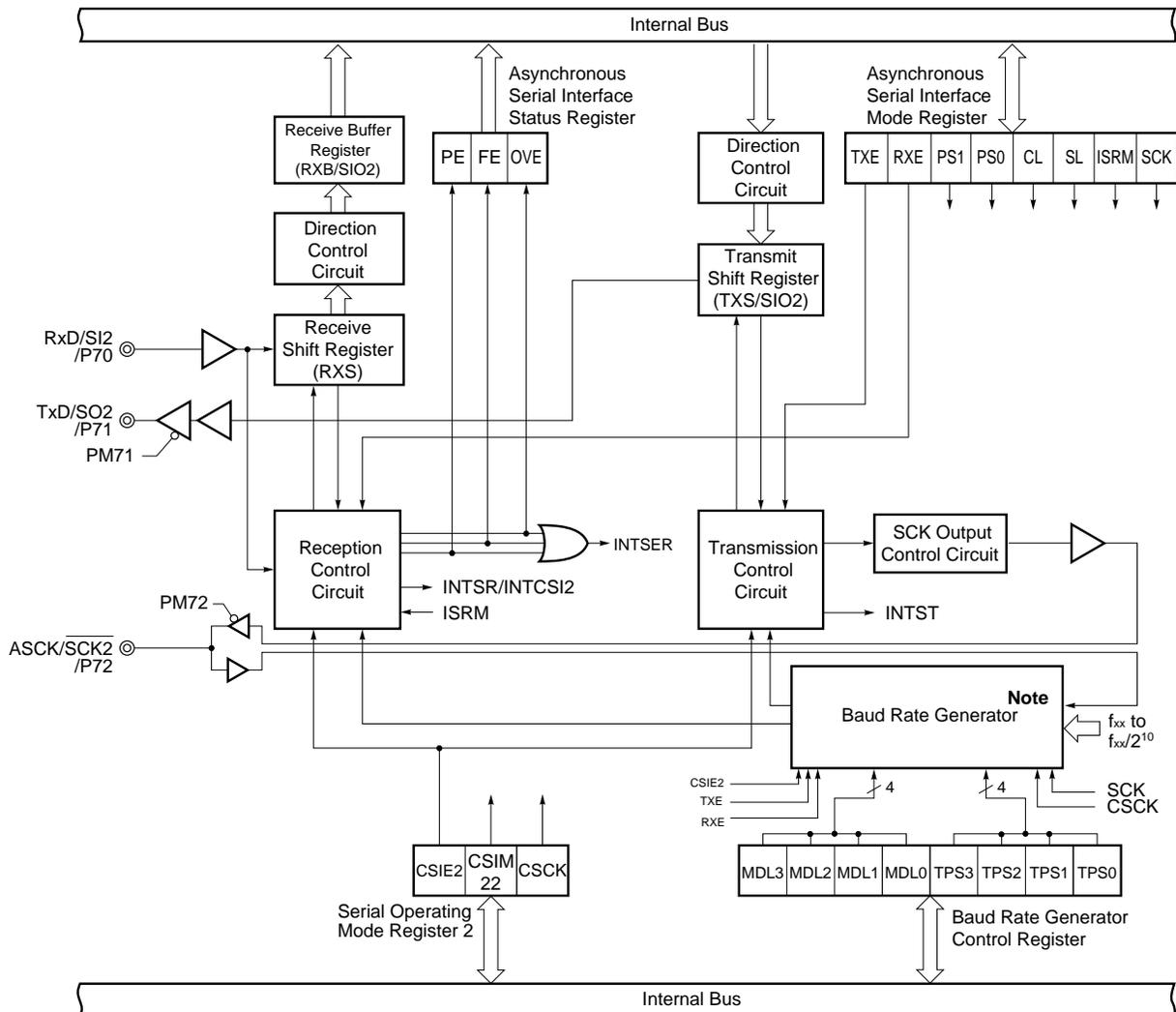
Serial interface channel 2 consists of the following hardware.

Table 20-1. Serial Interface Channel 2 Configuration

Item	Configuration
Register	Transmit shift register (TXS) Receive shift register (RXS) Receive buffer register (RXB)
Control register	Serial operating mode register 2 (CSIM2) Asynchronous serial interface mode register (ASIM) Asynchronous serial interface status register (ASIS) Baud rate generator control register (BRGC) Port mode register 7 (PM7) ^{Note}

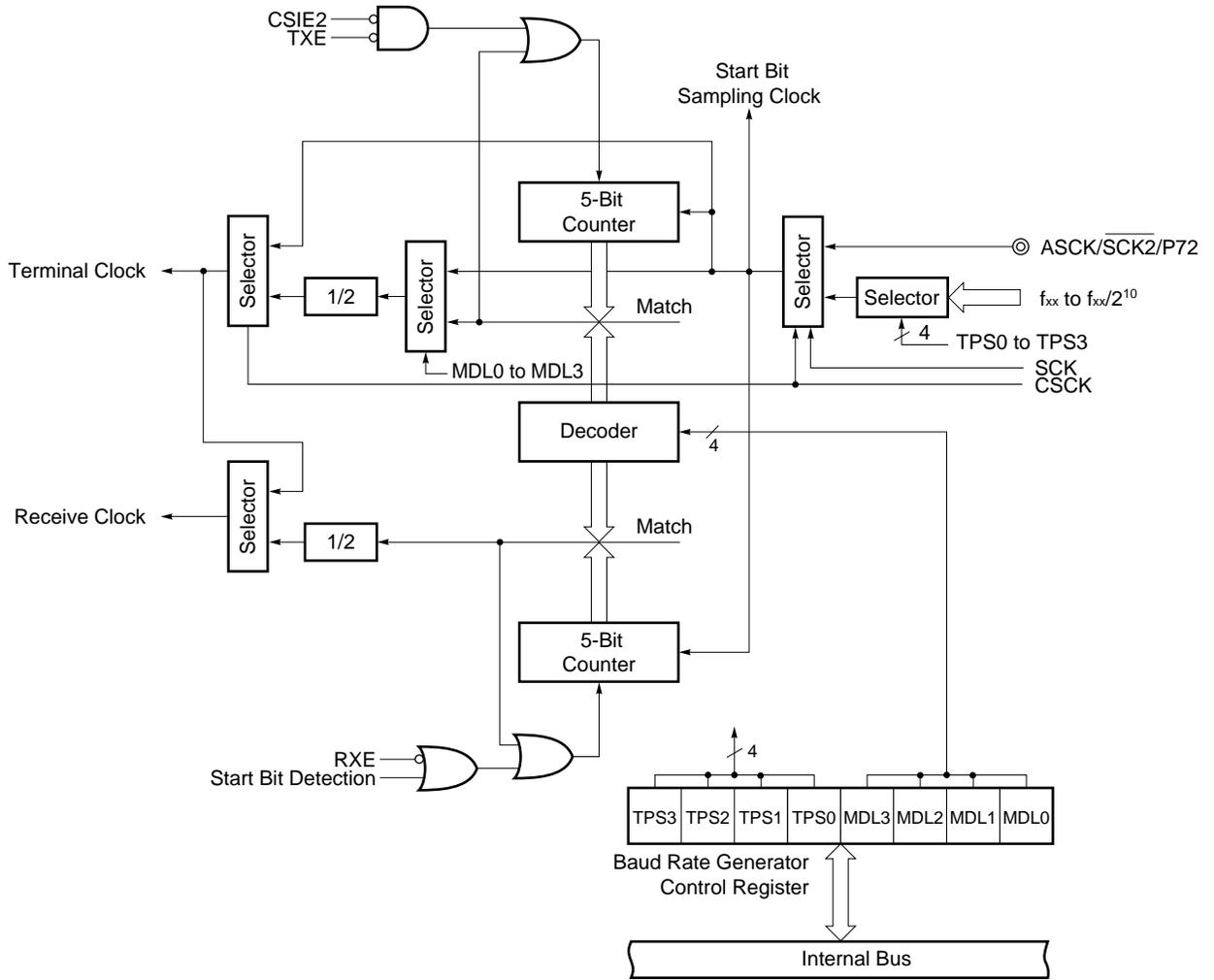
Note Refer to **Figure 6-15 Block Diagram of P70** and **Figure 6-16 Block Diagram of P71 and P72**.

Figure 20-1. Serial Interface Channel 2 Block Diagram



Note See **Figure 20-2** for the baud rate generator configuration.

Figure 20-2. Baud Rate Generator Block Diagram



(1) Transmit shift register (TXS)

This register is used to set the transmit data. The data written in TXS is transmitted as serial data. If the data length is specified as 7 bits, bits 0 to 6 of the data written in TXS are transferred as transmit data. Writing data to TXS starts the transmit operation. TXS is written to with an 8-bit memory manipulation instruction. It cannot be read. TXS value is FFH after $\overline{\text{RESET}}$ input.

Caution TXS must not be written to during a transmit operation. TXS and the receive buffer register (RXB) are allocated to the same address, and when a read is performed, the value of RXB is read.

(2) Receive shift register (RXS)

This register is used to convert serial data input to the RxD pin to parallel data. When one byte of data is received, the receive data is transferred to the receive buffer register (RXB). RXS cannot be directly manipulated by a program.

(3) Receive buffer register (RXB)

This register holds receive data. Each time one byte of data is received, new receive data is transferred from the receive shift register (RXS). If the data length is specified as 7 bits, the receive data is transferred to bits 0 to 6 of RXB, and the MSB of RXB is always set to 0. RXB is read with an 8-bit memory manipulation instruction. It cannot be written to. RXB value is FFH after $\overline{\text{RESET}}$ input.

Caution RXB and the transmit shift register (TXS) are allocated to the same address, and when a write is performed, the value is written to TXS.

(4) Transmission control circuit

This circuit performs transmit operation control such as the addition of a start bit, parity bit and stop bit to data written in the transmit shift register (TXS) in accordance with the contents set in the asynchronous serial interface mode register (ASIM).

(5) Reception control circuit

This circuit controls receive operations in accordance with the contents set in the asynchronous serial interface mode register (ASIM). It performs error checks for parity errors, etc., during a receive operation, and if an error is detected, sets a value in the asynchronous serial interface status register (ASIS) in accordance with the error contents.

20.3 Serial Interface Channel 2 Control Registers

Serial interface channel 2 is controlled by the following four registers.

- Serial Operating Mode Register 2 (CSIM2)
- Asynchronous Serial Interface Mode Register (ASIM)
- Asynchronous Serial Interface Status Register (ASIS)
- Baud Rate Generator Control Register (BRGC)

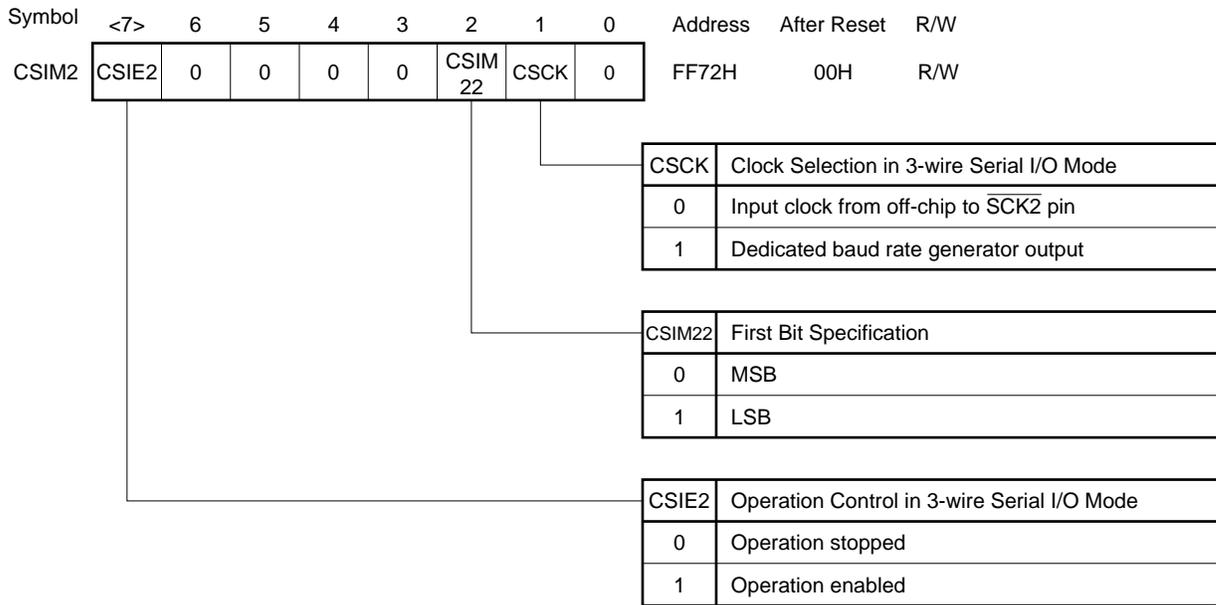
(1) Serial operating mode register 2 (CSIM2)

This register is set when serial interface channel 2 is used in the 3-wire serial I/O mode.

CSIM2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM2 to 00H.

Figure 20-3. Serial Operating Mode Register 2 Format



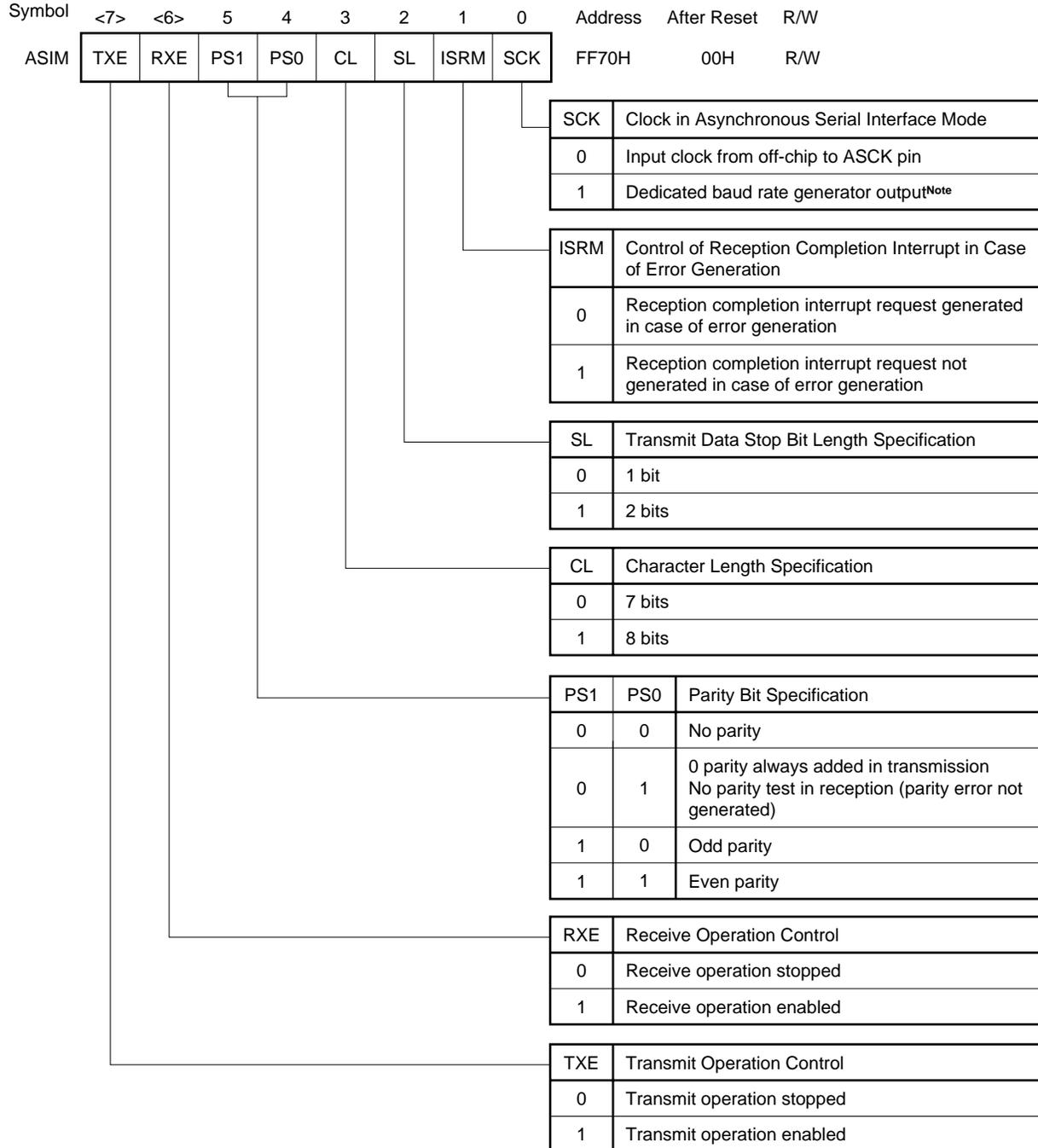
Cautions 1. Ensure that bits 0 and 3 to 6 are set to 0.

2. When UART mode is selected, CSIM2 should be set to 00H.

(2) Asynchronous serial interface mode register (ASIM)

This register is set when serial interface channel 2 is used in the asynchronous serial interface mode. ASIM is set with a 1-bit or 8-bit memory manipulation instruction. $\overline{\text{RESET}}$ input sets ASIM to 00H.

Figure 20-4. Asynchronous Serial Interface Mode Register Format



Note When SCK is set to 1 and the baud rate generator output is selected, the ASCK pin can be used as an input/output port.

- Cautions**
1. When the 3-wire serial I/O mode is selected, 00H should be set in ASIM.
 2. The serial transmit/receive operation must be stopped before changing the operating mode.

Table 20-2. Serial Interface Channel 2 Operating Mode Settings

(1) Operation Stop Mode

ASIM			CSIM2			PM70	P70	PM71	P71	PM72	P72	Start Bit	Shift Clock	P70/SI2 /RxD Pin Functions	P71/SO2 /TxD Pin Functions	P72/ $\overline{\text{SCK2}}$ /ASCK Pin Functions
TXE	RXE	SCK	CSIE2	CSIM22	CSCK											
0	0	x	0	x	x	x ^{Note 1}	—	—	P70	P71	P72					
Other than above												Setting prohibited				

(2) 3-wire Serial I/O Mode

ASIM			CSIM2			PM70	P70	PM71	P71	PM72	P72	Start Bit	Shift Clock	P70/SI2 /RxD Pin Functions	P71/SO2 /TxD Pin Functions	P72/ $\overline{\text{SCK2}}$ /ASCK Pin Functions		
TXE	RXE	SCK	CSIE2	CSIM22	CSCK													
0	0	0	1	0	0	x ^{Note 2}	x ^{Note 2}	0	1	1	x	MSB	External clock	SI2 ^{Note 2}	SO2 (CMOS output)	$\overline{\text{SCK2}}$ input		
					1					0	1					Internal clock	$\overline{\text{SCK2}}$ output	
			1	1	0					1	x	LSB	External clock			SI2 ^{Note 2}	SO2 (CMOS output)	$\overline{\text{SCK2}}$ input
										1	0							1
Other than above												Setting prohibited						

(3) Asynchronous Serial Interface Mode

ASIM			CSIM2			PM70	P70	PM71	P71	PM72	P72	Start Bit	Shift Clock	P70/SI2 /RxD Pin Functions	P71/SO2 /TxD Pin Functions	P72/ $\overline{\text{SCK2}}$ /ASCK Pin Functions
TXE	RXE	SCK	CSIE2	CSIM22	CSCK											
1	0	0	0	0	0	x ^{Note 1}	x ^{Note 1}	0	1	1	x	LSB	External clock	P70	TxD (CMOS output)	ASCK input
		1			x ^{Note 1}											x ^{Note 1}
0	1	0	0	0	0	1	x	x ^{Note 1}	x ^{Note 1}	1	x	External clock	RxD	P71	ASCK input	
		1			x ^{Note 1}											x ^{Note 1}
1	1	0	0	0	0	1	x	0	1	1	x	External clock	Internal clock	TxD (CMOS output)	ASCK input	
		1			x ^{Note 1}										x ^{Note 1}	Internal clock
Other than above												Setting prohibited				

- Notes**
1. Can be used freely as port function.
 2. Can be used as P70 (CMOS input/output) when only transmitter is used.

Remark x : Don't care
 PMxx : Port mode register
 Pxx : Port output latch

(3) Asynchronous serial interface status register (ASIS)

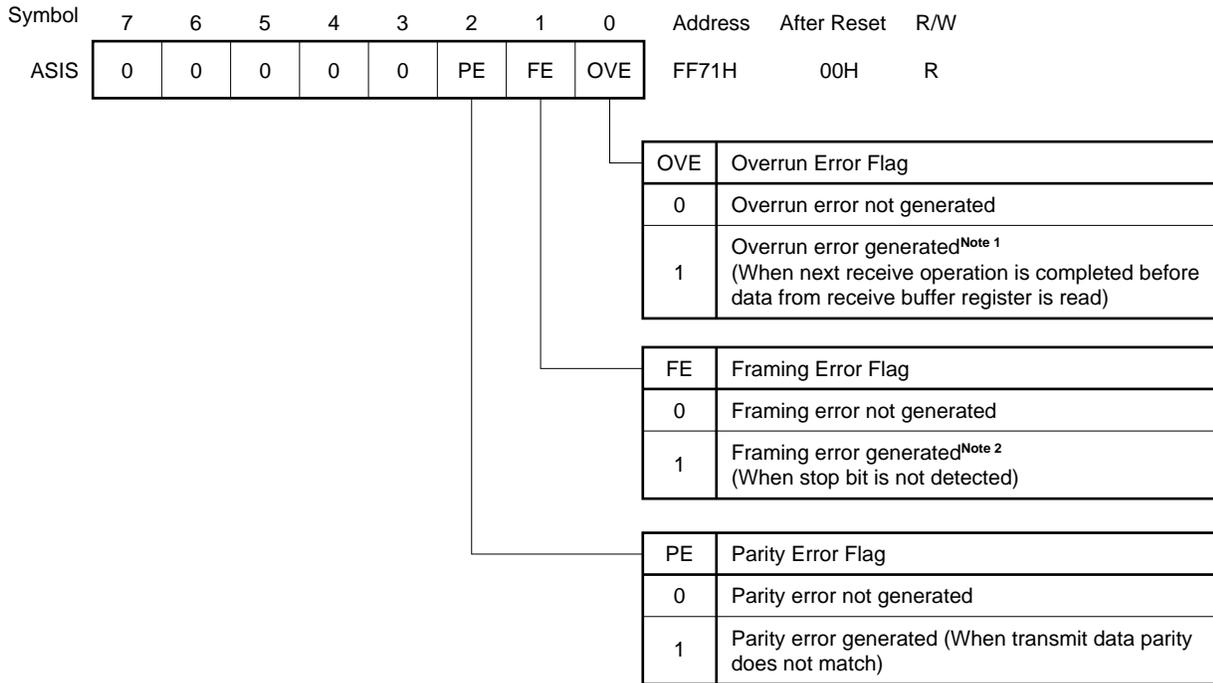
This is a register which displays the type of error when a reception error is generated in the asynchronous serial interface mode.

ASIS is read with a 1-bit or 8-bit memory manipulation instruction.

In 3-wire serial I/O mode, the contents of the ASIS are undefined.

$\overline{\text{RESET}}$ input sets ASIS to 00H.

Figure 20-5. Asynchronous Serial Interface Status Register Format



Notes 1. The receive buffer register (RXB) must be read when an overrun error is generated. Overrun errors will continue to be generated until RXB is read.

2. Even if the stop bit length has been set as 2 bits by bit 2 (SL) of the asynchronous serial interface mode register, only single stop bit detection is performed during reception.

(4) Baud rate generator control register (BRGC)

This register sets the serial clock for serial interface channel 2.

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

$\overline{\text{RESET}}$ input sets BRGC to 00H.

Figure 20-6. Baud Rate Generator Control Register Format (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
BRGC	TPS3	TPS2	TPS1	TPS0	MDL3	MDL2	MDL1	MDL0	FF73H	00H	R/W

MDL3	MDL2	MDL1	MDL0	Baud Rate Generator Input Clock Selection	k
0	0	0	0	f _{sck} /16	0
0	0	0	1	f _{sck} /17	1
0	0	1	0	f _{sck} /18	2
0	0	1	1	f _{sck} /19	3
0	1	0	0	f _{sck} /20	4
0	1	0	1	f _{sck} /21	5
0	1	1	0	f _{sck} /22	6
0	1	1	1	f _{sck} /23	7
1	0	0	0	f _{sck} /24	8
1	0	0	1	f _{sck} /25	9
1	0	1	0	f _{sck} /26	10
1	0	1	1	f _{sck} /27	11
1	1	0	0	f _{sck} /28	12
1	1	0	1	f _{sck} /29	13
1	1	1	0	f _{sck} /30	14
1	1	1	1	f _{sck} ^{Note}	—

Note Can only be used in 3-wire serial I/O mode.

- Remarks**
1. f_{sck} : 5-bit counter source clock
 2. k : Value set in MDL0 to MDL3 (0 ≤ k ≤ 14)

Figure 20-6. Baud Rate Generator Control Register Format (2/2)

TPS3	TPS2	TPS1	TPS0	5-Bit Counter Source Clock Selection				n
				MCS = 1		MCS = 0		
0	0	0	0	$f_{xx}/2^{10}$	$f_{xx}/2^{10}$ (4.9 kHz)	$f_x/2^{11}$ (2.4 kHz)		11
0	1	0	1	f_{xx}	f_x (5.0 MHz)	$f_x/2$ (2.5 MHz)		1
0	1	1	0	$f_{xx}/2$	$f_x/2$ (2.5 MHz)	$f_x/2^2$ (1.25 MHz)		2
0	1	1	1	$f_{xx}/2^2$	$f_x/2^2$ (1.25 MHz)	$f_x/2^3$ (625 kHz)		3
1	0	0	0	$f_{xx}/2^3$	$f_x/2^3$ (625 kHz)	$f_x/2^4$ (313 kHz)		4
1	0	0	1	$f_{xx}/2^4$	$f_x/2^4$ (313 kHz)	$f_x/2^5$ (156 kHz)		5
1	0	1	0	$f_{xx}/2^5$	$f_x/2^5$ (156 kHz)	$f_x/2^6$ (78.1 kHz)		6
1	0	1	1	$f_{xx}/2^6$	$f_x/2^6$ (78.1 kHz)	$f_x/2^7$ (39.1 kHz)		7
1	1	0	0	$f_{xx}/2^7$	$f_x/2^7$ (39.1 kHz)	$f_x/2^8$ (19.5 kHz)		8
1	1	0	1	$f_{xx}/2^8$	$f_x/2^8$ (19.5 kHz)	$f_x/2^9$ (9.8 kHz)		9
1	1	1	0	$f_{xx}/2^9$	$f_x/2^9$ (9.8 kHz)	$f_x/2^{10}$ (4.9 kHz)		10
Other than above				Setting prohibited				

Caution When a write is performed to BRGC during a communication operation, baud rate generator output is disrupted and communication cannot be performed normally. Therefore, BRGC must not be written to during a communication operation.

- Remarks**
1. f_x : Main system clock oscillation frequency
 2. f_{xx} : Main system clock frequency (f_x or $f_x/2$)
 3. MCS : Bit 0 of oscillation mode selection register (OSMS)
 4. n : Value set in TPS0 to TPS3 ($1 \leq n \leq 11$)
 5. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

The baud rate transmit/receive clock generated is either a signal scaled from the main system clock, or a signal scaled from the clock input from the ASCK pin.

(a) Generation of baud rate transmit/receive clock by means of main system clock

The transmit/receive clocks generated by scaling the main system clock. The baud rate generated from the main system clock is found from the following expression.

$$[\text{Baud rate}] = \frac{f_{xx}}{2^n \times (k + 16)} \text{ [Hz]}$$

- where,
- f_x : Main system clock oscillation frequency
 - f_{xx} : Main system clock frequency (f_x or $f_x/2$)
 - n : Value set in TPS0 to TPS3 ($1 \leq n \leq 11$)
 - k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

Table 20-3. Relationship between Main System Clock and Baud Rate

Baud Rate (bps)	$f_x = 5.0 \text{ MHz}$				$f_x = 4.19 \text{ MHz}$			
	MCS = 1		MCS = 0		MCS = 1		MCS = 0	
	BRGC Set Value	Error (%)	BRGC Set Value	Error (%)	BRGC Set Value	Error (%)	BRGC Set Value	Error (%)
75	—		00H	1.73	0BH	1.14	EBH	1.14
110	06H	0.88	E6H	0.88	03H	-2.01	E3H	-2.01
150	00H	1.73	E0H	1.73	EBH	1.14	DBH	1.14
300	E0H	1.73	D0H	1.73	DBH	1.14	CBH	1.14
600	D0H	1.73	C0H	1.73	CBH	1.14	BBH	1.14
1200	C0H	1.73	B0H	1.73	BBH	1.14	ABH	1.14
2400	B0H	1.73	A0H	1.73	ABH	1.14	9BH	1.14
4800	A0H	1.73	90H	1.73	9BH	1.14	8BH	1.14
9600	90H	1.73	80H	1.73	8BH	1.14	7BH	1.14
19200	80H	1.73	70H	1.73	7BH	1.14	6BH	1.14
31250	74H	0	64H	0	71H	-1.31	61H	-1.31
38400	70H	1.73	60H	1.73	6BH	1.14	5BH	1.14
76800	60H	1.73	50H	1.73	5BH	1.14	—	—

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

(b) Generation of baud rate transmit/receive clock by means of external clock from ASCK pin

The transmit/receive clock is generated by scaling the clock input from the ASCK pin. The baud rate generated from the clock input from the ASCK pin is obtained with the following expression.

$$[\text{Baud rate}] = \frac{f_{\text{ASCK}}}{2 \times (k + 16)} \text{ [Hz]}$$

f_{ASCK} : Frequency of clock input to ASCK pin

k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

Table 20-4. Relationship between ASCK Pin Input Frequency and Baud Rate (When BRGC is set to 00H)

Baud Rate (bps)	ASCK Pin Input Frequency
75	2.4 kHz
110	3.52 kHz
150	4.8 kHz
300	9.6 kHz
600	19.2 kHz
1200	38.4 kHz
2400	76.8 kHz
4800	153.6 kHz
9600	307.2 kHz
19200	614.4 kHz
31250	1000.0 kHz
38400	1228.8 kHz

20.4 Serial Interface Channel 2 Operation

Serial interface channel 2 has the following three modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode
- 3-wire serial I/O mode

20.4.1 Operation stop mode

In the operation stop mode, serial transfer is not performed, and therefore power consumption can be reduced.

In the operation stop mode, the P70/SI2/RxD, P71/SO2/TxD and P72/ $\overline{\text{SCK2}}$ /ASCK pins can be used as normal input/output ports.

(1) Register setting

Operation stop mode is set using serial operating mode register 2 (CSIM2) and the asynchronous serial interface mode register (ASIM).

(a) Serial operating mode register 2 (CSIM2)

CSIM2 is set with a 1-bit or 8-bit memory manipulation instruction.

RESET input sets CSIM2 to 00H.

Symbol	<7>	6	5	4	3	2	1	0	Address	After Reset	R/W
CSIM2	CSIE2	0	0	0	0	CSIM 22	CSCK	0	FF72H	00H	R/W

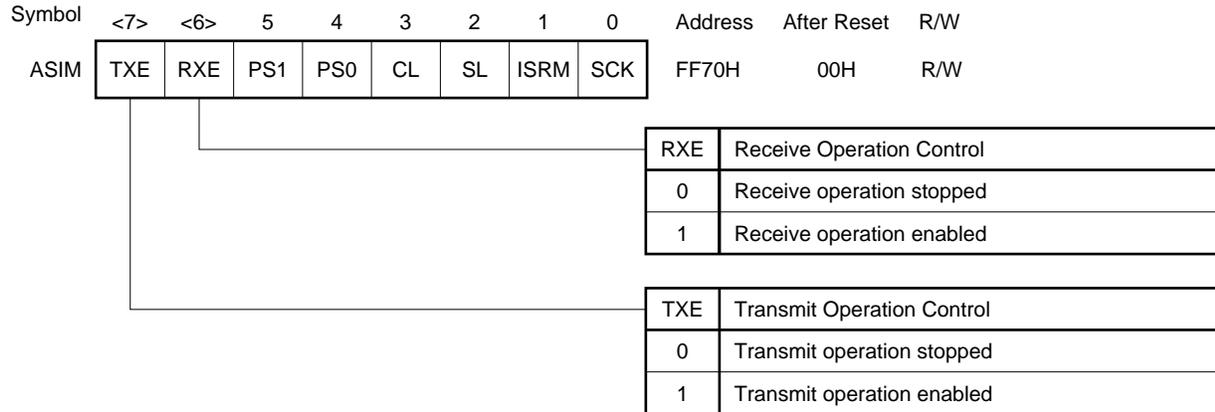
CSIE2	Operation Control in 3-wire Serial I/O Mode
0	Operation stopped
1	Operation enabled

Caution Ensure that bit 0 and bits 3 through 6 are set to 0.

(b) Asynchronous serial interface mode register (ASIM)

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

$\overline{\text{RESET}}$ input sets ASIM to 00H.



20.4.2 Asynchronous serial interface (UART) mode

In this mode, one byte of data is transmitted/received following the start bit, and full-duplex operation is possible.

A dedicated UART baud rate generator is incorporated, allowing communication over a wide range of baud rates.

In addition, the baud rate can be defined by scaling the input clock to the ASCK pin.

The MIDI standard baud rate (31.25 kbps) can be used by employing the dedicated UART baud rate generator.

(1) Register setting

UART mode is set using serial operating mode register 2 (CSIM2), the asynchronous serial interface mode register (ASIM), the asynchronous serial interface status register (ASIS), and the baud rate generator control register (BRGC).

(a) Serial operating mode register 2 (CSIM2)

CSIM2 is set with a 1-bit or 8-bit memory manipulation instruction.

RESET input sets CSIM2 to 00H.

When the UART mode is selected, 00H should be set in CSIM2.

Symbol	<7>	6	5	4	3	2	1	0	Address	After Reset	R/W
CSIM2	CSIE2	0	0	0	0	CSIM22	CSCK	0	FF72H	00H	R/W

CSCK	Clock Selection in 3-wire Serial I/O Mode
0	Input clock from off-chip to SCK2 pin
1	Dedicated baud rate generator output

CSIM22	First Bit Specification
0	MSB
1	LSB

CSIE2	Operation Control in 3-wire Serial I/O Mode
0	Operation stopped
1	Operation enabled

Caution Ensure that bit 0 and bits 3 through 6 are set to 0.

(b) Asynchronous serial interface mode register (ASIM)

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

$\overline{\text{RESET}}$ input sets ASIM to 00H.

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After Reset	R/W
ASIM	TXE	RXE	PS1	PS0	CL	SL	ISRM	SCK	FF70H	00H	R/W

SCK	Clock Selection in Asynchronous Serial Interface Mode	
0	Input clock from off-chip to ASCK pin	
1	Dedicated baud rate generator output ^{Note}	

ISRM	Control of Reception Completion Interrupt in Case of Error Generation	
0	Reception completion interrupt request generated in case of error generation	
1	Reception completion interrupt request not generated in case of error generation	

SL	Transmit Data Stop Bit Length Specification	
0	1 bit	
1	2 bits	

CL	Character Length Specification	
0	7 bits	
1	8 bits	

PS1	PS0	Parity Bit Specification
0	0	No parity
0	1	0 parity always added in transmission No parity test in reception (parity error not generated)
1	0	Odd parity
1	1	Even parity

RXE	Receive Operation Control	
0	Receive operation stopped	
1	Receive operation enabled	

TXE	Transmit Operation Control	
0	Transmit operation stopped	
1	Transmit operation enabled	

Note When SCK is set to 1 and the baud rate generator output is selected, the ASCK pin can be used as an input/output port.

Caution The serial transmit/receive operation must be stopped before changing the operating mode.

(c) Asynchronous serial interface status register (ASIS)

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

$\overline{\text{RESET}}$ input sets ASIS to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ASIS	0	0	0	0	0	PE	FE	OVE	FF71H	00H	R

OVE	Overrun Error Flag
0	Overrun error not generated
1	Overrun error generated ^{Note 1} (When next receive operation is completed before data from receive buffer register is read)

FE	Framing Error Flag
0	Framing error not generated
1	Framing error generated ^{Note 2} (When stop bit is not detected)

PE	Parity Error Flag
0	Parity error not generated
1	Parity error generated (When transmit data parity does not match)

- Notes**
1. The receive buffer register (RXB) must be read when an overrun error is generated. Overrun errors will continue to be generated until RXB is read.
 2. Even if the stop bit length has been set as 2 bits by bit 2 (SL) of the asynchronous serial interface mode register (ASIM), only single stop bit detection is performed during reception.

(d) Baud rate generator control register (BRGC)

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

$\overline{\text{RESET}}$ input sets BRGC to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
BRGC	TPS3	TPS2	TPS1	TPS0	MDL3	MDL2	MDL1	MDL0	FF73H	00H	R/W

MDL3	MDL2	MDL1	MDL0	Baud Rate Generator Input Clock Selection	k
0	0	0	0	$f_{\text{sck}}/16$	0
0	0	0	1	$f_{\text{sck}}/17$	1
0	0	1	0	$f_{\text{sck}}/18$	2
0	0	1	1	$f_{\text{sck}}/19$	3
0	1	0	0	$f_{\text{sck}}/20$	4
0	1	0	1	$f_{\text{sck}}/21$	5
0	1	1	0	$f_{\text{sck}}/22$	6
0	1	1	1	$f_{\text{sck}}/23$	7
1	0	0	0	$f_{\text{sck}}/24$	8
1	0	0	1	$f_{\text{sck}}/25$	9
1	0	1	0	$f_{\text{sck}}/26$	10
1	0	1	1	$f_{\text{sck}}/27$	11
1	1	0	0	$f_{\text{sck}}/28$	12
1	1	0	1	$f_{\text{sck}}/29$	13
1	1	1	0	$f_{\text{sck}}/30$	14

TPS3	TPS2	TPS1	TPS0	5-Bit Counter Source Clock Selection				n
				MCS = 1		MCS = 0		
0	0	0	0	$f_{\text{xx}}/2^{10}$	$f_{\text{x}}/2^{10}$ (4.9 kHz)	$f_{\text{x}}/2^{11}$ (2.4 kHz)	11	
0	1	0	1	f_{xx}	f_{x} (5.0 MHz)	$f_{\text{x}}/2$ (2.5 MHz)	1	
0	1	1	0	$f_{\text{xx}}/2$	$f_{\text{x}}/2$ (2.5 MHz)	$f_{\text{x}}/2^2$ (1.25 MHz)	2	
0	1	1	1	$f_{\text{xx}}/2^2$	$f_{\text{x}}/2^2$ (1.25 MHz)	$f_{\text{x}}/2^3$ (625 kHz)	3	
1	0	0	0	$f_{\text{xx}}/2^3$	$f_{\text{x}}/2^3$ (625 kHz)	$f_{\text{x}}/2^4$ (313 kHz)	4	
1	0	0	1	$f_{\text{xx}}/2^4$	$f_{\text{x}}/2^4$ (313 kHz)	$f_{\text{x}}/2^5$ (156 kHz)	5	
1	0	1	0	$f_{\text{xx}}/2^5$	$f_{\text{x}}/2^5$ (156 kHz)	$f_{\text{x}}/2^6$ (78.1 kHz)	6	
1	0	1	1	$f_{\text{xx}}/2^6$	$f_{\text{x}}/2^6$ (78.1 kHz)	$f_{\text{x}}/2^7$ (39.1 kHz)	7	
1	1	0	0	$f_{\text{xx}}/2^7$	$f_{\text{x}}/2^7$ (39.1 kHz)	$f_{\text{x}}/2^8$ (19.5 kHz)	8	
1	1	0	1	$f_{\text{xx}}/2^8$	$f_{\text{x}}/2^8$ (19.5 kHz)	$f_{\text{x}}/2^9$ (9.8 kHz)	9	
1	1	1	0	$f_{\text{xx}}/2^9$	$f_{\text{x}}/2^9$ (9.8 kHz)	$f_{\text{x}}/2^{10}$ (4.9 kHz)	10	
Other than above				Setting prohibited				

Caution When a write is performed to BRGC during a communication operation, baud rate generator output is disrupted and communication cannot be performed normally. Therefore, BRGC must not be written to during a communication operation.

- Remarks**
1. f_{SCK} : 5-bit counter source clock
 2. k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)
 3. f_x : Main system clock oscillation frequency
 4. f_{XX} : Main system clock frequency (f_x or $f_x/2$)
 5. MCS : Bit 0 of oscillation mode selection register (OSMS)
 6. n : Value set in TPS0 to TPS3 ($1 \leq n \leq 11$)
 7. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

The baud rate transmit/receive clock generated is either a signal scaled from the main system clock, or a signal scaled from the clock input from the ASCK pin.

(i) Generation of baud rate transmit/receive clock by means of main system clock

The transmit/receive clock is generated by scaling the main system clock. The baud rate generated from the main system clock is obtained with the following expression.

$$[\text{Baud rate}] = \frac{f_{xx}}{2^n \times (k + 16)} \text{ [Hz]}$$

- where, f_x : Main system clock oscillation frequency
- f_{xx} : Main system clock frequency (f_x or $f_x/2$)
- n : Value set in TPS0 to TPS3 ($1 \leq n \leq 11$)
- k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

Table 20-5. Relationship between Main System Clock and Baud Rate

Baud Rate (bps)	fx = 5.0 MHz				fx = 4.19 MHz			
	MCS = 1		MCS = 0		MCS = 1		MCS = 0	
	BRGC Set Value	Error (%)						
75	—		00H	1.73	0BH	1.14	EBH	1.14
110	06H	0.88	E6H	0.88	03H	-2.01	E3H	-2.01
150	00H	1.73	E0H	1.73	EBH	1.14	DBH	1.14
300	E0H	1.73	D0H	1.73	DBH	1.14	CBH	1.14
600	D0H	1.73	C0H	1.73	CBH	1.14	BBH	1.14
1200	C0H	1.73	B0H	1.73	BBH	1.14	ABH	1.14
2400	B0H	1.73	A0H	1.73	ABH	1.14	9BH	1.14
4800	A0H	1.73	90H	1.73	9BH	1.14	8BH	1.14
9600	90H	1.73	80H	1.73	8BH	1.14	7BH	1.14
19200	80H	1.73	70H	1.73	7BH	1.14	6BH	1.14
31250	74H	0	64H	0	71H	-1.31	61H	-1.31
38400	70H	1.73	60H	1.73	6BH	1.14	5BH	1.14
76800	60H	1.73	50H	1.73	5BH	1.14	—	—

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

(ii) Generation of baud rate transmit/receive clock by means of external clock from ASCK pin

The transmit/receive clock is generated by scaling the clock input from the ASCK pin. The baud rate generated from the clock input from the ASCK pin is obtained with the following expression.

$$[\text{Baud rate}] = \frac{f_{\text{ASCK}}}{2^n \times (k + 16)} \text{ [Hz]}$$

where, f_{ASCK} : Frequency of clock input to ASCK pin

k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

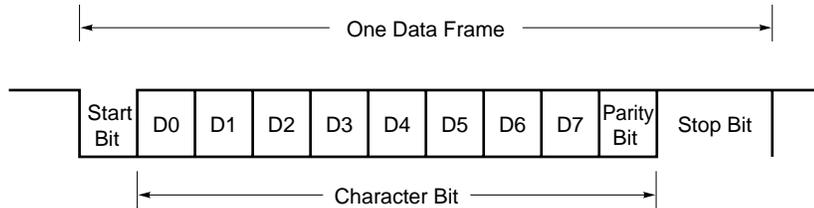
Table 20-6. Relationship between ASCK Pin Input Frequency and Baud Rate (When BRGC is set to 00H)

Baud Rate (bps)	ASCK Pin Input Frequency
75	2.4 kHz
110	3.52 kHz
150	4.8 kHz
300	9.6 kHz
600	19.2 kHz
1200	38.4 kHz
2400	76.8 kHz
4800	153.6 kHz
9600	307.2 kHz
19200	614.4 kHz
31250	1000.0 kHz
38400	1228.8 kHz

(2) Communication operation**(a) Data format**

The transmit/receive data format is shown in Figure 20-7.

Figure 20-7. Asynchronous Serial Interface Transmit/Receive Data Format



One data frame consists of the following bits.

- Start bit 1 bit
- Character bits 7 bits/8 bits
- Parity bit Even parity/odd parity/0 parity/no parity
- Stop bit(s) 1 bit/2 bits

The specification of character bit length, parity selection, and specification of stop bit length for each data frame is carried out with asynchronous serial interface mode register (ASIM).

When 7 bits are selected as the number of character bits, only the lower 7 bits (bits 0 to 6) are valid; in transmission the most significant bit (bit 7) is ignored, and in reception the most significant bit (bit 7) is always "0".

The serial transfer rate is selected by ASIM and the baud rate generator control register (BRGC).

If a serial data receive error is generated, the receive error contents can be determined by reading the status of the asynchronous serial interface status register (ASIS).

(b) Parity types and operation

The parity bit is used to detect a bit error in the communication data. Normally, the same kind of parity bit is used on the transmitting side and the receiving side. With even parity and odd parity, a one-bit (odd number) error can be detected. With 0 parity and no parity, an error cannot be detected.

(i) Even parity**• At transmission**

Control is executed so that the number of bits with a value of "1" contained in the transmit data including parity bit is an even number. The parity bit value should be as follows.

The number of bits with a value of "1" is an odd number in transmit data : 1

The number of bits with a value of "1" is an even number in transmit data : 0

• At reception

The number of bits with a value of "1" contained in the receive data including parity bit are counted, and if this is an odd number, a parity error is generated.

(ii) Odd parity**• At transmission**

Conversely to the situation with even parity, control is executed so that the number of bits with a value of "1" contained in the transmit data including parity bit is an odd number. The parity bit value should be as follows.

The number of bits with a value of "1" is an odd number in transmit data : 0

The number of bits with a value of "1" is an even number in transmit data : 1

• At reception

The number of bits with a value of "1" contained in the receive data including parity bit are counted, and if this is an even number, a parity error is generated.

(iii) 0 Parity

When transmitting, the parity bit is set to "0" irrespective of the transmit data.

At reception, a parity bit check is not performed. Therefore, a parity error is not generated, irrespective of whether the parity bit is set to "0" or "1".

(iv) No parity

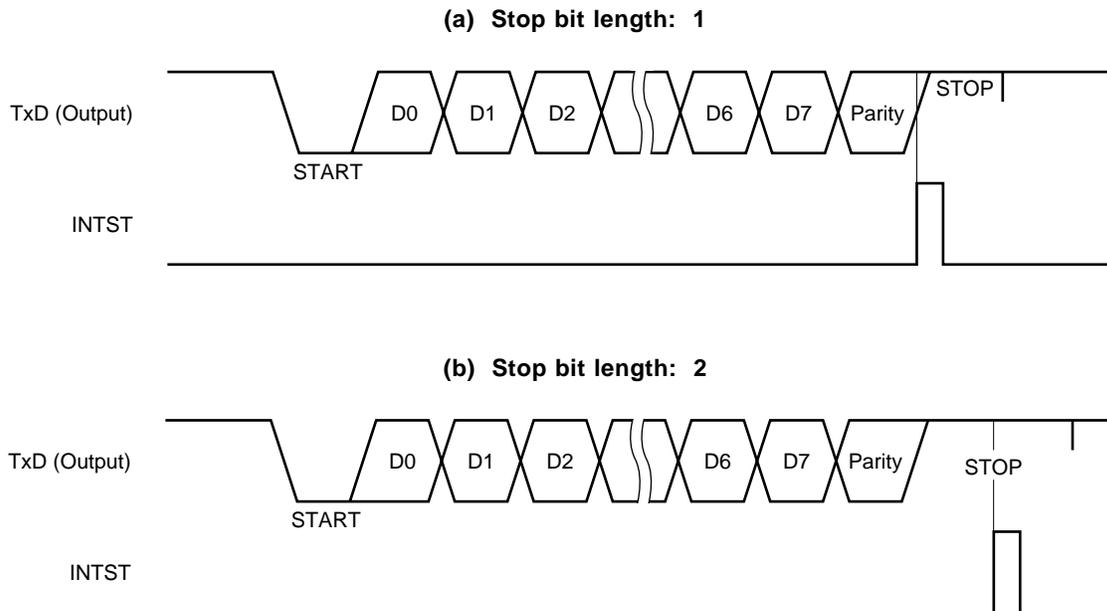
A parity bit is not added to the transmit data. At reception, data is received assuming that there is no parity bit. Since there is no parity bit, a parity error is not generated.

(c) Transmission

A transmit operation is started by writing transmit data to the transmit shift register (TXS). The start bit, parity bit, and stop bit(s) are added automatically.

When the transmit operation starts, the data in the transmit shift register (TXS) is shifted out, and when the transmit shift register (TXS) is empty, a transmission completion interrupt request (INTST) is generated.

Figure 20-8. Asynchronous Serial Interface Transmission Completion Interrupt Request Generation Timing



Caution The asynchronous serial interface mode register (ASIM) should not be rewritten during a transmit operation. If the ASIM register is rewritten during transmission, subsequent transmit operations may not be possible (the normal state is restored by $\overline{\text{RESET}}$ input).

It is possible to determine whether transmission is in progress by software by using a transmission completion interrupt request (INTST) or the interrupt request flag (STIF) set by the INTST.

(d) Reception

When the RXE bit of the asynchronous serial interface mode register (ASIM) is set (1), a receive operation is enabled and sampling of the RxD pin input is performed.

RxD pin input sampling is performed using the serial clock specified by ASIM.

When the RxD pin input becomes low, the 5-bit counter of baud rate generator (see **Figure 20-2**) starts counting, and at the time when the half time determined by specified baud rate has passed, the data sampling start timing signal is output. If the RxD pin input sampled again as a result of this start timing signal is low, it is identified as a start bit, the 5-bit counter is initialized and starts counting, and data sampling is performed. When character data, a parity bit and one stop bit are detected after the start bit, reception of one frame of data ends.

When one frame of data has been received, the receive data in the shift register is transferred to the receive buffer register (RXB), and a reception completion interrupt request (INTSR) is generated.

★

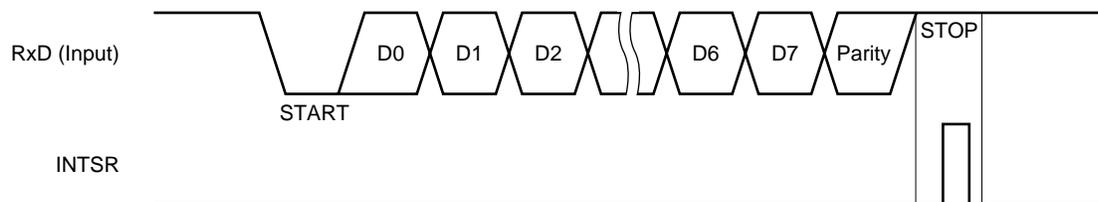
If an error is generated, the receive data in which the error was generated is still transferred to RXB. When an error is generated, if bit 1 (ISRM) of ASIM is cleared to 0, INTSR is generated.

If ISRM bit is set to 1, INTSR is not generated.

If the RXE bit is reset (0) during the receive operation, the receive operation is stopped immediately.

In this case, the contents of RXB and ASIS are not changed, and INTSR and INTSER are not generated.

Figure 20-9. Asynchronous Serial Interface Reception Completion Interrupt Request Generation Timing



Caution The receive buffer register (RXB) must be read even if a receive error is generated. If RXB is not read, an overrun error will be generated when the next data is received, and the receive error state will continue indefinitely.

(e) Receive errors

Three kinds of errors can occur during a receive operation: a parity error, framing error, or overrun error.

★ If the data reception result error flag is set in the asynchronous serial interface status register (ASIS), a receive error interrupt request (INTSER) is generated. A receive error interrupt is generated prior to a receive completion interrupt request (INTSR). Receive error causes are shown in Table 20-7.

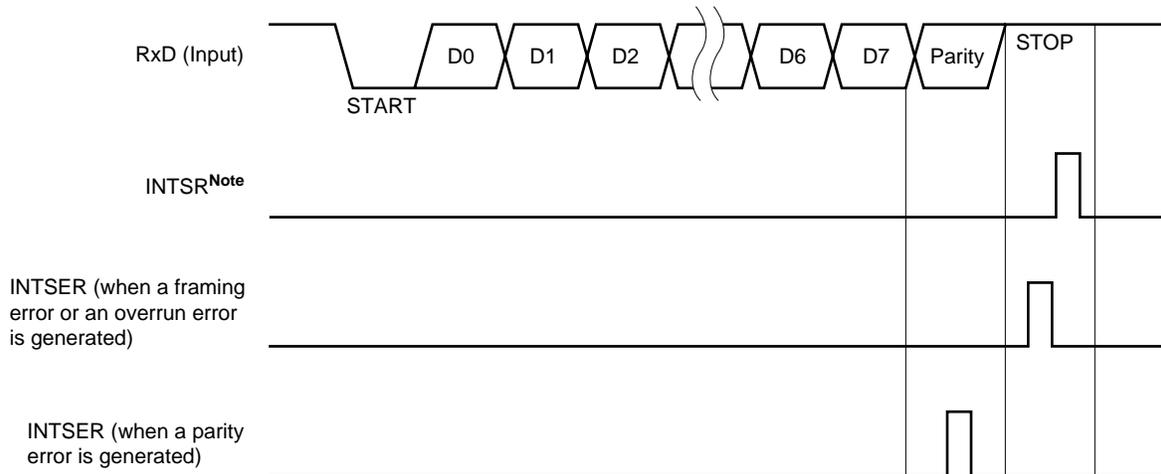
It is possible to determine what kind of error was generated during reception by reading the contents of ASIS in the reception error interrupt servicing (see **Figures 20-9** and **20-10**).

The contents of ASIS are reset (0) by reading the receive buffer register (RXB) or receiving the next data (if there is an error in the next data, the corresponding error flag is set).

Table 20-7. Receive Error Causes

Receive Errors	Cause
Parity error	Transmission-time parity specification and reception data parity do not match
Framing error	Stop bit not detected
Overrun error	Reception of next data is completed before data is read from receive register buffer

★ **Figure 20-10. Receive Error Timing**



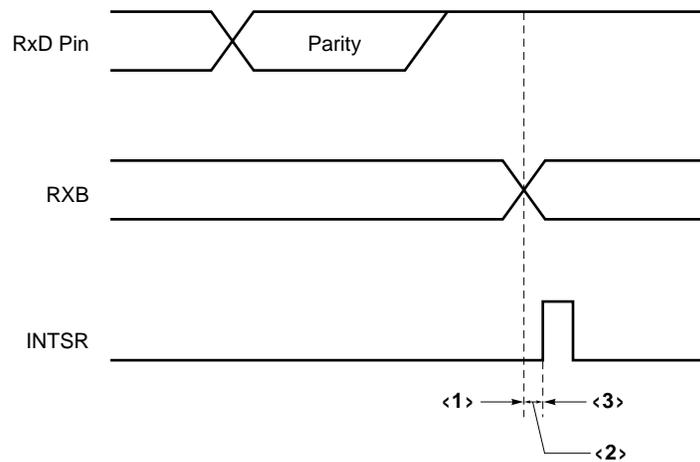
★ **Note** If a receive error is generated while bit 1 (ISRM) of the asynchronous serial interface mode register (ASIM) is set to (1), INTSR is not generated.

- Cautions**
- 1. The contents of the asynchronous serial interface status register (ASIS) are reset (0) by reading the receive buffer register (RXB) or receiving the next data. To ascertain the error contents, ASIS must be read before reading RXB.**
 - 2. The receive buffer register (RXB) must be read even if a receive error is generated. If RXB is not read, an overrun error will be generated when the next data is received, and the receive error state will continue indefinitely.**

(3) UART mode cautions

- (a) When transmit operation is stopped by clearing (0) bit 7 (TXE) of the asynchronous serial interface mode register (ASIM) during transmission, be sure to set the transmit shift register (TXS) to FFH, then set the TXE to 1, before executing the next transmission.
- (b) When receive operation is stopped by clearing (0) bit 6 (RXE) of the asynchronous serial interface mode register (ASIM) during reception, the state of the receive buffer register (RXB) and whether a receive completion interrupt request (INTSR) is generated or not differ depending on the receive stop timing. Figure 20-11 shows the timing.

Figure 20-11. State of Receive Buffer Register (RXB) When Receive Operation is Stopped and Whether Interrupt Request (INTSR) is Generated or Not



When RXE is set to 0 at a time indicated by <1>, RXB holds the previous data and does not generate INTSR.
 When RXE is set to 0 at a time indicated by <2>, RXB renews the data and does not generate INTSR.
 When RXE is set to 0 at a time indicated by <3>, RXB renews the data and generates INTSR.

20.4.3 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connection of peripheral I/Os and display controllers, etc., which incorporate a conventional synchronous clocked serial interface, such as the 75X/XL Series, 78K Series, 17K Series, etc.

Communication is performed using three lines: the serial clock ($\overline{SCK2}$), serial output (SO2), and serial input (SI2).

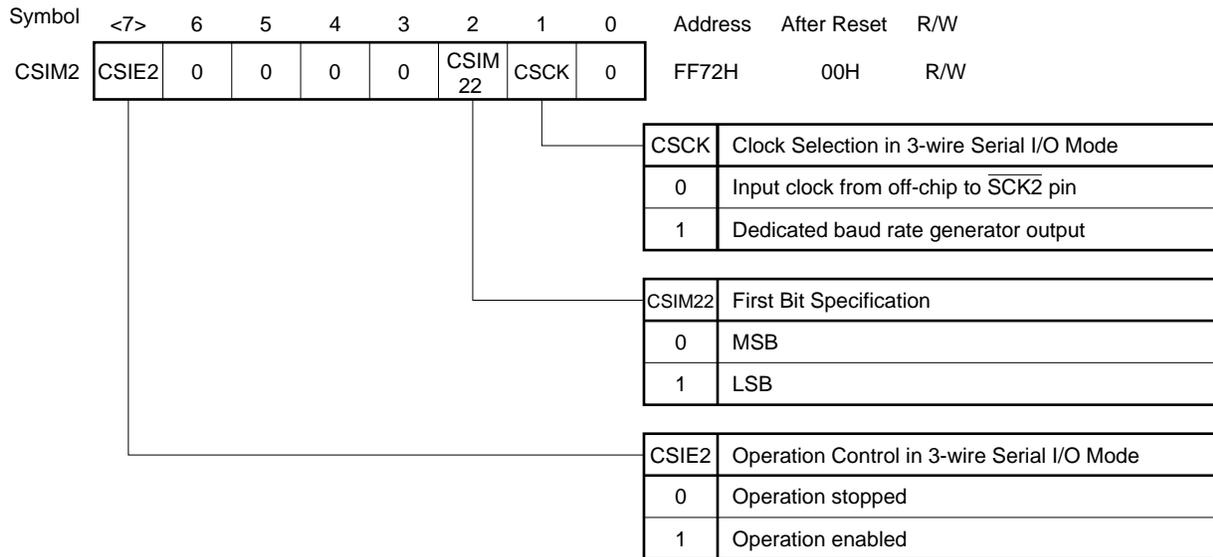
(1) Register setting

3-wire serial I/O mode is set using serial operating mode register 2 (CSIM2), the asynchronous serial interface mode register (ASIM), and the baud rate generator control register (BRGC).

(a) Serial operating mode register 2 (CSIM2)

CSIM2 is set with a 1-bit or 8-bit memory manipulation instruction.

\overline{RESET} input sets CSIM2 to 00H.



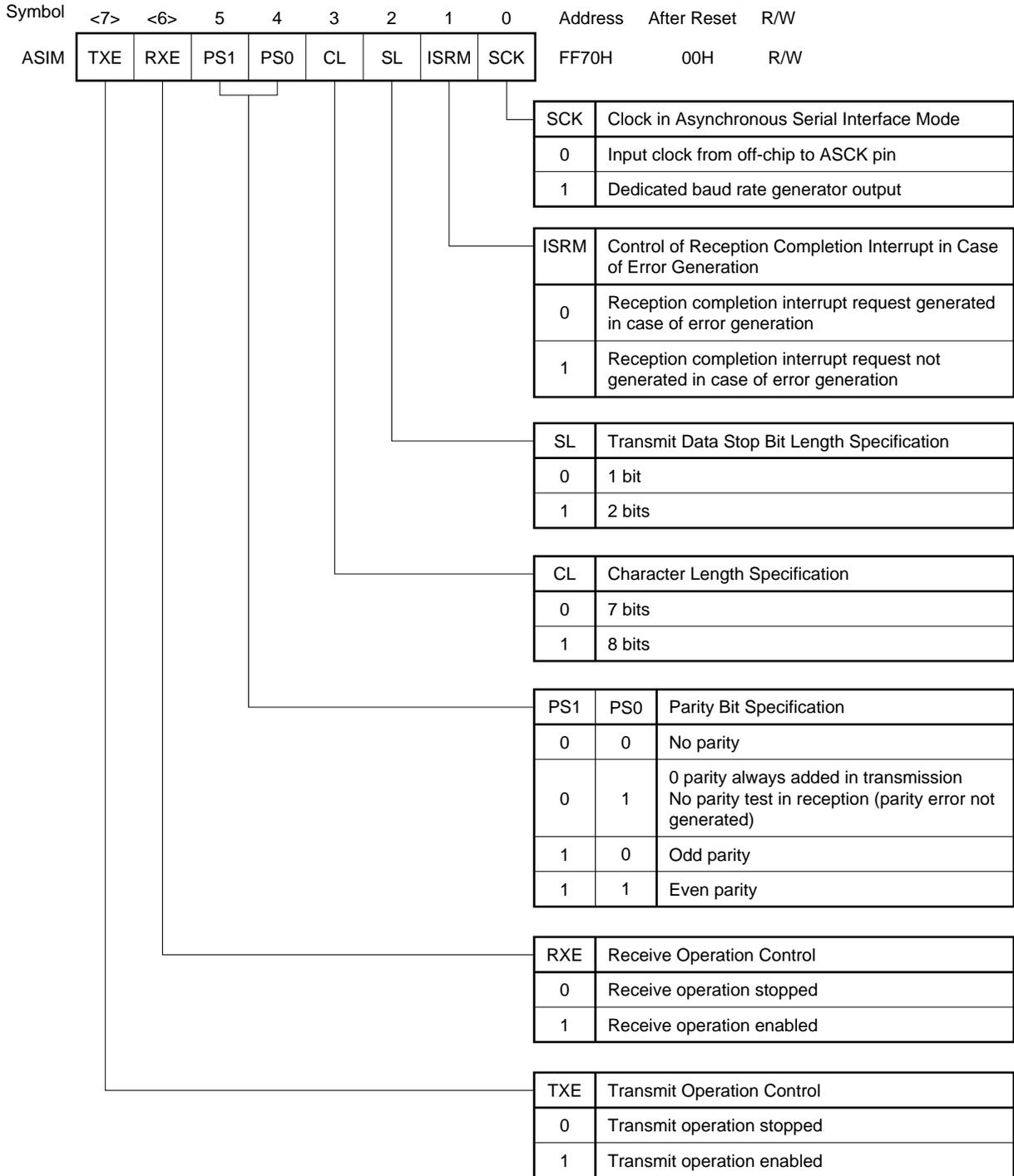
Caution Ensure that bit 0 and bits 3 through 6 are set to 0.

(b) Asynchronous serial interface mode register (ASIM)

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

$\overline{\text{RESET}}$ input sets ASIM to 00H.

When the 3-wire serial I/O mode is selected, 00H should be set in ASIM.



(c) Baud rate generator control register (BRGC)

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

$\overline{\text{RESET}}$ input sets BRGC to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
BRGC	TPS3	TPS2	TPS1	TPS0	MDL3	MDL2	MDL1	MDL0	FF73H	00H	R/W

MDL3	MDL2	MDL1	MDL0	Baud Rate Generator Input Clock Selection	k
0	0	0	0	f _{sck} /16	0
0	0	0	1	f _{sck} /17	1
0	0	1	0	f _{sck} /18	2
0	0	1	1	f _{sck} /19	3
0	1	0	0	f _{sck} /20	4
0	1	0	1	f _{sck} /21	5
0	1	1	0	f _{sck} /22	6
0	1	1	1	f _{sck} /23	7
1	0	0	0	f _{sck} /24	8
1	0	0	1	f _{sck} /25	9
1	0	1	0	f _{sck} /26	10
1	0	1	1	f _{sck} /27	11
1	1	0	0	f _{sck} /28	12
1	1	0	1	f _{sck} /29	13
1	1	1	0	f _{sck} /30	14
1	1	1	1	f _{sck}	—

f_{sck} : 5-bit counter source clock

k : Value set in MDL0 to MDL3 (0 ≤ k ≤ 14)

TPS3	TPS2	TPS1	TPS0	5-Bit Counter Source Clock Selection				n
				MCS = 1		MCS = 0		
0	0	0	0	$f_{xx}/2^{10}$	$f_x/2^{10}$ (4.9 kHz)	$f_x/2^{11}$ (2.4 kHz)		11
0	1	0	1	f_{xx}	f_x (5.0 MHz)	$f_x/2$ (2.5 MHz)		1
0	1	1	0	$f_{xx}/2$	$f_x/2$ (2.5 MHz)	$f_x/2^2$ (1.25 MHz)		2
0	1	1	1	$f_{xx}/2^2$	$f_x/2^2$ (1.25 MHz)	$f_x/2^3$ (625 kHz)		3
1	0	0	0	$f_{xx}/2^3$	$f_x/2^3$ (625 kHz)	$f_x/2^4$ (313 kHz)		4
1	0	0	1	$f_{xx}/2^4$	$f_x/2^4$ (313 kHz)	$f_x/2^5$ (156 kHz)		5
1	0	1	0	$f_{xx}/2^5$	$f_x/2^5$ (156 kHz)	$f_x/2^6$ (78.1 kHz)		6
1	0	1	1	$f_{xx}/2^6$	$f_x/2^6$ (78.1 kHz)	$f_x/2^7$ (39.1 kHz)		7
1	1	0	0	$f_{xx}/2^7$	$f_x/2^7$ (39.1 kHz)	$f_x/2^8$ (19.5 kHz)		8
1	1	0	1	$f_{xx}/2^8$	$f_x/2^8$ (19.5 kHz)	$f_x/2^9$ (9.8 kHz)		9
1	1	1	0	$f_{xx}/2^9$	$f_x/2^9$ (9.8 kHz)	$f_x/2^{10}$ (4.9 kHz)		10
Other than above				Setting prohibited				

Caution When a write is performed to BRGC during a communication operation, baud rate generator output is disrupted and communication cannot be performed normally. Therefore, BRGC must not be written to during a communication operation.

- Remarks**
1. f_x : Main system clock oscillation frequency
 2. f_{xx} : Main system clock frequency (f_x or $f_x/2$)
 3. MCS : Bit 0 of oscillation mode selection register (OSMS)
 4. n : Value set in TPS0 to TPS3 ($1 \leq n \leq 11$)
 5. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

When the internal clock is used as the serial clock in the 3-wire serial I/O mode, set BRGC as described below. BRGC Setting is not required if an external serial clock is used.

(i) When the baud rate generator is not used:

Select a serial clock frequency with TPS0 through TPS3. Be sure then to set MDL0 through MDL3 to 1,1,1,1.

The serial clock frequency becomes 1/2 of the source clock frequency for the 5-bit counter.

(ii) When the baud rate generator is used:

Select a serial clock frequency with MDL0 through MDL3 and TPS0 through TPS3. Be sure then to set MDL0 through MDL3 to 1,1,1,1.

The serial clock frequency is calculated by the following formula.

$$\text{Serial clock frequency} = \frac{f_{xx}}{2^n \times (k + 16)} \quad [\text{Hz}]$$

f_x : Main system clock oscillation frequency

f_{xx} : Main system clock frequency (f_x or $f_x/2$)

n : Value set in TPS0 to TPS3 ($1 \leq n \leq 11$)

k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

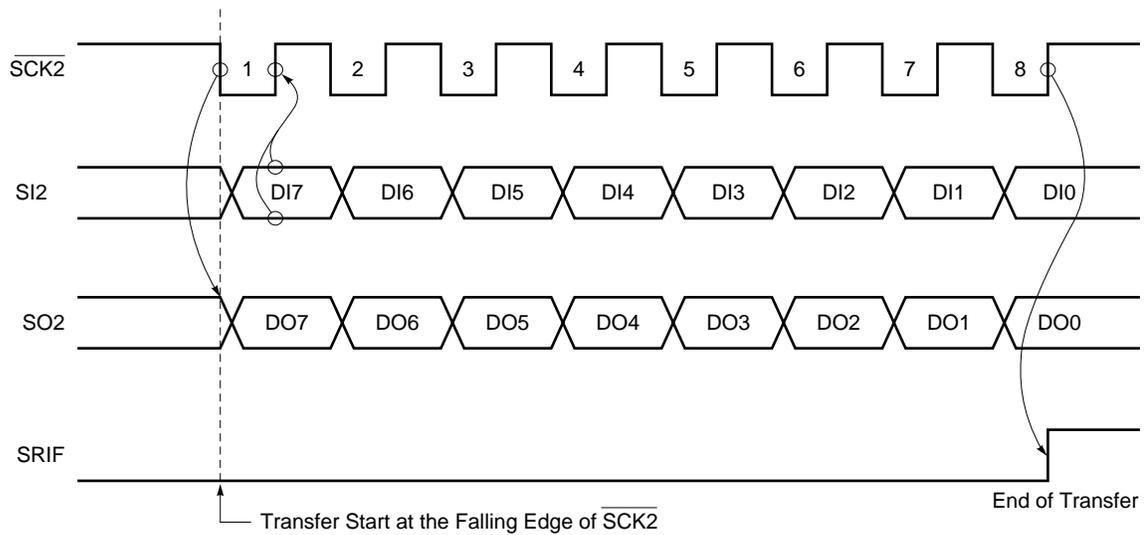
(2) Communication operation

In the 3-wire serial I/O mode, data transmission/reception is performed in 8-bit units. Data is transmitted/received bit by bit in synchronization with the serial clock.

Transmit shift register (TXS/SIO2) and receive shift register (RXS) shift operations are performed in synchronization with the fall of the serial clock ($\overline{\text{SCK2}}$). Then transmit data is held in the SO2 latch and output from the SO2 pin. Also, receive data input to the SI2 pin is latched in the receive buffer register (RXB/SIO2) on the rise of $\overline{\text{SCK2}}$.

At the end of an 8-bit transfer, the operation of TXS/SIO2 and RXS stops automatically, and the interrupt request flag (SRIF) is set.

Figure 20-12. 3-Wire Serial I/O Mode Timing



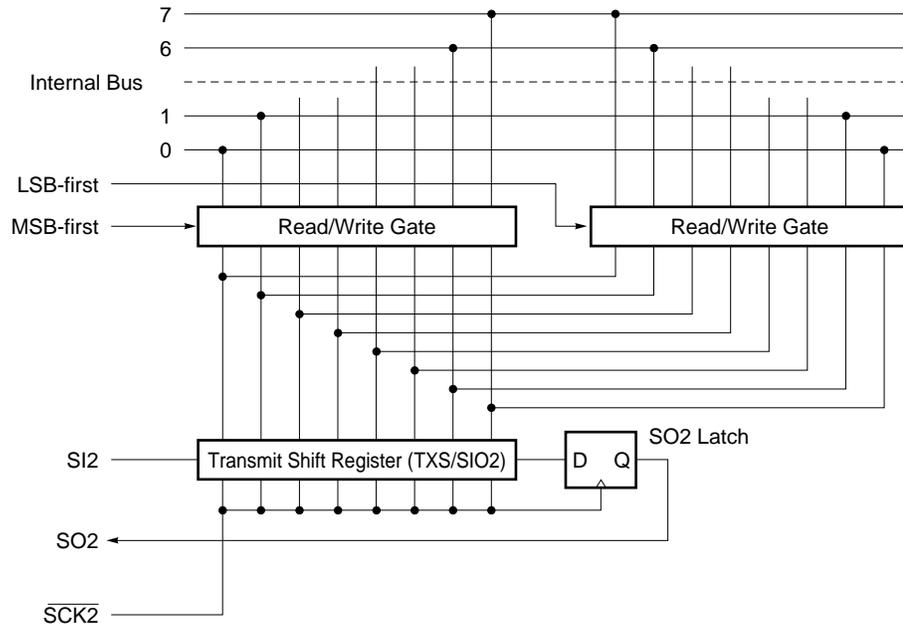
★ (3) **MSB/LSB switching as the start bit**

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

Figure 20-13 shows the configuration of the transmit shift register (TXS/SIO2) 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 (CSIM22) of the serial operating mode register 2 (CSIM2).

Figure 20-13. Circuit of Switching in Transfer Bit Order



Start bit switching is realized by switching the bit order for data write to SIO2. The SIO2 shift order remains unchanged.

Thus, MSB-first and LSB-first must be switched before writing data to the shift register.

(4) **Transfer start**

Serial transfer is started by setting transfer data to the transmission shift register (TXS/SIO2) when the following two conditions are satisfied.

- Serial interface channel 2 operation control bit (CSIE2) = 1
- Internal serial clock is stopped or $\overline{\text{SCK2}}$ is a high level after 8-bit serial transfer.

Caution If CSIE2 is set to “1” after data write to TXS/SIO2, transfer does not start.

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

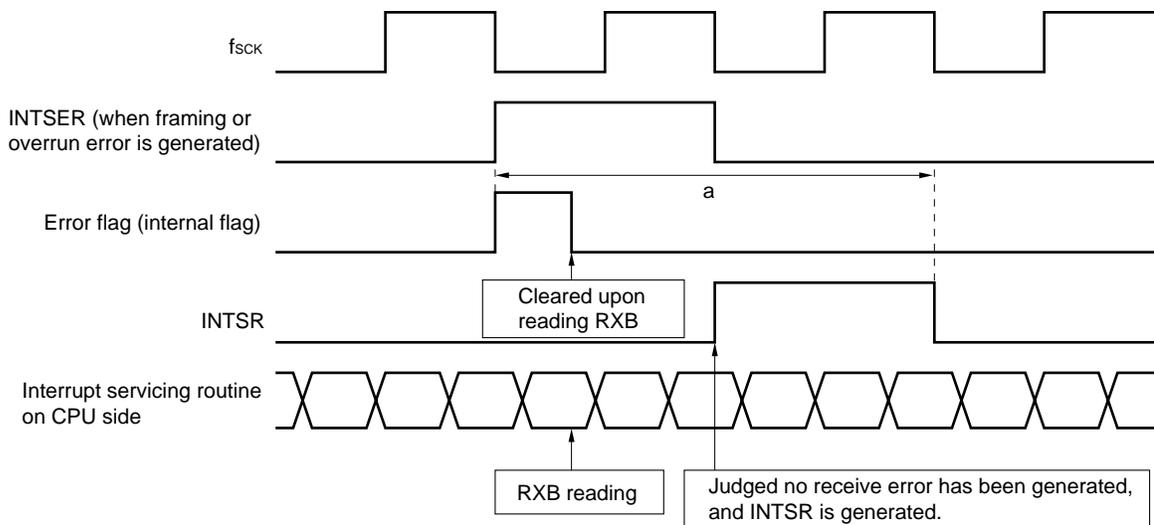
★ 20.4.4 Restrictions on using UART mode

In the UART mode, a receive completion interrupt request (INTSR) is generated after a certain period of time following the generation and clearing of the receive error interrupt request (INTSER). Thereby, the phenomenon shown below may occur.

● Details

If the bit 1 (ISRM) of the asynchronous serial interface mode register (ASIM) is set to 1, the setting is made such that receive completion interrupt request (INTSR) will not be generated upon the generation of a receive error. However, in the receive error interrupt request (INTSER) servicing, if the receive buffer register (RXB) is read within a certain timing (“a” in Figure 20-14), internal error flag is cleared (to 0). Therefore, no receive error is judged to have been generated, and INTSR, which is not supposed to be generated, will be generated. Figure 20-14 illustrates the operation above.

Figure 20-14. Receive Completion Interrupt Request Generation Timing (ISRM = 1)



Remark ISRM : Bit 1 of asynchronous serial interface mode register (ASIM)

f_{sck} : 5-bit counter source clock of baud rate generator

RXB : Receive buffer register

To avoid this phenomenon, implement the following countermeasures.

● Countermeasures

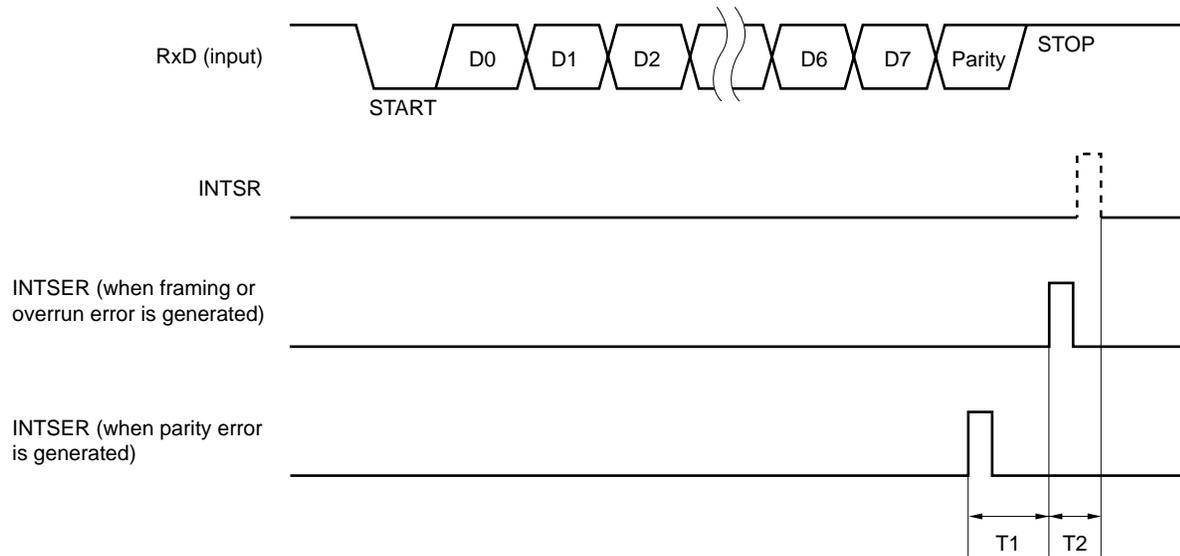
• In the case of framing error or overrun error

Prohibit the reading of the receive buffer register (RXB) for a certain period (“T₂” in Figure 20-15) after the generation of a receive error interrupt request (INTSER).

• In the case of parity error

Prohibit the reading of the receive buffer register (RXB) for a certain period (“T₁ + T₂” in Figure 20-15) after the generation of a receive error interrupt request (INTSER).

Figure 20-15. Period that Reading Receive Buffer Register is Prohibited



T1 : The amount of time for one unit of data sent in the baud rate selected with the baud rate generator control register (BRGC) (1/baud rate)

T2 : The amount of time for 2 clocks of 5-bit counter source clock (f_{sck}) selected with BRGC

● Example of countermeasures

An example of the countermeasures is shown below.

[Condition]

$f_x = 5.0 \text{ MHz}$

Processor clock control register (PCC) = 00H

Oscillation mode selection register (OSMS) = 01H

Baud rate generator control register (BRGC) = B0H (when 2400 bps is selected for baud rate)

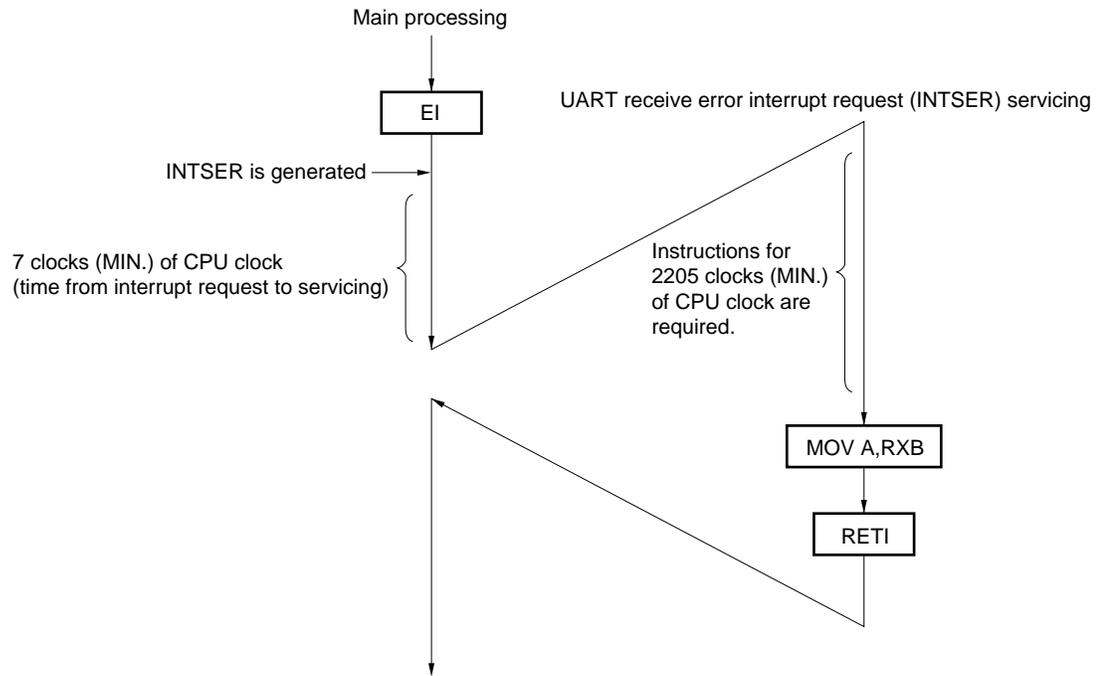
$$T_{CY} = 0.4 \mu\text{s} \quad (t_{CY} = 0.2 \mu\text{s})$$

$$T1 = \frac{1}{2400} = 416.7 \mu\text{s}$$

$$T2 = 12.8 \times 2 = 25.6 \mu\text{s}$$

$$\frac{T1 + T2}{t_{CY}} = 2212 \text{ (clock)}$$

[Example]



[MEMO]

CHAPTER 21 REAL-TIME OUTPUT PORT

21.1 Real-Time Output Port Functions

Data set previously in the real-time output buffer register can be transferred to the output latch by hardware concurrently with timer interrupt request or external interrupt request generation, then output externally. This is called the real-time output function. The pins that output data externally are called real-time output ports.

By using a real-time output, a signal which has no jitter can be output. This port is therefore suitable for control of stepping motors, etc.

Port mode/real-time output port mode can be specified bit-wise.

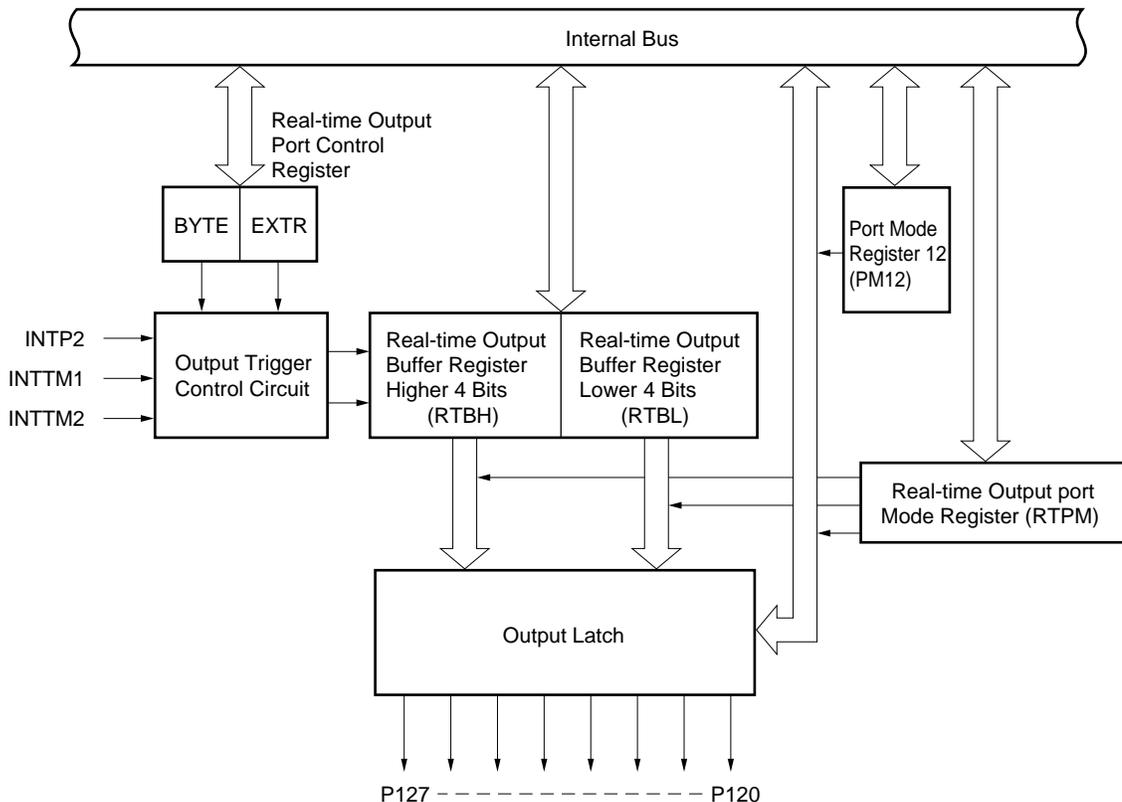
21.2 Real-Time Output Port Configuration

The real-time output port consists of the following hardware.

Table 21-1. Real-time Output Port Configuration

Item	Configuration
Register	Real-time output buffer register (RTBL, RTBH)
Control register	Port mode register 12 (PM12) Real-time output port mode register (RTPM) Real-time output port control register (RTPC)

Figure 21-1. Real-time Output Port Block Diagram



(1) Real-time output buffer register (RTBL, RTBH)

Addresses of RTBL and RTBH are mapped individually in the special function register (SFR) area as shown in Figure 21-2.

When specifying 4 bits x 2 channels as the operating mode, data are set individually in RTBL and RTBH. When specifying 8 bits x 1 channel as the operating mode, data are set to both RTBL and RTBH by writing 8-bit data to either RTBL or RTBH.

Table 21-2 shows operations during manipulation of RTBL and RTBH.

Figure 21-2. Real-time Output Buffer Register Configuration

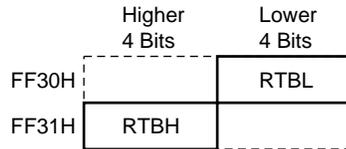


Table 21-2. Operation in Real-time Output Buffer Register Manipulation

Operating Mode	Register to be Manipulated	In Read ^{Note 1}		In Write ^{Note 2}	
		Higher 4 Bits	Lower 4 Bits	Higher 4 Bits	Lower 4 Bits
4 Bits x 2 Channels	RTBL	RTBH	RTBL	Invalid	RTBL
	RTBH	RTBH	RTBL	RTBH	Invalid
8 Bits x 1 Channel	RTBL	RTBH	RTBL	RTBH	RTBL
	RTBH	RTBH	RTBL	RTBH	RTBL

- Notes**
1. Only the bits set in the real-time output port mode can be read. When a read is performed to the bits set in the port mode, 0 is read out.
 2. After setting data in the real-time output port, output data should be set in RTBL and RTBH by the time a real-time output trigger is generated.

21.3 Real-Time Output Port Control Registers

The following three registers control the real-time output port.

- Port mode register 12 (PM12)
- Real-time output port mode register (RTPM)
- Real-time output port control register (RTPC)

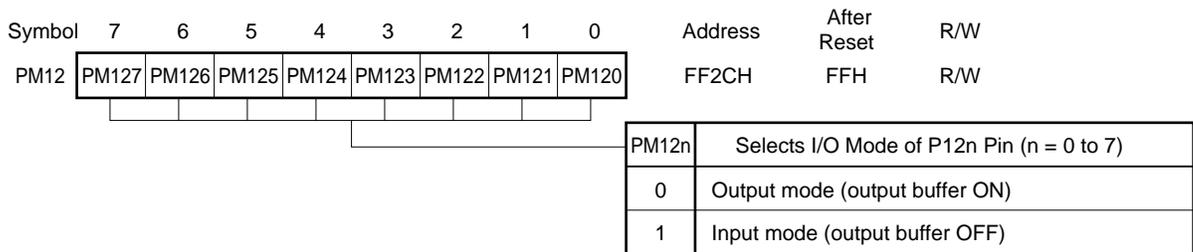
(1) Port mode register 12 (PM12)

This register sets the input or output mode of port 12 pins (P120 to P127) which also function as real-time output pins (RTP0 to RTP7). To use port 12 as a real-time output port, the port pin that performs real-time output must be set in the output mode (PM12n = 0: n = 0 to 7).

PM12 is set with a 1-bit or 8-bit memory manipulation instruction.

RESET input sets this register to FFH.

Figure 21-3. Port Mode Register 12 Format



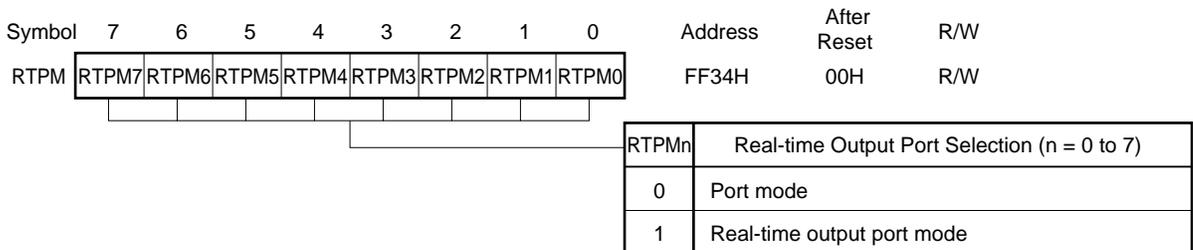
(2) Real-time output port mode register (RTPM)

This register selects the real-time output port mode/port mode bit-wise.

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

RESET input sets this register to 00H.

Figure 21-4. Real-time Output Port Mode Register Format



- Cautions**
1. When using these bits as a real-time output port, set the ports to which real-time output is performed to the output mode (set the bits of the port mode register 12 (PM12) to 0).
 2. In the port specified as a real-time output port, data cannot be set to the output latch. Therefore, when setting an initial value, data should be set to the output latch before setting the real-time output mode.

(3) Real-time output port control register (RTPC)

This register sets the real-time output port operating mode and output trigger.

Table 21-3 shows the relationship between the real-time output port operating mode and output trigger.

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

RESET input sets this register to 00H.

Figure 21-5. Real-time Output Port Control Register Format

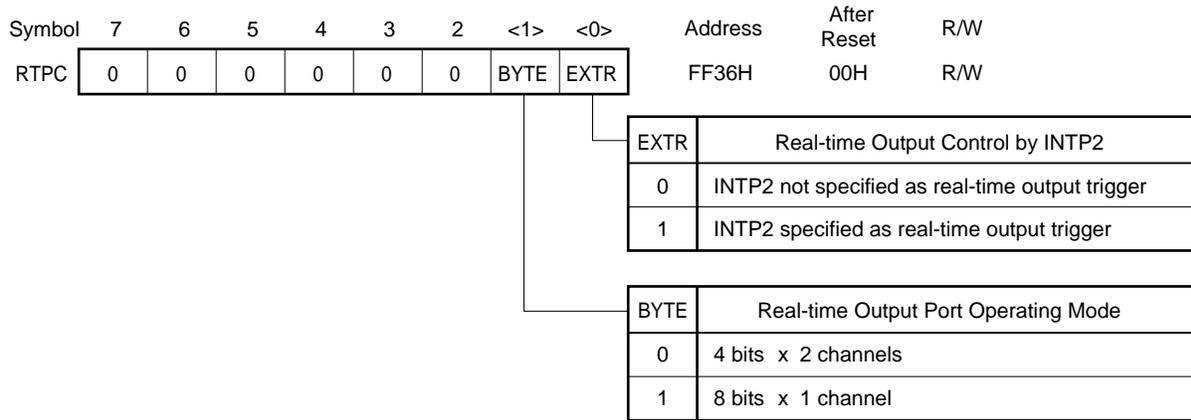


Table 21-3. Real-time Output Port Operating Mode and Output Trigger

BYTE	EXTR	Operating Mode	RTBH → Port Output	RTBL → Port Output
0	0	4 bits x 2 channels	INTTM2	INTTM1
	1		INTTM1	INTP2
1	0	8 bits x 1 channel	INTTM1	
	1		INTP2	

CHAPTER 22 INTERRUPT FUNCTIONS

22.1 Interrupt Function Types

The following three types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally even in a disabled state. It does not undergo interrupt priority control and is given top priority over all other interrupt requests.

It generates a standby release signal.

The non-maskable interrupt has one source of interrupt request from the watchdog timer.

(2) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into a high interrupt priority group and a low interrupt priority group by setting the priority specify flag register (PR0L, PR0H, and PR1L).

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 (see **Table 22-1**).

A standby release signal is generated.

The maskable interrupt has seven sources of external interrupt requests and fifteen sources of internal interrupt requests.

(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.

22.2 Interrupt Sources and Configuration

There are total of 24 non-maskable, maskable, and software interrupts in the interrupt sources (see **Table 22-1**).

Table 22-1. Interrupt Source List (1/2)

Interrupt Type	Default Priority ^{Note 1}	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type ^{Note 2}
		Name	Trigger			
Non-maskable	–	INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0004H	(A)
		INTWDT	Watchdog timer overflow (with interval timer mode selected)			(B)
Maskable	0	INTP0	Pin input edge detection	External	0006H	(C)
	1	INTP1			0008H	(D)
	2	INTP2			000AH	
	3	INTP3			000CH	
	4	INTP4			000EH	
	5	INTP5			0010H	
	6	INTP6			0012H	
	7					
	8	INTCSI0	End of serial interface channel 0 transfer	Internal	0014H	(B)
	9	INTCSI1	End of serial interface channel 1 transfer		0016H	
	10	INTSER	Serial interface channel 2 UART reception error generation		0018H	
	11	INTSR	End of serial interface channel 2 UART reception		001AH	
INTCSI2		End of serial interface channel 2 3-wire transfer				
12	INTST	End of serial interface channel 2 UART transfer	001CH			

- Notes**
1. Default priorities are intended for two or more simultaneously generated maskable interrupt requests. 0 is the highest priority and 20 is the lowest priority.
 2. Basic configuration types (A) to (E) correspond to (A) to (E) of Figure 22-1.

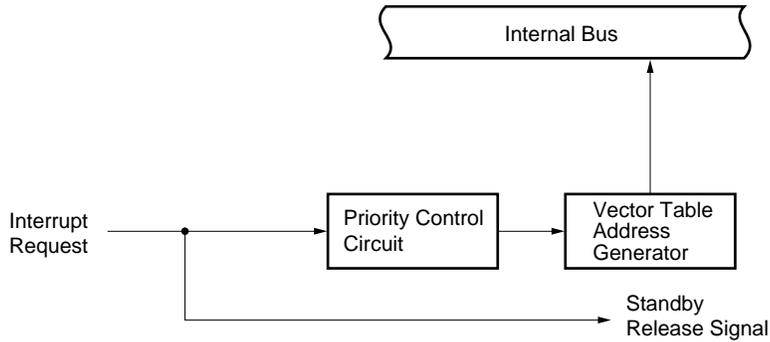
Table 22-1. Interrupt Source List (2/2)

Interrupt Type	Default Priority ^{Note 1}	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type ^{Note 2}
		Name	Trigger			
Maskable	13	INTTM3	Reference time interval signal from watch timer	Internal	001EH	(B)
	14	INTTM00	Generation of 16-bit timer register, capture/compare register (CR00) match signal		0020H	
	15	INTTM01	Generation of 16-bit timer register, capture/compare register (CR01) match signal		0022H	
	16	INTTM1	Generation of 8-bit timer/event counter 1 match signal		0024H	
	17	INTTM2	Generation of 8 bit timer/event counter 2 match signal		0026H	
	18	INTAD	End of A/D converter conversion		0028H	
	19	INTTM5	Generation of 8-bit timer/event counter 5 match signal		002AH	
	20	INTTM6	Generation of 8-bit timer/event counter 6 match signal		002CH	
Software	—	BRK	BRK instruction execution	—	003EH	(E)

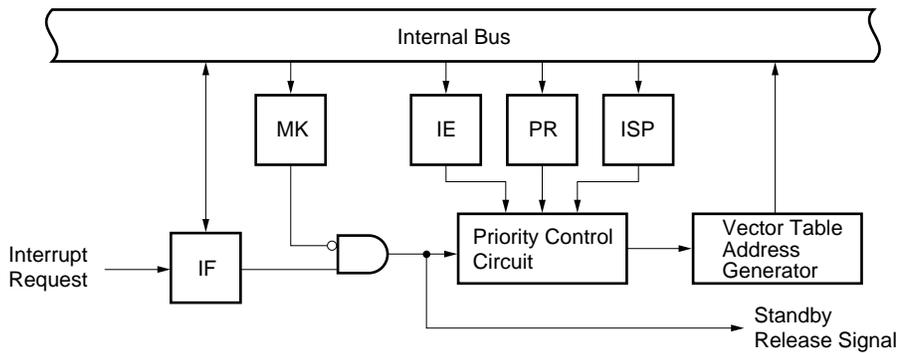
- Notes**
1. Default priorities are intended for two or more simultaneously generated maskable interrupt requests. 0 is the highest priority and 20 is the lowest priority.
 2. Basic configuration types (A) to (E) correspond to (A) to (E) of Figure 22-1.

Figure 22-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt (INTP0)

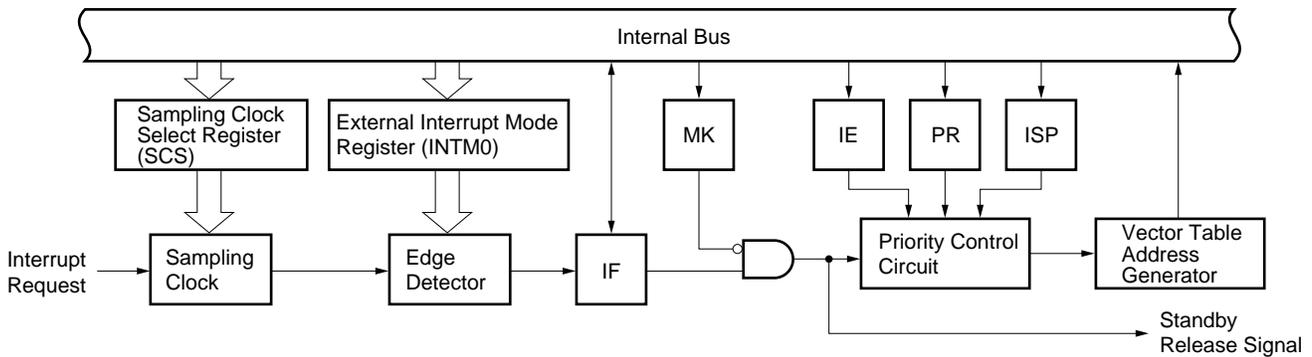
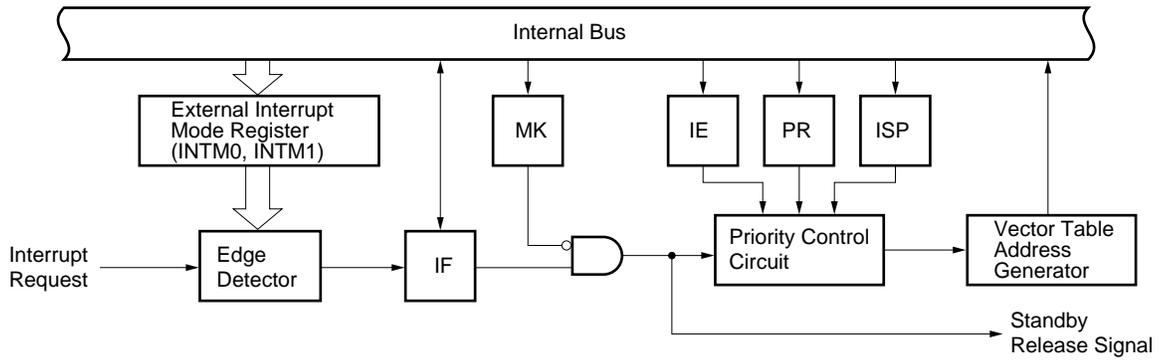
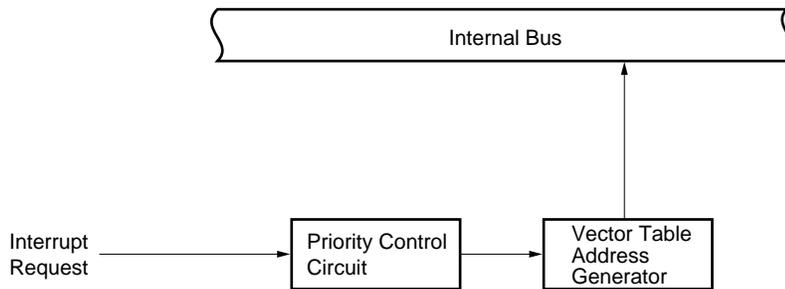


Figure 22-1. Basic Configuration of Interrupt Function (2/2)

(D) External maskable interrupt (except INTP0)



(E) Software interrupt



- IF : Interrupt request flag
- IE : Interrupt enable flag
- ISP : Inservice priority flag
- MK : Interrupt mask flag
- PR : Priority specify flag

22.3 Interrupt Function Control Registers

The following six types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H, IF1L)
- Interrupt mask flag register (MK0L, MK0H, MK1L)
- Priority specify flag register (PR0L, PR0H, PR1L)
- External interrupt mode register (INTM0, INTM1)
- Sampling clock select register (SCS)
- Program status word (PSW)

Table 22-2 gives a listing of interrupt request flags, interrupt mask flags, and priority specify flags corresponding to interrupt request sources.

Table 22-2. Various Flags Corresponding to Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specify Flag	
		Register		Register		Register
INTWDT	TMIF4	IF0L	TMMK4	MK0L	TMPR4	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTP4	PIF4		PMK4		PPR4	
INTP5	PIF5		PMK5		PPR5	
INTP6	PIF6	PMK6	PPR6			
INTCSI0	CSIF0	IF0H	CSIMK0	MK0H	CSIPR0	PR0H
INTCSI1	CSIF1		CSIMK1		CSIPR1	
INTSER	SERIF		SERMK		SERPR	
INTSR/INTCSI2	SRIF		SRMK		SRPR	
INTST	STIF		STMK		STPR	
INTTM3	TMIF3		TMMK3		TMPR3	
INTTM00	TMIF00		TMMK00		TMPR00	
INTTM01	TMIF01		TMMK01		TMPR01	
INTTM1	TMIF1	IF1L	TMMK1	MK1L	TMPR1	PR1L
INTTM2	TMIF2		TMMK2		TMPR2	
INTAD	ADIF		ADMK		ADPR	
INTTM5	TMIF5		TMMK5		TMPR5	
INTTM6	TMIF6		TMMK6		TMPR6	

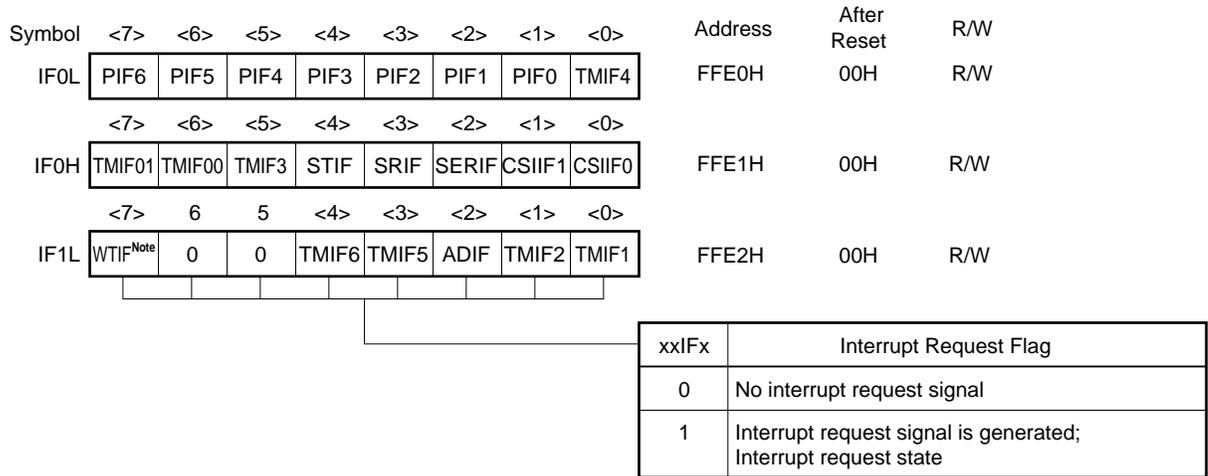
(1) Interrupt request flag registers (IF0L, IF0H, IF1L)

The interrupt request flag is set to 1 when the corresponding interrupt request is generated or an instruction is executed. It is cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon application of $\overline{\text{RESET}}$ input.

IF0L, IF0H, and IF1L 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.

$\overline{\text{RESET}}$ input sets these registers to 00H.

Figure 22-2. Interrupt Request Flag Register Format



Note WTIF is test input flag. Vectored interrupt request is not generated.

- Cautions**
1. TMIF4 flag is R/W enabled only when a watchdog timer is used as an interval timer. If used in the watchdog timer mode 1, set TMIF4 flag to 0.
 2. Set always 0 in IF1L bit 5 and bit 6.

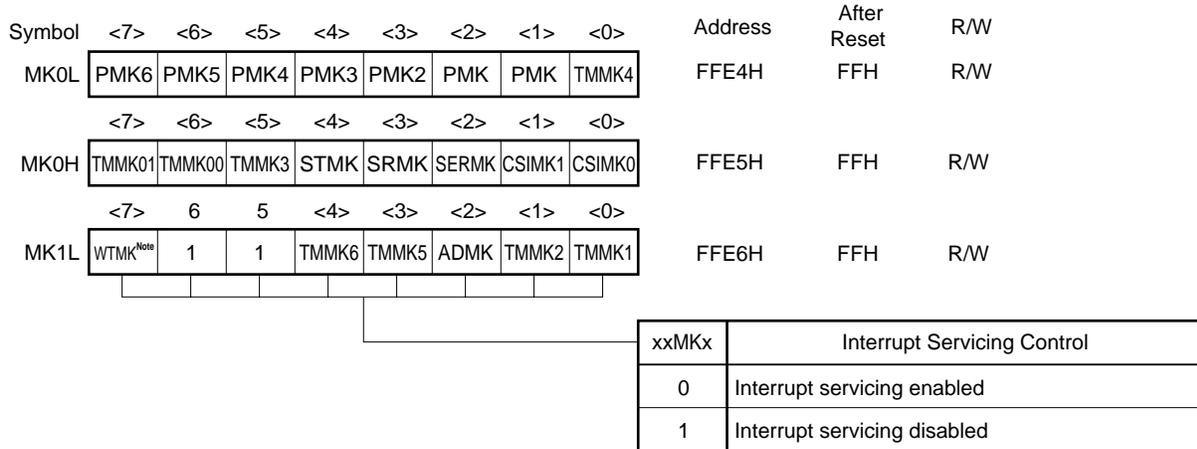
(2) Interrupt mask flag registers (MK0L, MK0H, MK1L)

The interrupt mask flag is used to enable/disable the corresponding maskable interrupt service and to set standby clear enable/disable.

MK0L, MK0H, and MK1L are set with a 1-bit or 8-bit memory manipulation instruction. If IF0L and IF0H are used as a 16-bit register MK0, use a 16-bit memory manipulation instruction for the setting.

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

Figure 22-3. Interrupt Mask Flag Register Format



Note WTMK controls standby mode release enable/disable and does not control interrupt functions.

- Cautions**
1. If TMMK4 flag is read when a watchdog timer is used in the watchdog timer mode 1, the read 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. Set always 1 in MK1L bit 5 and bit 6.

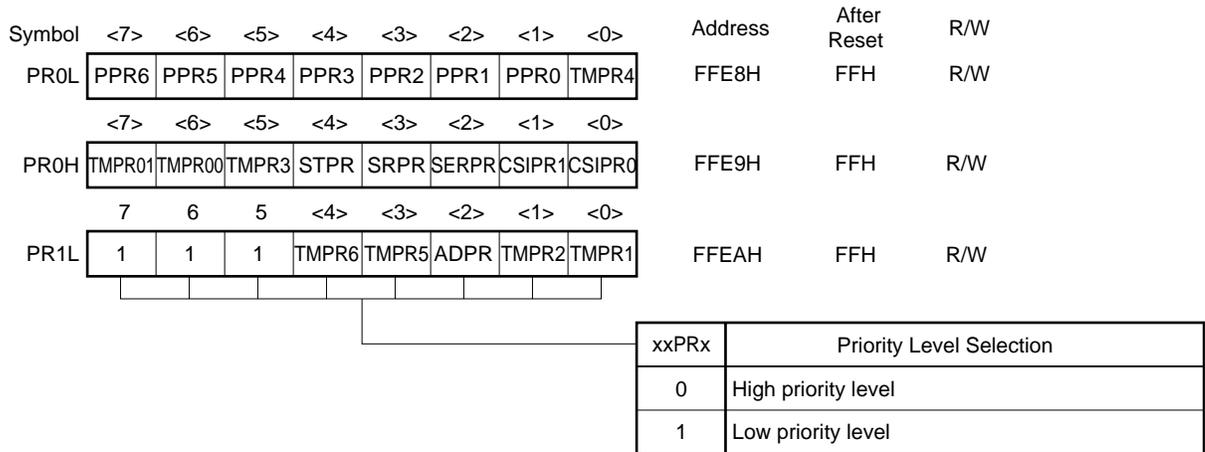
(3) Priority specify flag registers (PR0L, PR0H, PR1L)

The priority specify flag is used to set the corresponding maskable interrupt priority orders.

PR0L, PR0H, and PR1L are set with a 1-bit or 8-bit memory manipulation instruction. If IF0L and IF0H 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 22-4. Priority Specify Flag Register Format



- Cautions**
1. When a watchdog timer is used in the watchdog timer mode 1, set 1 in TMPR4 flag.
 2. Set always 1 in PR1L bit 5 to bit 7.

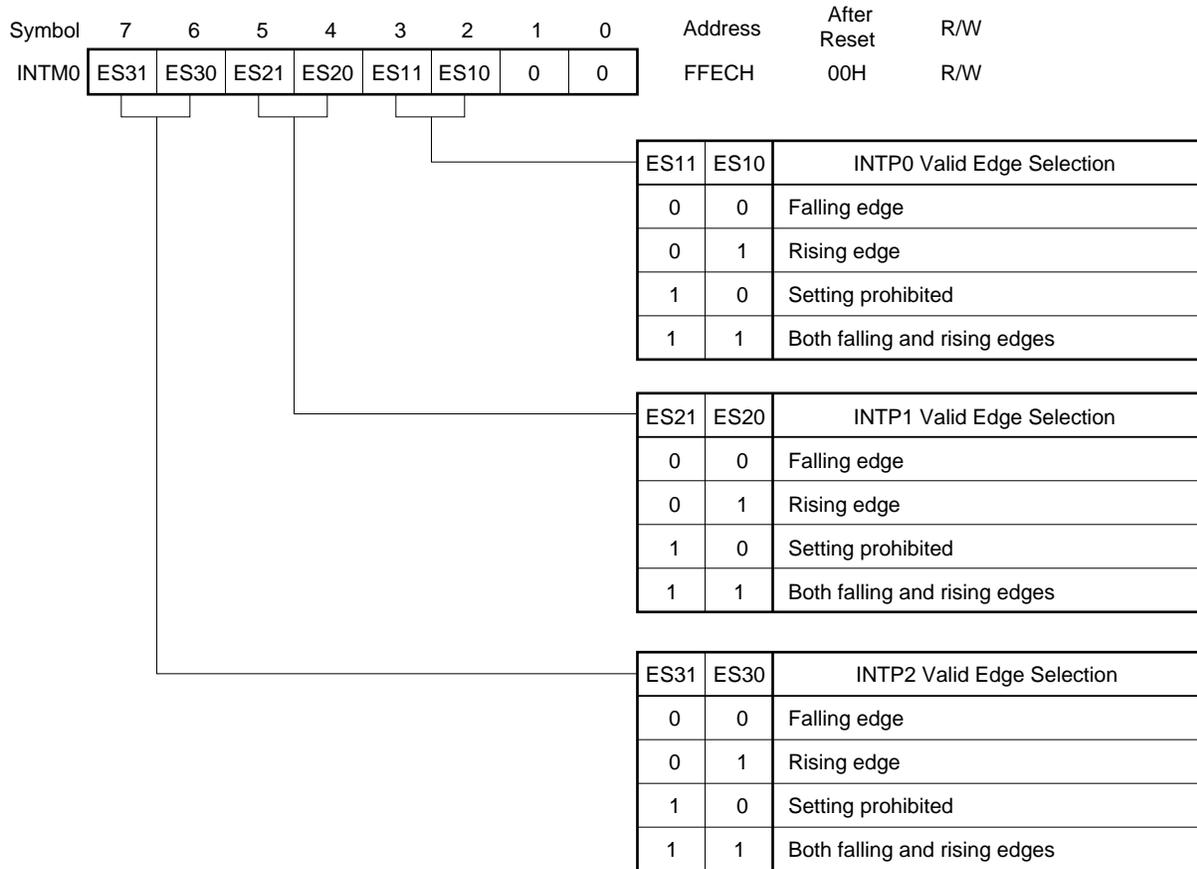
(4) External interrupt mode register (INTM0, INTM1)

These registers set the valid edge for INTP0 to INTP6.

INTM0 and INTM1 are set by 8-bit memory manipulation instructions.

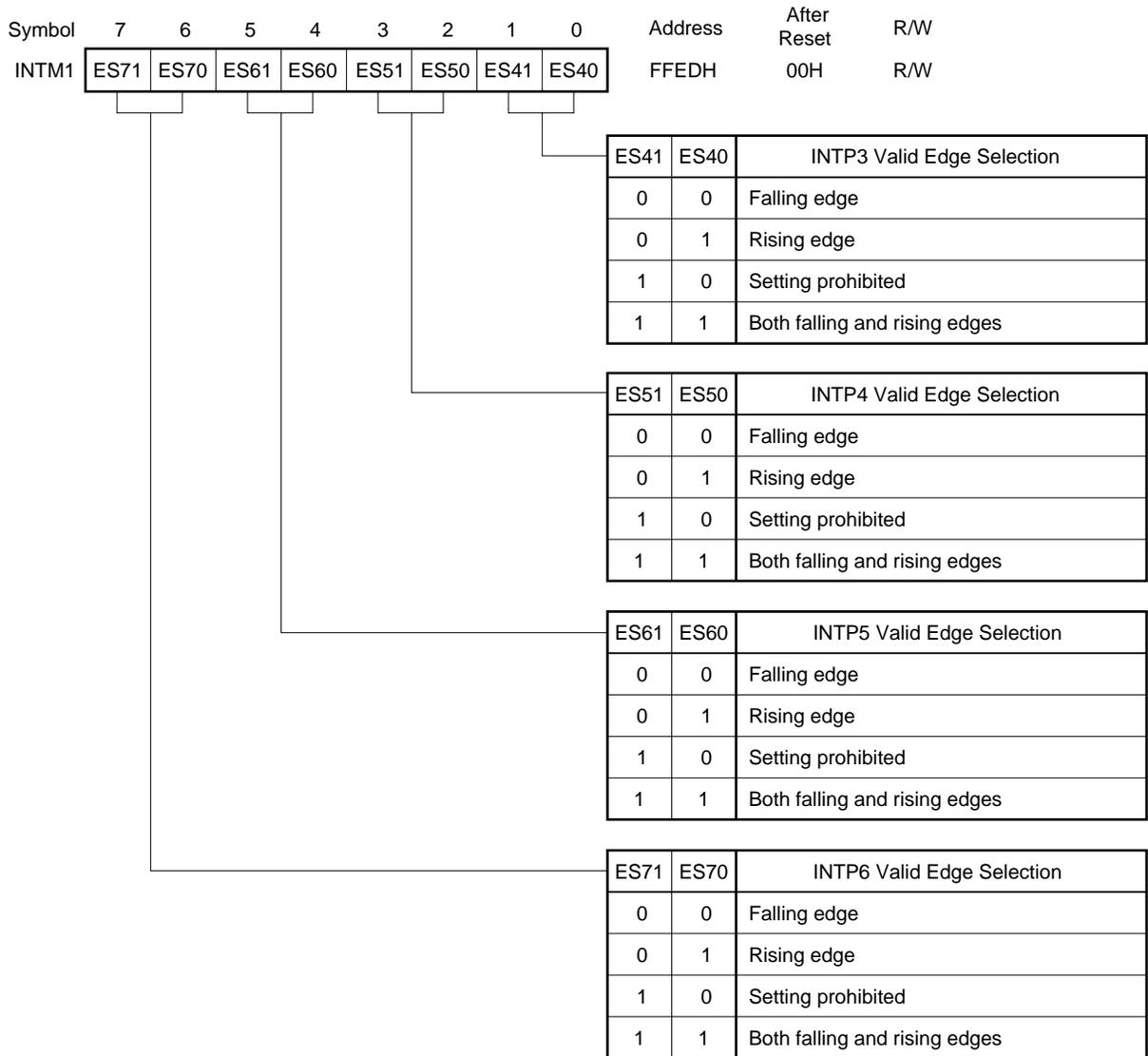
$\overline{\text{RESET}}$ input sets these registers to 00H.

Figure 22-5. External Interrupt Mode Register 0 Format



Caution Set 0, 0, 0 to bits 1 through 3 (TMC01 through TMC03) of the 16-bit timer mode control register (TMC0) and stop the timer operation before setting the valid edges of INTP0/TI00/P00 pin.

Figure 22-6. External Interrupt Mode Register 1 Format



(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 22-7. Sampling Clock Select Register Format



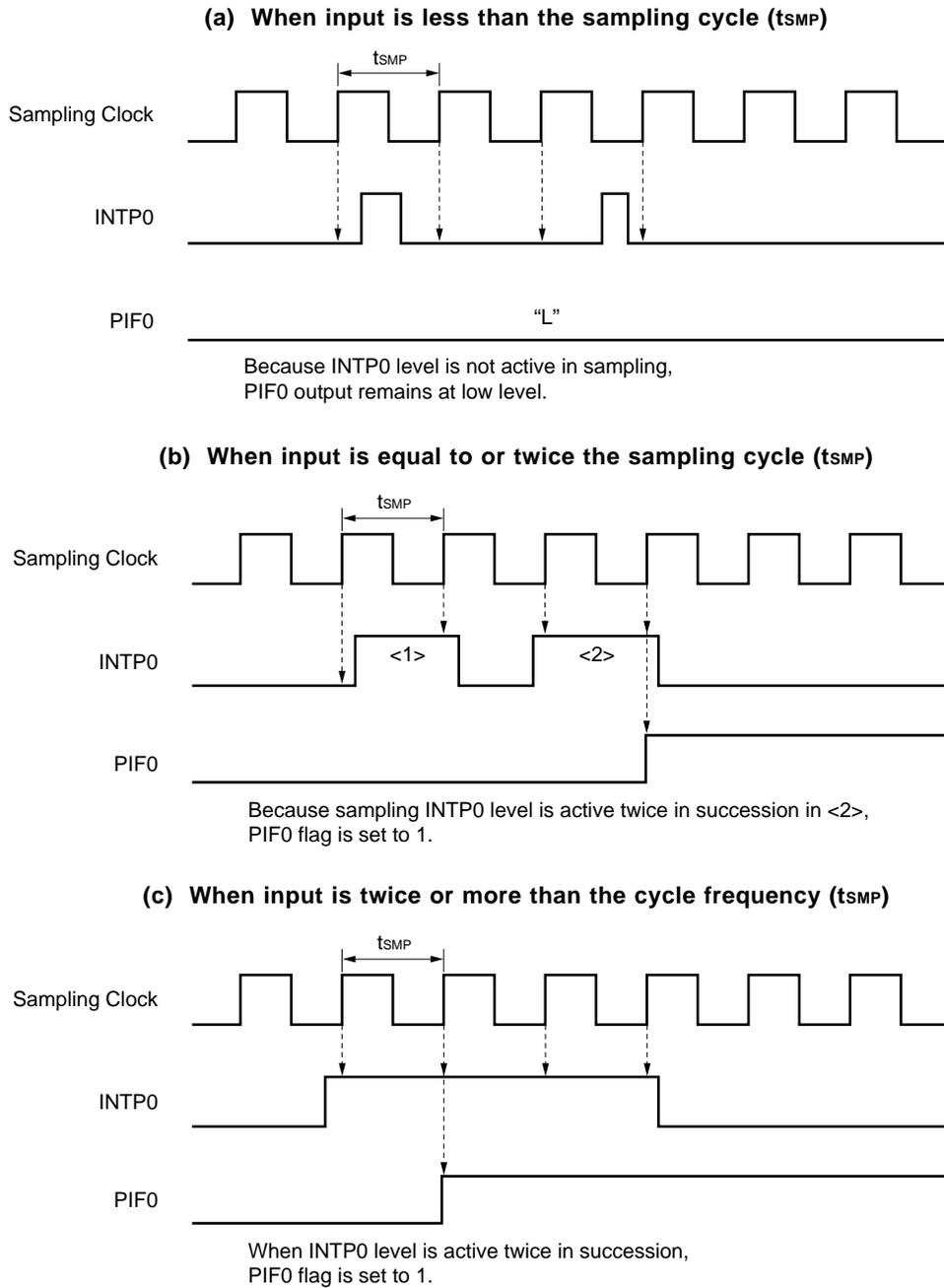
Caution $f_{xx}/2^N$ is a clock to be supplied to the CPU and $f_{xx}/2^5$, $f_{xx}/2^6$ and $f_{xx}/2^7$ are clocks to be supplied to the peripheral hardware. $f_{xx}/2^N$ stops in the HALT mode.

- Remarks**
1. N : Value (N = 0 to 4) at bits 0 to 2 (PCC0 to PCC2) of processor clock control register
 2. f_{xx} : Main system clock frequency (f_x or $f_x/2$)
 3. f_x : Main system clock oscillation frequency
 4. MCS : Bit 0 of oscillation mode selection register (OSMS)
 5. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

When the setting INTP0 input level is active twice in succession, the noise eliminator sets interrupt request flag (PIF0) to 1.

Figure 22-8 shows the noise eliminator input/output timing.

Figure 22-8. Noise Eliminator Input/Output Timing (during Rising Edge Detection)

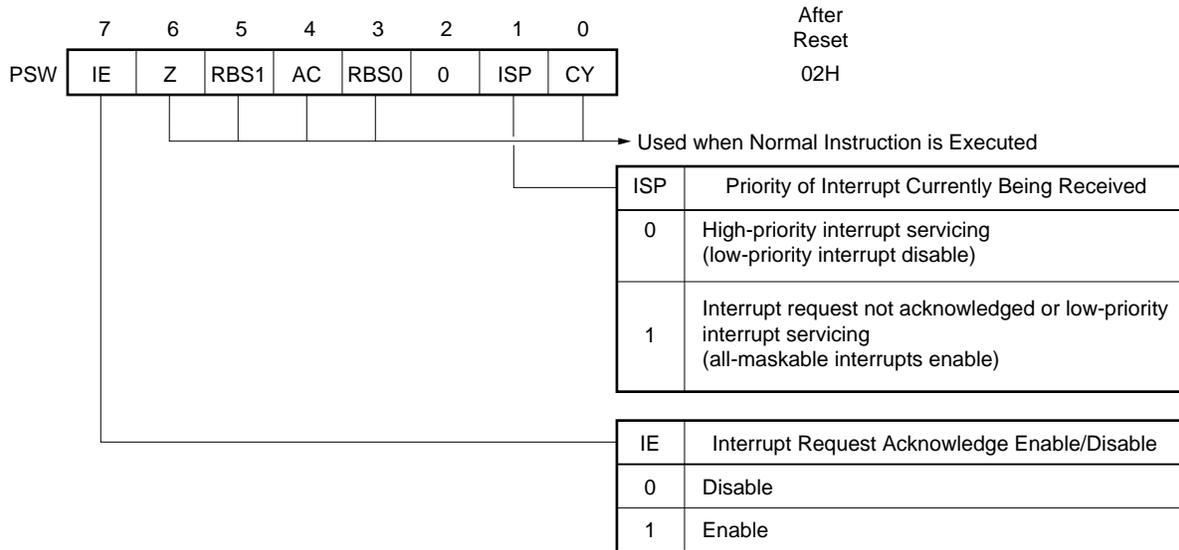


(6) Program status word (PSW)

The program status word is a register to hold the instruction execution result and the current status for interrupt request. The IE flag to set maskable interrupt enable/disable and the ISP flag to control multiple interrupt servicing 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, and when the BRK instruction is executed, the contents of PSW automatically is saved into a stack and the IE flag is reset to 0. If a maskable interrupt request is acknowledged contents of the priority specify flag of the acknowledged interrupt are transferred to the ISP flag. The acknowledged contents of PSW is also saved into the stack with the PUSH PSW instruction. It is reset from the stack with the RETI, RETB, and POP PSW instructions. RESET input sets PSW to 02H.

Figure 22-9. Program Status Word Format



22.4 Interrupt Servicing Operations

22.4.1 Non-maskable interrupt request acknowledge operation

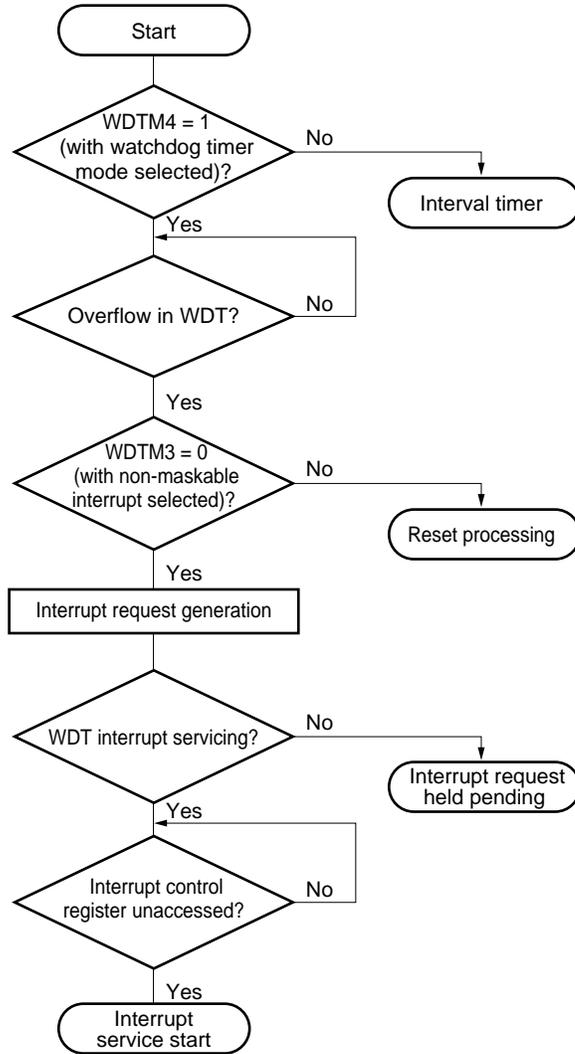
A non-maskable interrupt request is unconditionally acknowledged even if in an interrupt request acknowledge disable state. It does not undergo interrupt priority control and has highest priority over all other interrupts.

If a non-maskable interrupt request is acknowledged, the acknowledged interrupt is saved in the stacks, PSW and PC, in that order, the IE and ISP flags are reset to 0, and the vector table contents are loaded into PC and branched.

A new non-maskable interrupt request generated during execution of a non-maskable interrupt servicing program is acknowledged after the current execution of the non-maskable interrupt servicing program is terminated (following RETI instruction execution) and one main routine instruction is executed. If a new non-maskable interrupt request is generated twice or more during non-maskable interrupt service program execution, only one non-maskable interrupt request is acknowledged after termination of the non-maskable interrupt service program execution.

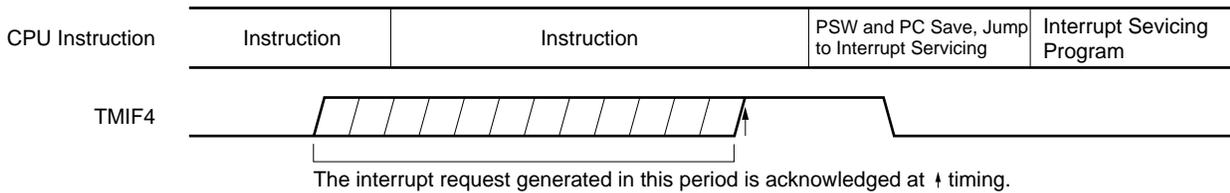
Figure 22-10 shows the flowchart from non-maskable interrupt request generation to acknowledge. Figure 22-11 shows the non-maskable interrupt request acknowledge timing. Figure 22-12 shows the acknowledge operation if multiple non-maskable interrupt requests are generated.

Figure 22-10. Flowchart from Non-Maskable Interrupt Generation to Acknowledge



WDTM : Watchdog timer mode register
 WDT : Watchdog timer

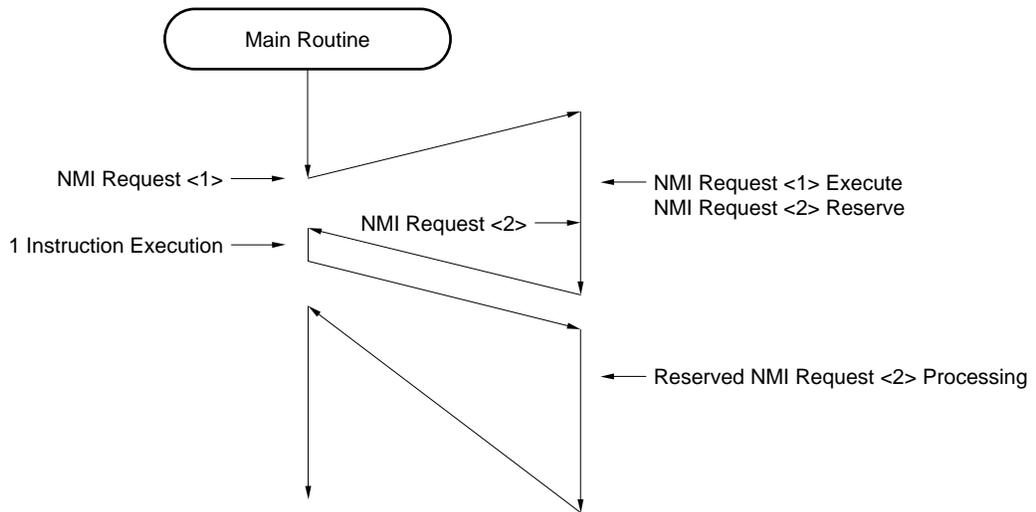
Figure 22-11. Non-Maskable Interrupt Request Acknowledge Timing



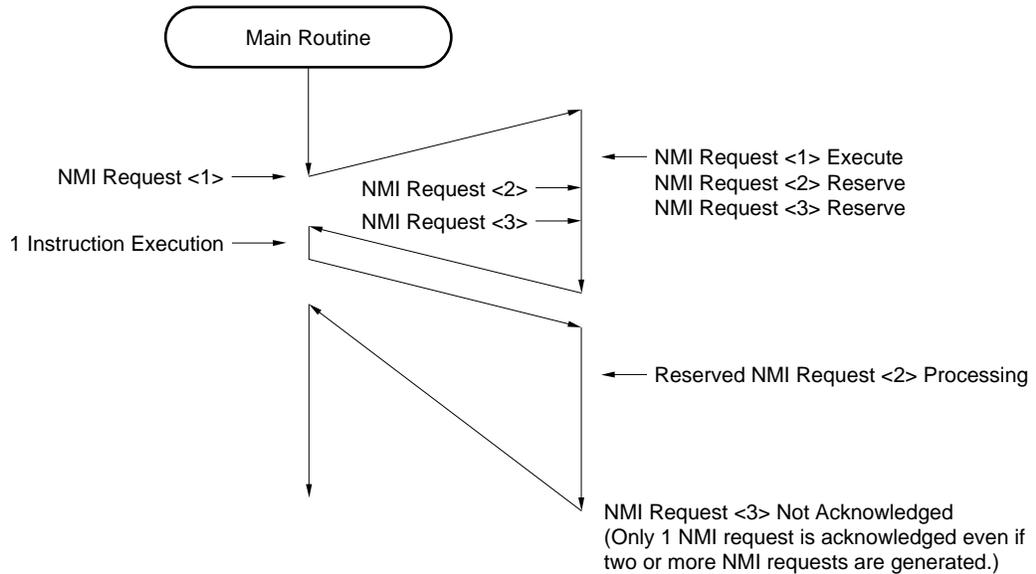
TMIF4 : Watchdog timer interrupt request flag

Figure 22-12. Non-Maskable Interrupt Request Acknowledge Operation

(a) If a new non-maskable interrupt request is generated during non-maskable interrupt servicing program execution



(b) If two non-maskable interrupt requests are generated during non-maskable interrupt servicing program execution



22.4.2 Maskable interrupt request acknowledge operation

A maskable interrupt request becomes acknowledgeable when an interrupt request flag is set to 1 and the interrupt mask (MK) flag is cleared to 0. A vectored interrupt request is acknowledged in an interrupt enable state (with IE flag set to 1). However, a low-priority interrupt request is not acknowledged during high-priority interrupt service (with ISP flag reset to 0).

Wait times maskable interrupt request generation to interrupt servicing are shown in Figure 22-3.

Refer to Figures 22-14 and 22-15 for the interrupt request acknowledge timing.

Table 22-3. Times from Maskable Interrupt Request Generation to Interrupt Service

	Minimum Time	Maximum Time ^{Note}
When xxPRx = 0	7 clocks	32 clocks
When xxPRx = 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: $\frac{1}{f_{\text{CPU}}}$ (f_{CPU} : CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request specified for higher priority with the priority specify flag is acknowledged first. If two or more requests are specified for the same priority with the priority specify flag, the interrupt request with the higher default priority is acknowledged first.

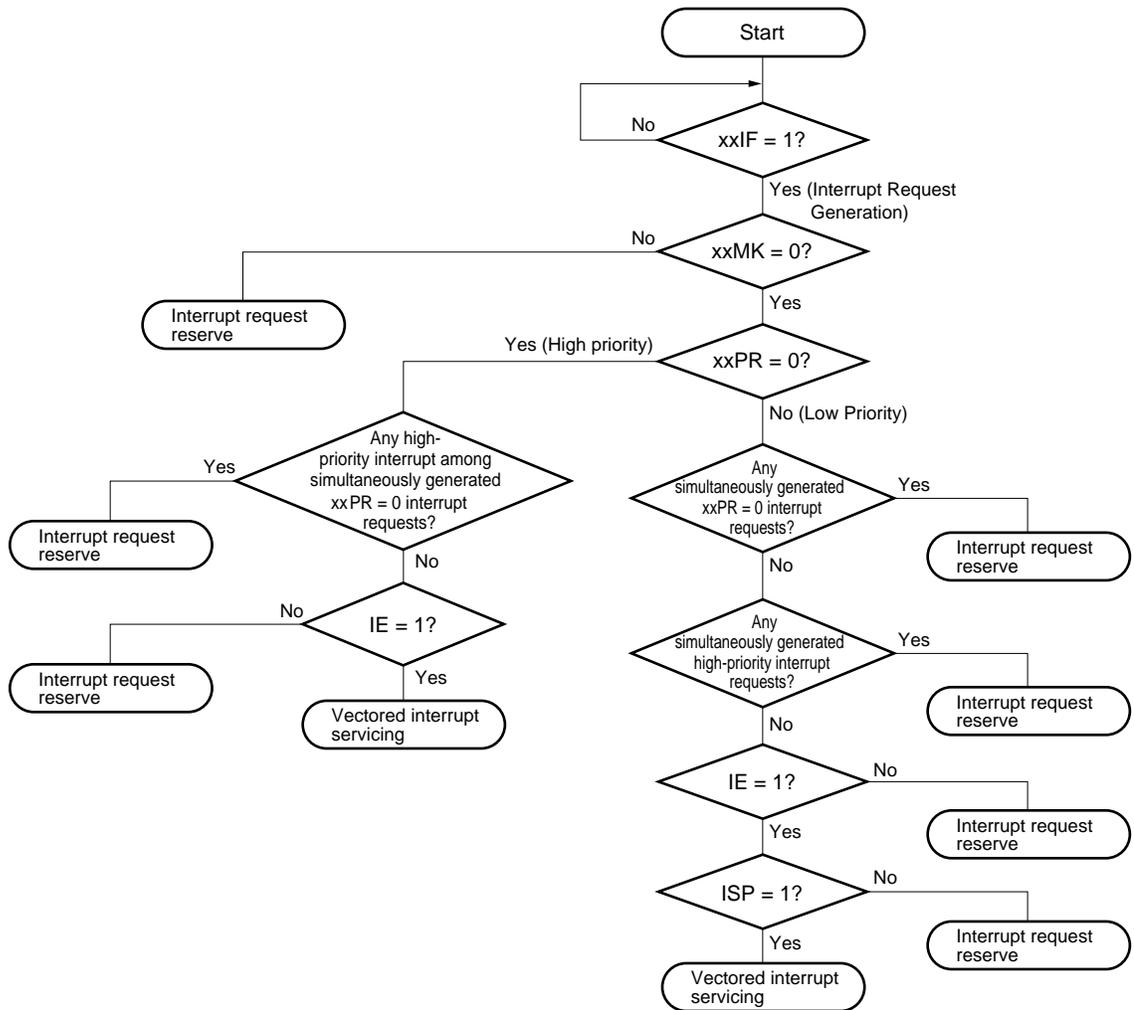
Any reserved interrupt requests are acknowledged when they become acknowledgeable.

Figure 22-13 shows interrupt request acknowledge algorithms.

When a maskable interrupt request is acknowledged, the contents of program status word (PSW) and program counter (PC) are saved to stacks, in this order. Then, the IE flag is reset (to 0), and the value of the acknowledged interrupt priority specify flag is 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 22-13. Interrupt Request Acknowledge Processing Algorithm



xxIF : Interrupt request flag

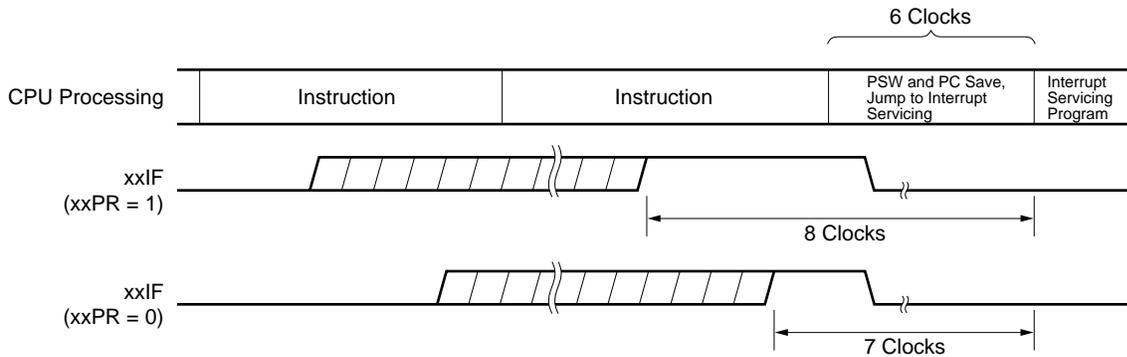
xxMK : Interrupt mask flag

xxPR : Priority specify flag

IE : Flag to control maskable interrupt request acknowledge

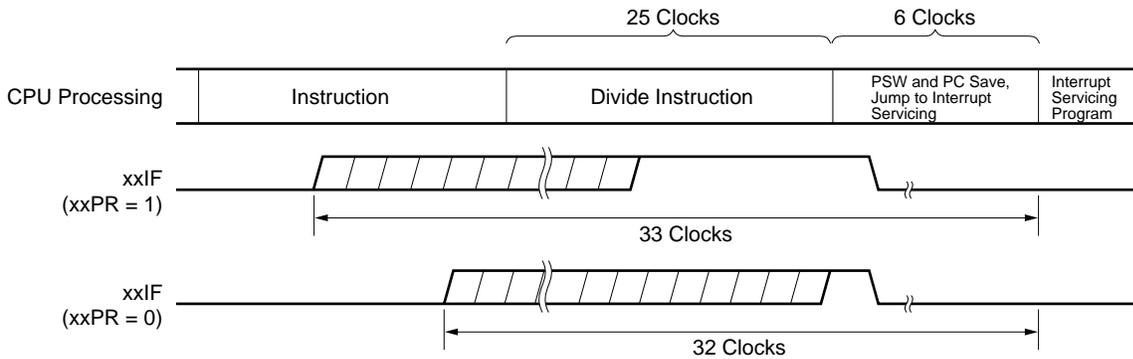
ISP : Flag to indicate the priority of interrupt being serviced (0 = an interrupt with higher priority is being serviced, 1 = interrupt request is not acknowledged or an interrupt with lower priority is being serviced)

Figure 22-14. Interrupt Request Acknowledge Timing (Minimum Time)



Remark 1 clock: $\frac{1}{f_{\text{CPU}}}$ (f_{CPU} : CPU clock)

Figure 22-15. Interrupt Request Acknowledge Timing (Maximum Time)



Remark 1 clock: $\frac{1}{f_{\text{CPU}}}$ (f_{CPU} : CPU clock)

22.4.3 Software interrupt request acknowledge operation

A software interrupt request is acknowledged by BRK instruction execution. Software interrupt cannot be disabled.

If a software interrupt is acknowledged, the contents of program status word (PSW) and program counter (PC) are saved to stacks, in this order. Then 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.

22.4.4 Multiple interrupt servicing

A multiple interrupt consists in acknowledging another interrupt during the execution of the interrupt.

A multiple interrupt is generated only in the interrupt request acknowledge enable state (IE = 1) (except non-maskable interrupt). As soon as an interrupt request is acknowledged, it enters the acknowledge disable state (IE = 0). Therefore, in order to enable a multiple interrupt, it is necessary to set the interrupt enable state by setting the IE flag (1) with the EI instruction during interrupt servicing.

Even in an interrupt enabled state, a multiple interrupt may not be enabled. However, it is controlled according to the interrupt priority. There are two priorities, the default priority and the programmable priority. The multiple interrupt is controlled by the programmable priority control.

If an interrupt request with the same or higher priority than that of the interrupt being serviced is generated, it is acknowledged as a multiple interrupt. In the case of an interrupt with a priority lower than that of the interrupt being processed, it is not acknowledged as a multiple interrupt.

Interrupt request not acknowledged as a multiple interrupt due to interrupt disable or a low priority is reserved and acknowledged following one instruction execution of the main processing after the completion of the interrupt being serviced.

During non-maskable interrupt servicing, multiple interrupts are not enabled.

Table 22-4 shows an interrupt request enabled for multiple interrupt during interrupt servicing, and Figure 22-16 shows multiple interrupt examples.

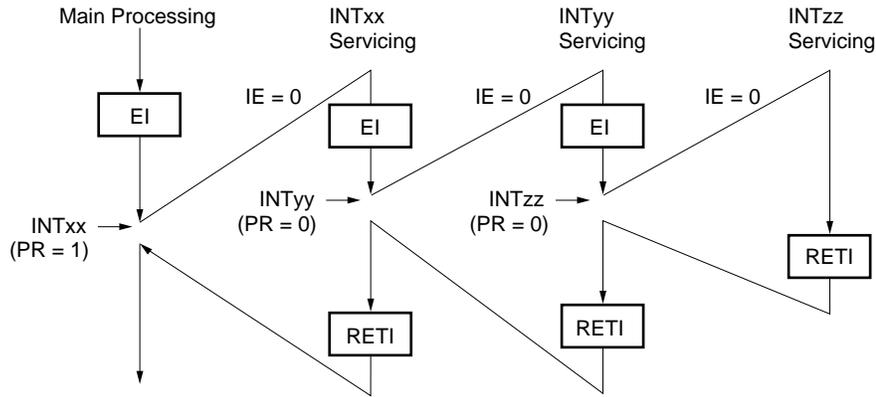
Table 22-4. Interrupt Request Enabled for Multiple Interrupt during Interrupt Servicing

Multiple Interrupt Request Interrupt being Serviced		Non-maskable Interrupt Request	Maskable Interrupt Request			
			xxPR = 0		xxPR = 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		E	E	D	E	D

- Remarks**
1. E : Multiple interrupt enable
 2. D : Multiple 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. xxPR is a flag contained in PR0L, PR0H, and PRIL
 - xxPR = 0 : Higher priority level
 - xxPR = 1 : Lower priority level

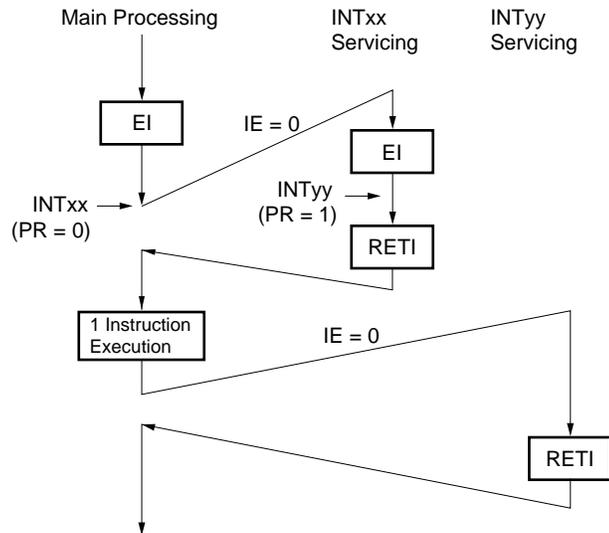
Figure 22-16. Multiple Interrupt Example (1/2)

Example 1. Two multiple interrupts generated



During interrupt INTxx servicing, two interrupt requests, INTyy and INTzz are acknowledged, and a multiple interrupt is generated. An EI instruction is issued before each interrupt request acknowledge, and the interrupt request acknowledge enable state is set.

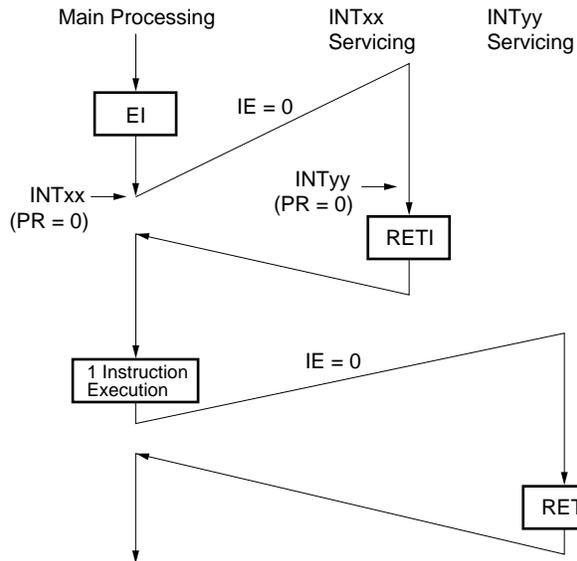
Example 2. Multiple interrupt is not generated by priority control



The interrupt request INTyy generated during interrupt INTxx servicing is not acknowledged because the interrupt priority is lower than that of INTxx, and a multiple interrupt is not generated. INTyy request is retained and acknowledged after execution of 1 instruction execution of the main processing.

- PR = 0 : Higher priority level
- PR = 1 : Lower priority level
- IE = 0 : Interrupt request acknowledge disable

Figure 22-16. Multiple Interrupt Example (2/2)

Example 3. A multiple interrupt is not generated because interrupts are not enabled

Because interrupts are not enabled in interrupt INTxx servicing (an EI instruction is not issued), interrupt request INTyy is not acknowledged, and a multiple interrupt is not generated. The INTyy request is reserved and acknowledged after 1 instruction execution of the main processing.

PR = 0 : Higher priority level

IE = 0 : Interrupt request acknowledge disable

22.4.5 Interrupt request reserve

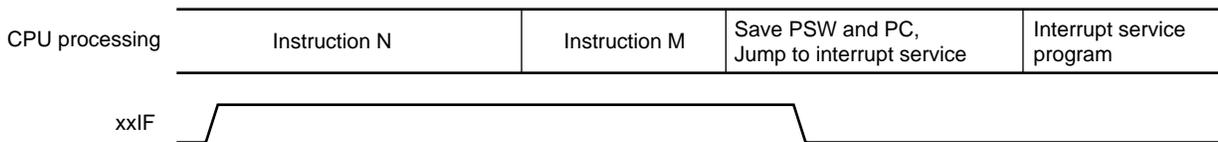
Some instructions may reserve the acknowledge of an instruction request until the completion of the execution of the next instruction even if the interrupt request is generated during the execution. The following shows such instructions (interrupt request reserve instruction).

- 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/CLR1 PSW.bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW.bit, \$addr16
- BF PSW.bit, \$addr16
- BTCLR PSW.bit, \$addr16
- EI
- DI
- Manipulate instructions for IF0L, IF0H, IF1L, MK0L, MK0H, MK1L, PR0L, PR0H, PR1L, INTM0, INTM1 registers

Caution BRK instruction is not an interrupt request reserve instruction described above. However, in a software interrupt started by the execution of BRK instruction, the IE flag is cleared to 0. Therefore, interrupt requests are not acknowledged even when a maskable interrupt request is issued during the execution of the BRK instruction. However, non-maskable interrupt requests are acknowledged.

Figure 22-17 shows the interrupt request hold timing.

Figure 22-17. Interrupt Request Hold



- Remarks**
1. Instruction N: Instruction that holds interrupts requests
 2. Instruction M: Instructions other than interrupt request pending instruction
 3. The xxPR (priority level) values do not affect the operation of xxIF (interrupt request).

22.5 Test Functions

In this function, when the watch timer overflows and when a rising edge of port 4 is detected, the corresponding test input flag is set (1), and a standby release signal is generated.

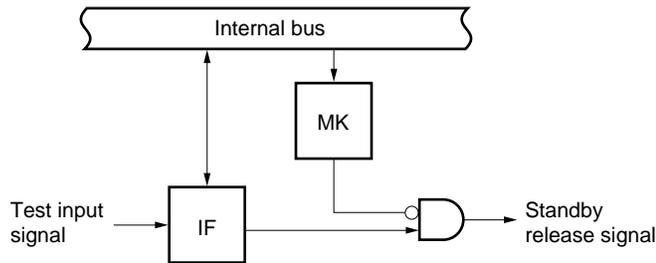
Unlike the interrupt function, vectored processing is not performed.

There are two test input factors as shown in Table 22-5. The basic configuration is shown in Figure 22-18.

Table 22-5. Test Input Factors

Test Input Factors		Internal/ external
Name	Trigger	
INTWT	Clock timer overflow	Internal
INTPT4	Negative edge detection at port 4	External

Figure 22-18. Basic Configuration of Test Function



IF : Test input flag
MK: Test mask flag

22.5.1 Registers controlling the test function

The test function is controlled by the following three registers.

- Interrupt request flag register 1L (IF1L)
- Interrupt mask flag register 1L (MK1L)
- Key return mode register (KRM)

The names of the test input flag and test mask flag corresponding to the test input signals are listed in Table 22-6.

Table 22-6. Flags Corresponding to Test Input Signals

Test Input Signal Name	Test Input Flag	Test Mask Flag
INTWT	WTIF	WTMK
INTPT4	KRIF	KRMK

(1) Interrupt request flag register 1L (IF1L)

It indicates whether a clock timer overflow is detected or not.

It is set by a 1-bit memory manipulation instruction and 8-bit memory manipulation instruction.

It is set to 00H by the $\overline{\text{RESET}}$ signal input.

Figure 22-19. Format of Interrupt Request Flag Register 1L

Symbol	<7>	6	5	<4>	<3>	<2>	<1>	<0>	Address	After Reset	R/W
IF1L	WTIF	0	0	TMIF6	TMIF5	ADIF	TMIF2	TMIF1	FFE2H	00H	R/W

WTIF	Clock timer overflow detection flag
0	Not detected
1	Detected

Caution Set bits 5 and 6 to 0.

(2) Interrupt mask flag register 1L (MK1L)

It is used to set the standby mode enable/disable at the time the standby mode is released by the clock timer.

It is set by a 1-bit memory manipulation instruction and 8-bit memory manipulation instruction.

It is set to FFH by the $\overline{\text{RESET}}$ signal input.

Figure 22-20. Format of Interrupt Mask Flag Register 1L

Symbol	<7>	6	5	<4>	<3>	<2>	<1>	<0>	Address	After Reset	R/W
MK1L	WTMK	1	1	TMMK6	TMMK5	ADMK	TMMK2	TMMK1	FFE6H	FFH	R/W

WTMK	Standby mode control by clock timer
0	Enables releasing the standby mode.
1	Disables releasing the standby mode.

Caution Set bits 5 and 6 to 1.

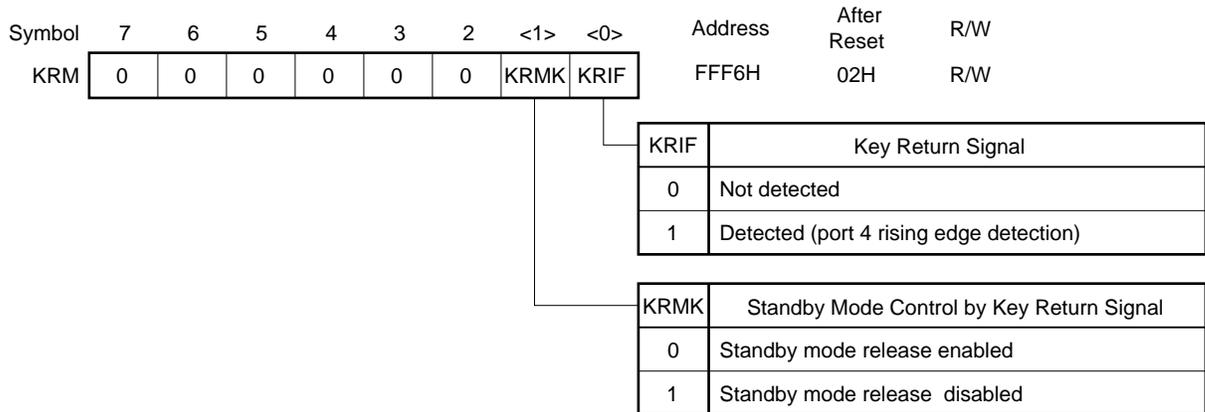
(3) Key return mode register (KRM)

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 22-21. Key Return Mode Register Format



Caution When port 4 falling edge detection is used, clear KRIF to 0 (not cleared to 0 automatically).

22.5.2 Test input signal acknowledge operation

(1) Internal test signal

If the watch timer overflows, an internal test input signal (INTWT) is generated, by which the WTIF flag is set. At this time, if it is not masked with the interrupt mask flag (WTMK), a standby release signal is generated. The watch function is available by checking the WTIF flag at a shorter cycle than the watch timer overflow cycle.

(2) External test signal

When a falling edge is input to the port 4 (P40 to P47) pins, an external test input signal (INTPT4) is generated, by which the KRIF flag is set. At this time, if it is not masked with the interrupt mask flag (KRMK), a standby release signal is generated. If port 4 is used as key matrix return signal input, whether or not a key input has been applied can be checked from the KRIF status.

[MEMO]

CHAPTER 23 EXTERNAL DEVICE EXPANSION FUNCTION

23.1 External Device Expansion Functions

The external device expansion functions connect external devices to areas other than the internal ROM, RAM, and SFR.

The external device expansion function can be used in the following two modes:

- Multiplexed bus mode
- Separate bus mode

(1) Multiplexed bus mode

External devices are connected via the multiplexed address/data bus. Connecting external devices using this mode may reduce the number of ports to be used. In this mode, ports 4 through 6 are used for control of address/data, read/write strobe, wait, address strobe as shown below.

Table 23-1. Pin Functions in External Memory Expansion Mode

Pin function at external device connection		Alternate function
Name	Function	
AD0 to AD7	Multiplexed address/data bus	P40 to P47
A8 to A15	Upper Address bus	P50 to P57
$\overline{\text{RD}}$	Read strobe signal	P64
$\overline{\text{WR}}$	Write strobe signal	P65
$\overline{\text{WAIT}}$	Wait signal	P66
ASTB	Address strobe signal	P67

Table 23-2. State of Port 4 to Port 6 Pins in External Memory Expansion Mode

Ports and bits Modes	Port 4	Port 5		Port 6	
	0 to 7	0	1 2 3 4 5 6 7	0 to 3	4 to 7
Single-chip mode	Port	Port		Port	Port
256-byte expansion mode	Address/data	Port		Port	$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB
4-Kbyte expansion mode	Address/data	Address	Port	Port	$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB
16-Kbyte expansion mode	Address/data	Address	Port	Port	$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB
Full address mode	Address/data	Address		Port	$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB

Caution When the external wait function is not used, the $\overline{\text{WAIT}}$ pin can be used as a port in all modes.

(2) Separate bus mode

External devices are connected using independent address and data buses. This connection requires no latches externally, resulting in reduction of external parts and area on the mounting board. In this mode, ports 4 through 6 and port 8 are used for control of address/data, read/write strobe, wait as shown below.

Table 23-3. Pin Functions in Separate Bus Mode

Pin function in separate bus mode		Alternate function
Name	Function	
AD0 to AD7	Multiplexed address/data bus	P40 to P47
A0 to A7	Lower Address bus	P80 to P87
A8 to A15	Upper Address bus	P50 to P57
\overline{RD}	Read strobe signal	P64
\overline{WR}	Write strobe signal	P65
\overline{WAIT}	Wait signal	P66

Caution In this mode, the address strobe signal is not required to be used though it is output from the ASTB/P67 pin. The output timings are shown in Figures 23-9 through 23-12.

Table 23-4. State of Port 4 to Port 6 and Port 8 Pins in Separate Bus Mode

Ports and bits	Port 4	Port 8	Port 5							Port 6		
	0 to 7	0 to 7	0	1	2	3	4	5	6	7	0 to 3	4 to 7
Modes												
Single-chip mode	Port	Port	Port							Port	Port	
256-byte expansion mode	Data	Address	Port							Port	\overline{RD} , \overline{WR} , \overline{WAIT} , (ASTB)	
4-Kbyte expansion mode	Data	Address	Address	Port			Port				\overline{RD} , \overline{WR} , \overline{WAIT} , (ASTB)	
16-Kbyte expansion mode	Data	Address	Address			Port				Port	\overline{RD} , \overline{WR} , \overline{WAIT} , (ASTB)	
Full address mode	Data	Address	Address							Port	\overline{RD} , \overline{WR} , \overline{WAIT} , (ASTB)	

Cautions 1. When the external wait function is not used, the \overline{WAIT} pin can be used as a port in all modes.
 2. In this mode, the address strobe signal is not required to be used though it is output from the ASTB/P67 pin. The output timings are shown in Figures 23-9 through 23-12.

Memory maps when using the external device expansion function are as follows.

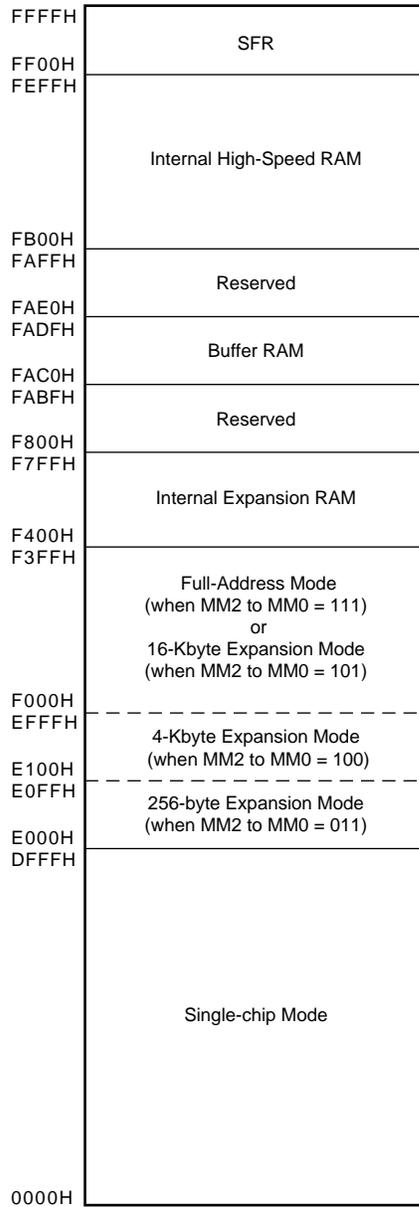
Figure 23-1. Memory Map when Using External Device Expansion Function (1/2)

**(a) Memory map of μ PD78076, 78076Y,
and of μ PD78P078, 78P078Y when
internal PROM capacity is 48 Kbytes**

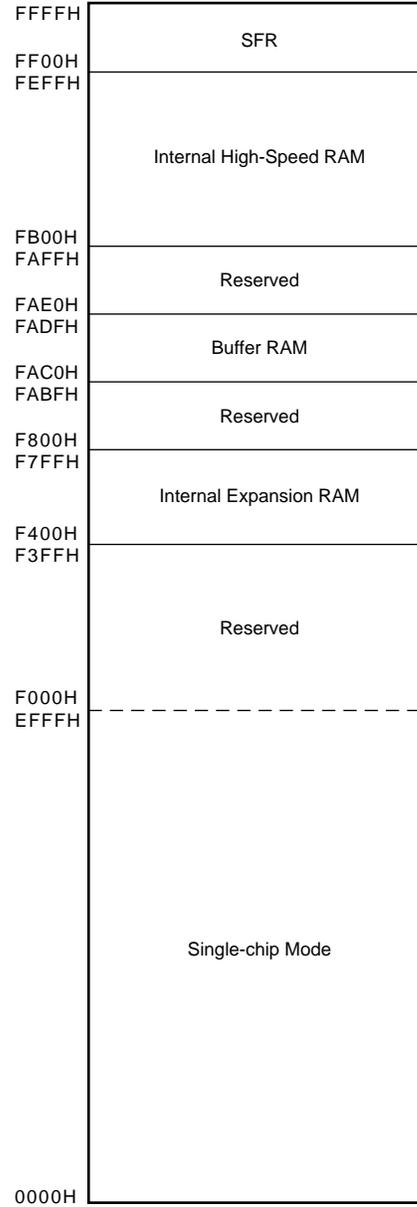
FFFFH	SFR
FF00H FEFFH	
	Internal High-Speed RAM
FB00H FAFFH	Reserved
FAE0H FADFH	Buffer RAM
FAC0H FABFH	Reserved
F800H F7FFH	Internal Expansion RAM
F400H F3FFH	Full-Address Mode (when MM2 to MM0 = 111) or 16-Kbyte Expansion Mode (when MM2 to MM0 = 101)
D000H CFFFH	4-Kbyte Expansion Mode (when MM2 to MM0 = 100)
C100H C0FFH	256-byte Expansion Mode (when MM2 to MM0 = 011)
C000H BFFFH	Single-chip Mode
0000H	

Figure 23-1. Memory Map when Using External Device Expansion Function (2/2)

(b) Memory map of μ PD78078, 78078Y, 78P078, 78P078Y when internal ROM capacity (PROM) is 56 Kbytes



(c) Memory map of μ PD78078, 78078Y and of μ PD78P078, 78P078Y when internal PROM capacity is 60 Kbytes



Caution When the internal ROM (PROM) size is 60 Kbytes, the area from F000H to F3FFH cannot be used. F000H to F3FFH can be used as external memory by setting the internal ROM size to less than 56 Kbytes by the internal memory size switching register (IMS).

23.2 External Device Expansion Function Control Register

The external device expansion function is controlled by the memory expansion mode register (MM) and internal memory size switching register (IMS).

(1) Memory expansion mode register (MM)

MM sets the wait count and external expansion area, and also sets the input/output of port 4.

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

$\overline{\text{RESET}}$ input sets this register to 10H.

Figure 23-2. Memory Expansion Mode Register Format

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

MM2	MM1	MM0	Single-chip/ Memory Expansion Mode Selection		P40 to P47, P50 to P57, P64 to P67, P80 to P87 Pin state					
					P40 to P47		P80 to P87	P50 to P53	P54, P55	P56, P57
0	0	0	Single-chip mode		Port mode	Input	Port mode			
0	0	1								
0	1	1	Memory expansion mode	256-byte mode	AD0 to AD7	A0 to A7 ^{Note 1}	Port mode			P64 = $\overline{\text{RD}}$ P65 = $\overline{\text{WR}}$ P66 = $\overline{\text{WAIT}}$ P67 = $\overline{\text{ASTB}}$
1	0	0		4-Kbyte mode			A8 to A11	Port mode		
1	0	1		16-Kbyte mode				A12, A13	Port mode	
1	1	1		Full address mode ^{Note 2}			A14, A15		Port mode	
Other than above			Setting prohibited							

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

- Notes**
- This can be used only in the separate bus mode. In the multiplexed bus mode, these pins enter the port mode.
 - The full address mode allows external expansion to the entire 64-Kbyte address space except for the internal ROM, RAM, and SFR areas and the reserved area.

Remark P60 to P63 enter the port mode without regard to the mode (single-chip mode or memory expansion mode).

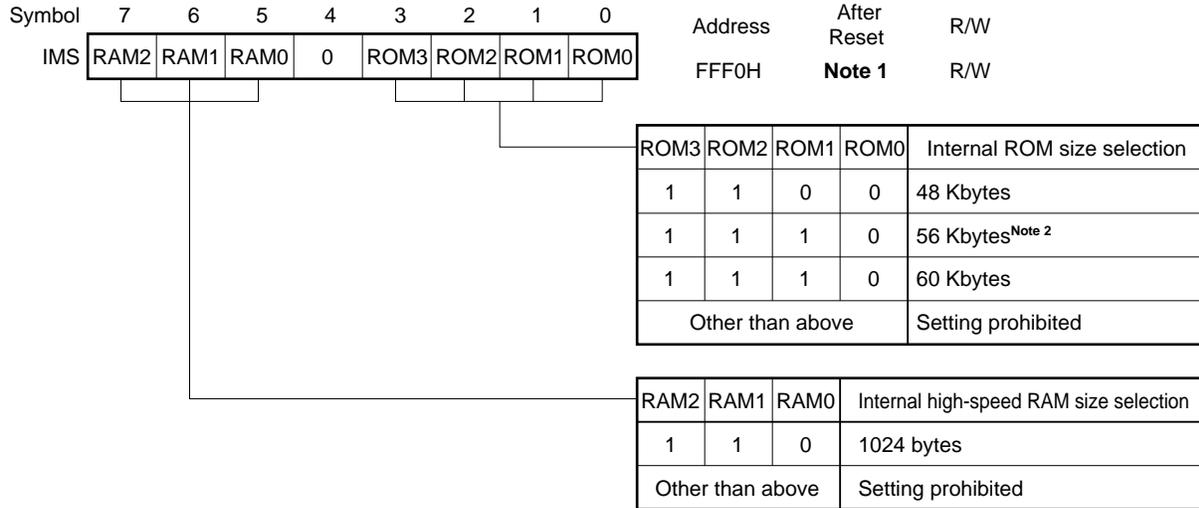
(2) Internal memory size switching register (IMS)

The μ PD78P078, 78P078Y can specify the internal memory size by internal memory size switching register (IMS).

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

$\overline{\text{RESET}}$ input sets this register to the value indicated in Table 23-5.

Figure 23-3. Internal Memory Size Switching Register Format



Notes 1. The values after reset depend on the product. (See Table 23-5)

Table 23-5. Values when the Internal Memory Size Switching Register is Reset

Type Number	Reset Value
μ PD78076, 78076Y	CCH
μ PD78078, 78078Y	CFH
μ PD78P078, 78P078Y	

2. When using the external device expansion function in the μ PD78078, 78078Y, 78P078, and 78P078Y set the internal ROM size to 56 Kbytes.

Caution When using the mask ROM products, do not set the values other than CCH and CFH to the IMS except for the cases where the external device extension function is used in μ PD78078 and 78078Y.

Remark IMS should be set according to the internal ROM capacity.

(3) External bus type select register (EBTS)

This register sets the operation mode of the external device expansion function. When the multiplexed bus mode is selected, the P80/A0 through P87/A7 pins can be used as an I/O port.

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

RESET signal input sets EBTS to 00H.

Figure 23-4. External Bus Type Select Register Format



23.3 External Device Expansion Function Timing

23.3.1 Timings in multiplexed bus mode

Timing control signal output pins in the multiplexed bus mode are as follows.

(1) $\overline{\text{RD}}$ pin (Alternate function: P64)

Read strobe signal output pin. The read strobe signal is output in data accesses and instruction fetches from external memory.

During internal memory access, the read strobe signal is not output (maintains high level).

(2) $\overline{\text{WR}}$ pin (Alternate function: P65)

Write strobe signal output pin. The write strobe signal is output in data access to external memory.

During internal memory access, the write strobe signal is not output (maintains high level).

(3) $\overline{\text{WAIT}}$ pin (Alternate function: P66)

External wait signal input pin. When the external wait is not used, the $\overline{\text{WAIT}}$ pin can be used as an input/output port.

During internal memory access, the external wait signal is ignored.

(4) ASTB pin (Alternate function: P67)

Address strobe signal output pin. Timing signal is output without regard to the data accesses and instruction fetches from external memory. The ASTB signal is also output when the internal memory is accessed.

(5) $\text{AD0 to AD7, A8 to A15}$ pins (Alternate function : P40 to P47, P80 to P87, P50 to P57)

Address/data signal output pin. Valid signal is output or input during data accesses and instruction fetches from external memory.

These signals change when the internal memory is accessed (output values are undefined).

Timing charts are shown in Figure 23-5 to 23-8.

Figure 23-5. Instruction Fetch from External Memory in Multiplexed Bus Mode

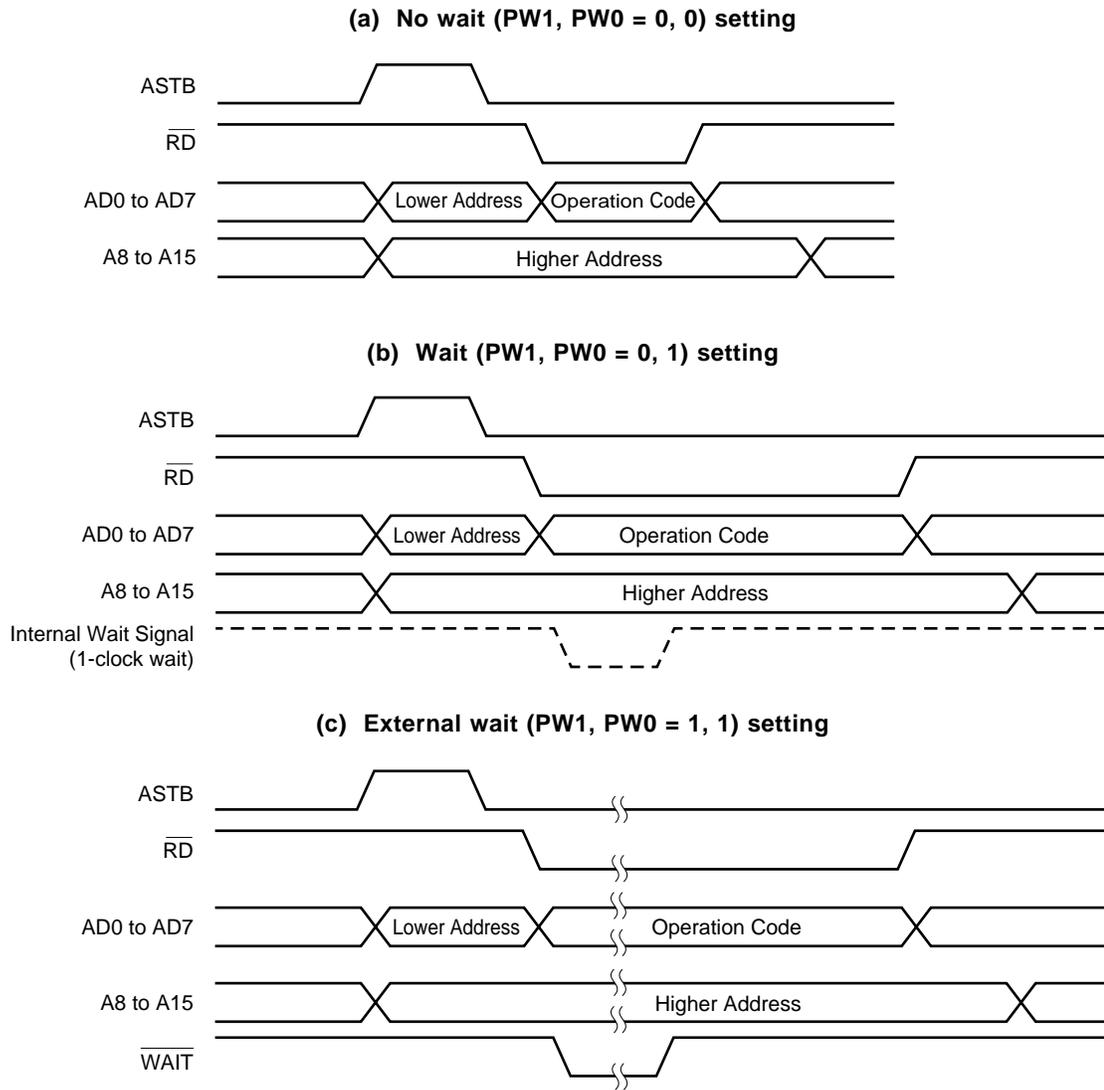


Figure 23-6. External Memory Read Timing in Multiplexed Bus Mode

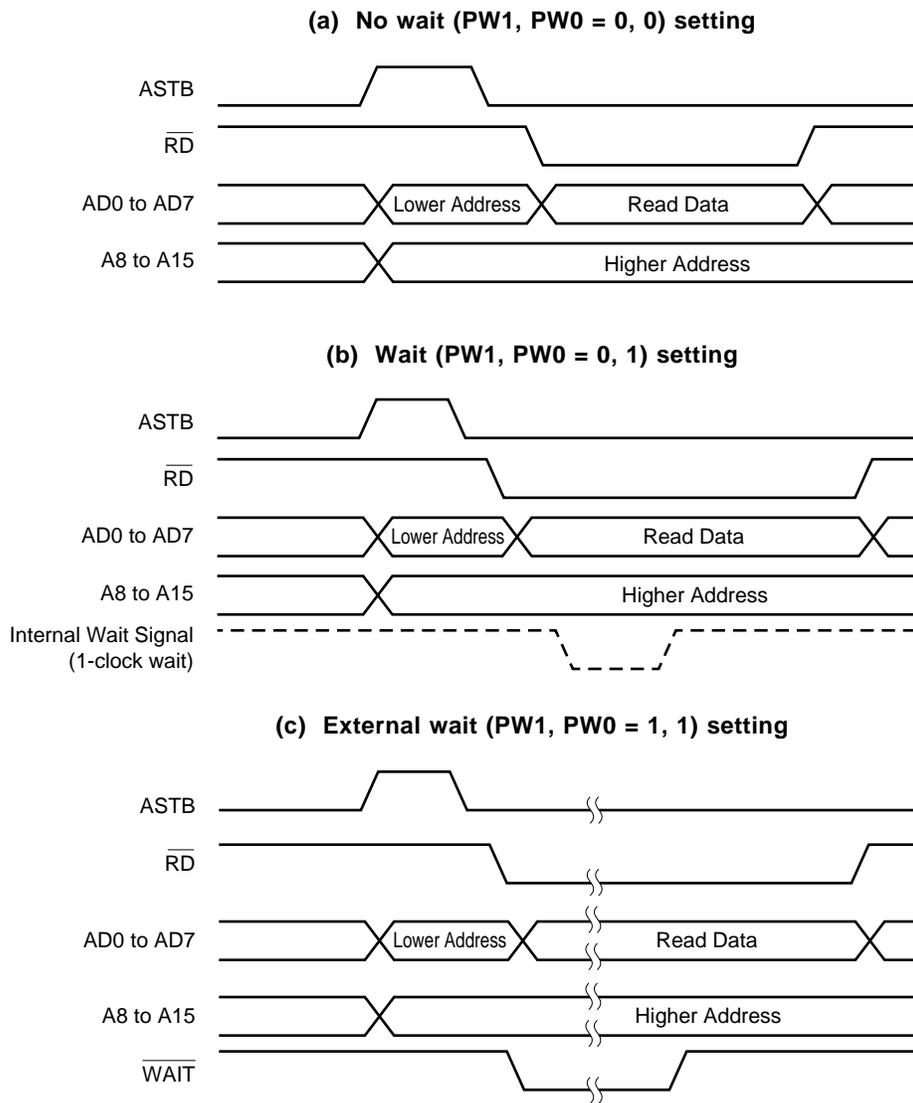
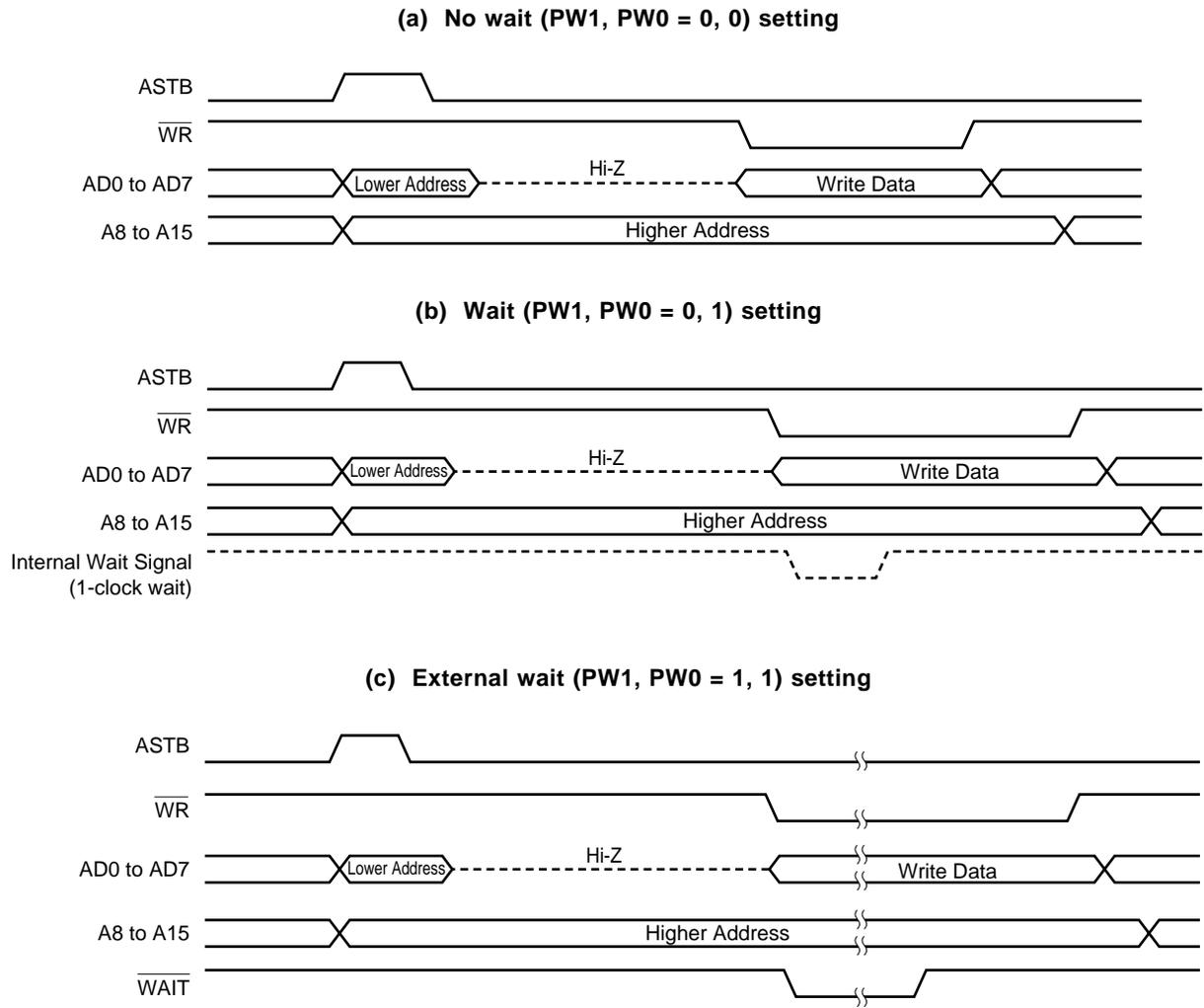


Figure 23-7. External Memory Write Timing in Multiplexed Bus Mode



23.3.2 Timings in separate bus mode

Timing control signal output pins in the separate bus mode are as follows.

(1) $\overline{\text{RD}}$ pin (Alternate function: P64)

Read strobe signal output pin. The read strobe signal is output in data accesses and instruction fetches from external memory.

During internal memory access, the read strobe signal is not output (maintains high level).

(2) $\overline{\text{WR}}$ pin (Alternate function: P65)

Write strobe signal output pin. The write strobe signal is output in data access to external memory.

During internal memory access, the write strobe signal is not output (maintains high level).

(3) $\overline{\text{WAIT}}$ pin (Alternate function: P66)

External wait signal input pin. When the external wait is not used, the $\overline{\text{WAIT}}$ pin can be used as an input/output port.

During internal memory access, the external wait signal is ignored.

(4) AD0 to AD7, A0 to A7, A8 to A15 pins (Alternate function : P40 to P47, P80 to P87, P50 to P57)

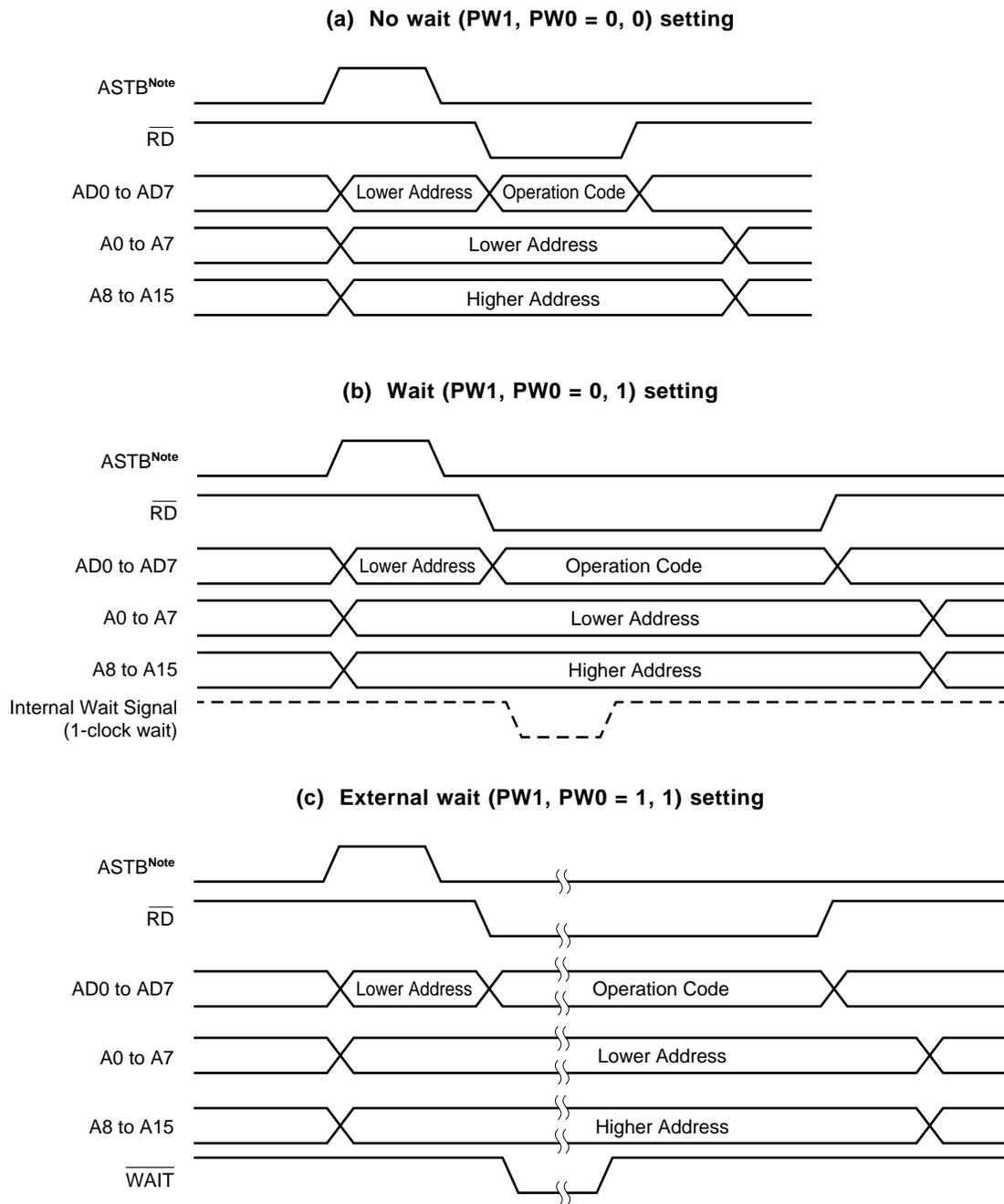
Address/data signal output pin. Valid signal is output or input during data accesses and instruction fetches from external memory.

These signals change when the internal memory is accessed (output values are undefined).

Timing charts are shown in Figure 23-9 to 23-12.

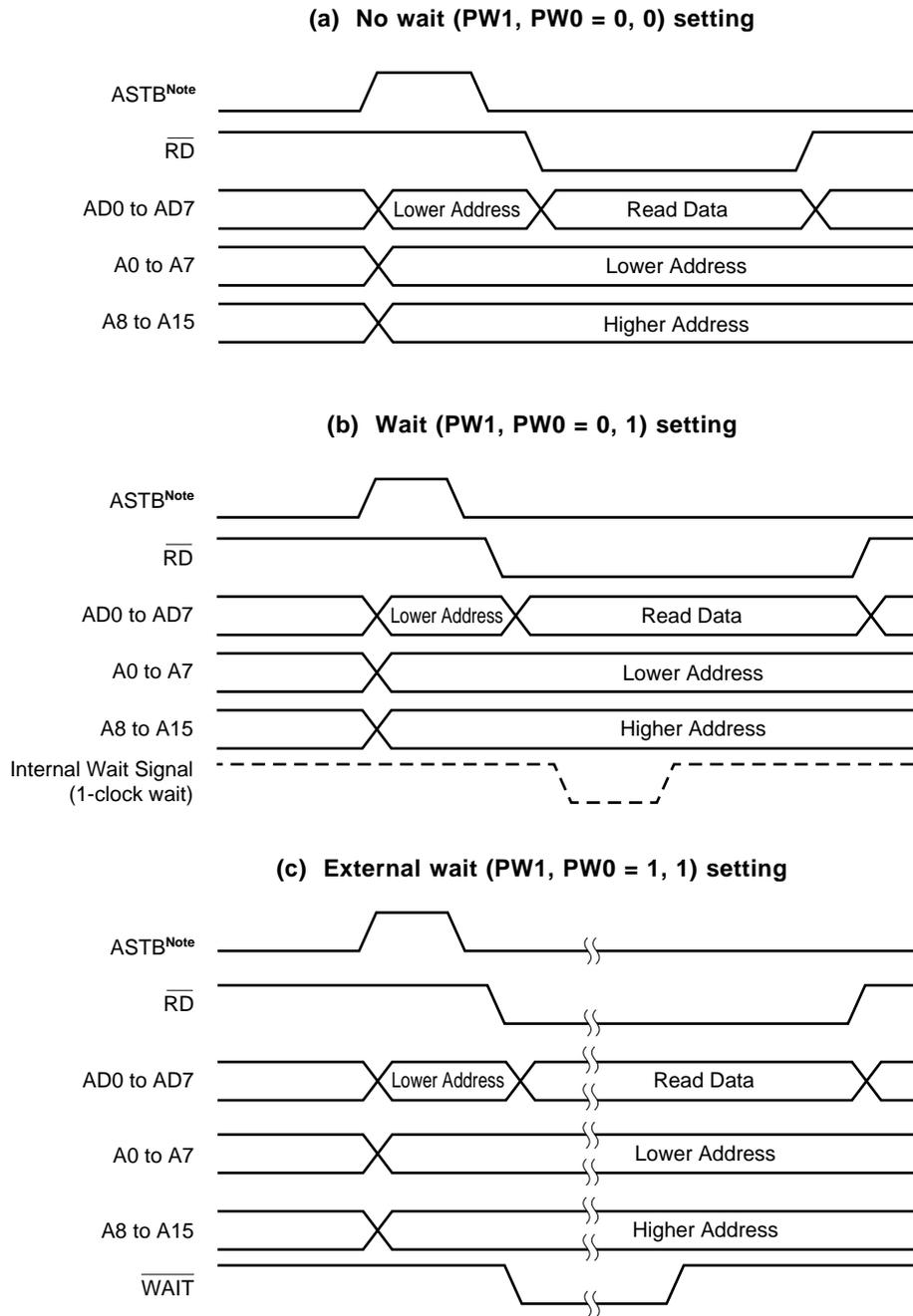
Caution In the separate bus mode, use of the address strobe signal is not required though it is output from the ASTB/P67 pin. The output timings of the signals are shown in Figure 23-9 to 23-12.

Figure 23-9. Instruction Fetch from External Memory in Separate Bus Mode



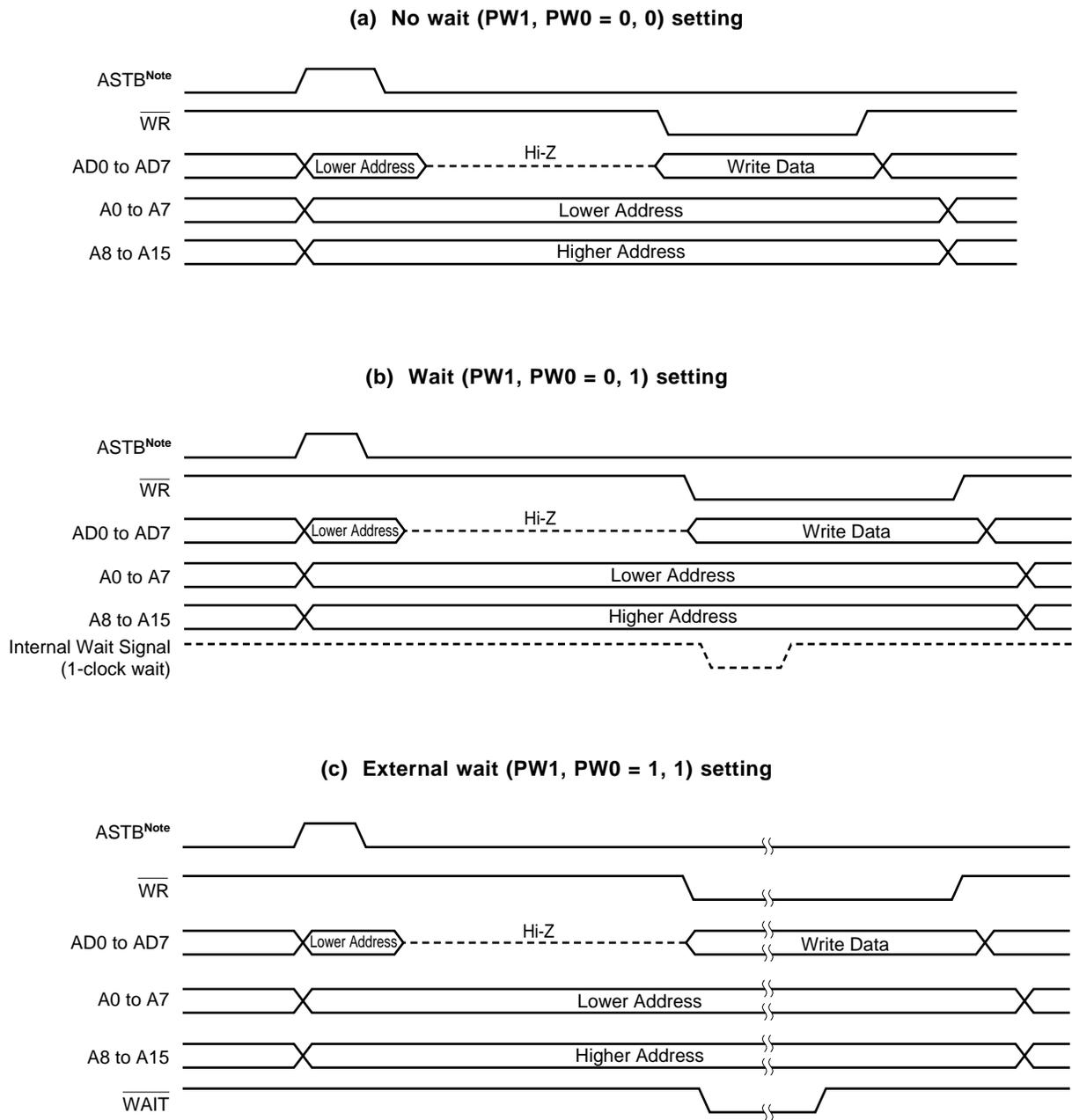
Note In the separate bus mode, use of the address strobe signal is not required though it is output from the ASTB/P67 pin.

Figure 23-10. External Memory Read Timing in Separate Bus Mode



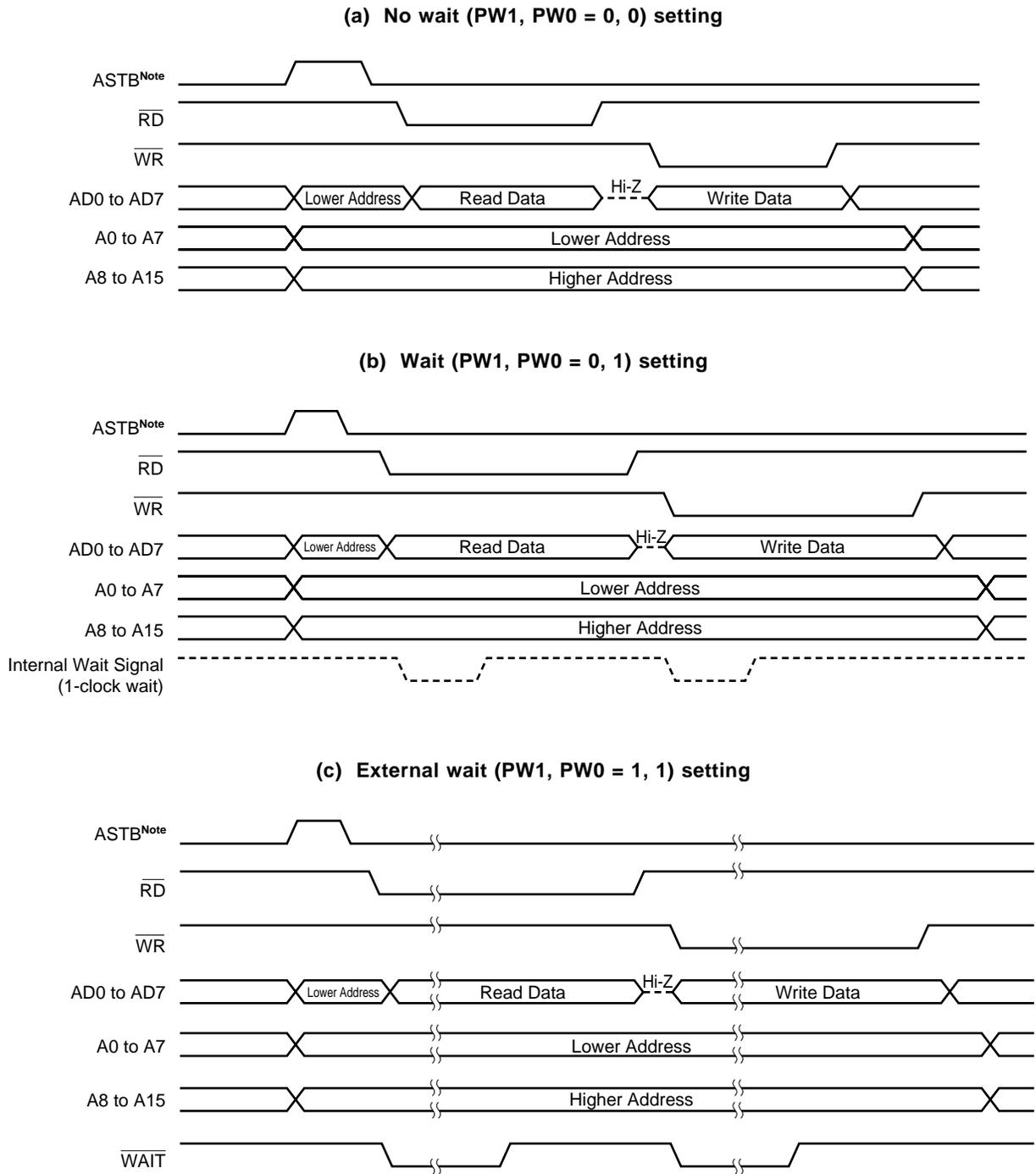
Note In the separate bus mode, use of the address strobe signal is not required though it is output from the ASTB/P67 pin.

Figure 23-11. External Memory Write Timing in Separate Bus Mode



Note In the separate bus mode, use of the address strobe signal is not required though it is output from the $\overline{\text{ASTB}}/\text{P67}$ pin.

Figure 23-12. External Memory Read Modify Write Timing in Separate Bus Mode



Note In the separate bus mode, use of the address strobe signal is not required though it is output from the ASTB/P67 pin.

[MEMO]

24.1 Standby Function and Configuration

24.1.1 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 watch applications.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the main system clock oscillator stops and the whole system stops. CPU current consumption can be considerably decreased.

Data memory low-voltage hold (down to $V_{DD} = 1.8$ 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.

- Cautions**
- 1. The STOP mode can be used only when the system operates with the main system clock (subsystem clock oscillation cannot be stopped). The HALT mode can be used with either the main system clock or the subsystem clock.**
 - 2. When proceeding to the STOP mode, be sure to stop the peripheral hardware operation and execute the STOP instruction.**
 - 3. The following sequence is recommended for power consumption reduction of the A/D converter when the standby function is used: first clear bit 7 (CS) to 0 to stop the A/D conversion operation, and then execute the HALT or STOP instruction.**

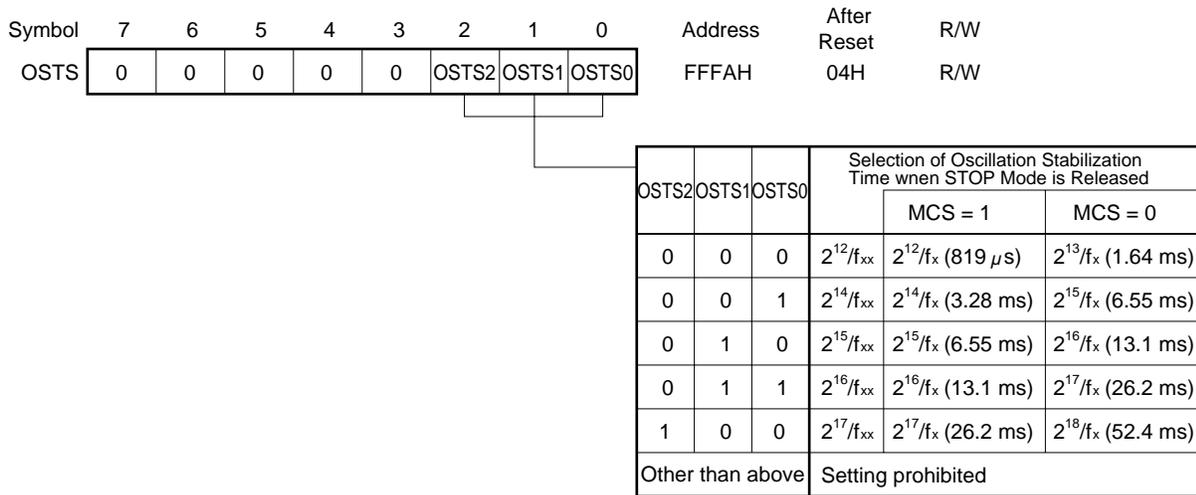
24.1.2 Standby function control register

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).

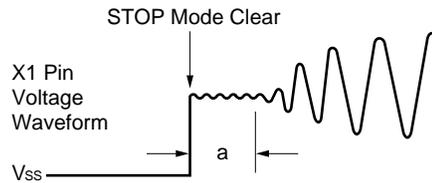
OSTS is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ 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 24-1. Oscillation Stabilization Time Select Register Format



Caution The wait time after STOP mode clear does not include the time from STOP mode clear to clock oscillation start (see “a” in the illustration below), whether the STOP mode is cleared by $\overline{\text{RESET}}$ input or by interrupt request generation.



- Remarks**
1. f_{xx} : Main system clock frequency (f_x or $f_x/2$)
 2. f_x : Main system clock oscillation frequency
 3. MCS : Bit 0 of oscillation mode select register (OSMS)
 4. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

24.2 Standby Function Operations

24.2.1 HALT mode

(1) HALT mode set and operating status

The HALT mode is set by executing the HALT instruction. It can be set with the main system clock or the subsystem clock. The operating status in the HALT mode is described below.

Table 24-1. HALT Mode Operating Status

Item	HALT mode setting		HALT execution during main system clock operation		HALT execution during subsystem clock operation	
	w/ subsystem clock ^{Note 1}	w/o. subsystem clock ^{Note 2}	Main system clock oscillates	Main system clock stops		
Clock Generator	Both main system and subsystem clocks can be oscillated. Clock supply to the CPU stops.					
CPU	Operation stops.					
Port (output latch)	Status before HALT mode setting is held.					
16-bit timer/event counter	Operable.				Operable when watch timer output is used as count clock (with fXT selected as count clock of watch timer).	
8-bit timer/event counter 1 and 2	Operable.				Operable when TI1 or TI2 is selected as count clock.	
8-bit timer/event counter 5 and 6	Operable.				Operable when TI5 or TI6 is selected as count clock.	
Watch timer	Operable when fxx/2 ⁷ is selected as count clock.	Operable.			Operable when fXT is selected as count clock.	
Watchdog timer	Operable.			Operation stops.		
A/D converter	Operable.				Operation stops.	
D/A converter	Operable.					
Real-time output port	Operable.					
Serial Interface						
When a function other than auto transmit/receive is used	Operable				Operable at external \overline{SCK} .	
When auto transmit/receive function is used	Operation stops.					
External interrupt						
INTP0	Operable when a clock (fxx/2 ⁵ , fxx/2 ⁶ , fxx/2 ⁷) for the peripheral hardware is selected as sampling clock.				Operation stops.	
INTP1 to INTP6	Operable.					
Bus lines in external expansion						
AD0 to AD7	Enters high impedance state.					
A0 to A15	Holds the state before HALT mode is set.					
ASTB	Outputs low level.					
\overline{WR} , \overline{RD}	Outputs high level.					
\overline{WAIT}	Enters high impedance state.					

Notes 1. Including case when external clock is supplied.

2. Including case when external clock is not supplied.

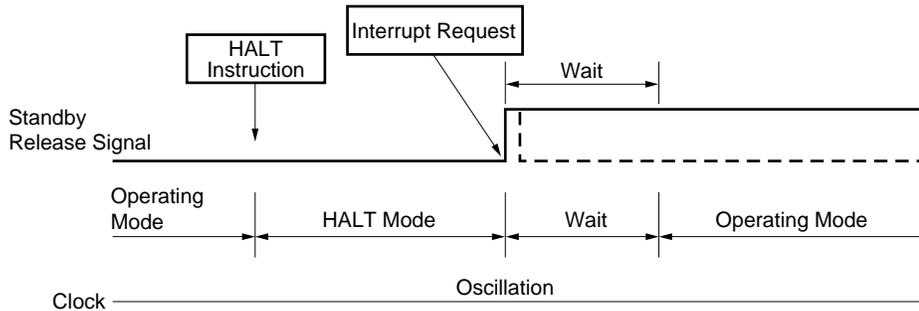
(2) HALT mode release

The HALT mode can be released with the following four types of sources.

(a) Release by unmasked interrupt request

An unmasked interrupt request is generated to release the HALT mode. If interrupt request acknowledge is enabled, vectored interrupt service is carried out. If disabled, the next address instruction is executed.

Figure 24-2. HALT Mode Released by Interrupt Request Generation



Remarks 1. The broken line indicates the case when the interrupt request which has released the standby status is acknowledged.

2. Wait time will be as follows:

- When vectored interrupt service is carried out: 8 to 9 clocks
- When vectored interrupt service is not carried out: 2 to 3 clocks

(b) Release by non-maskable interrupt request

The HALT mode is released and vectored interrupt service is carried out if non-maskable interrupt request is generated whether interrupt request acknowledge is enabled or disabled.

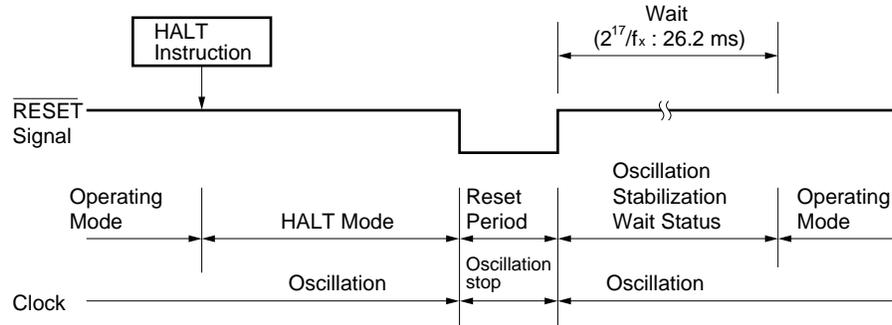
(c) Release by unmasked test input

The HALT mode is released by inputting an unmasked test signal and the next address instruction of the HALT instruction is executed.

(d) Release by $\overline{\text{RESET}}$ input

As is the case with normal reset operation, a program is executed after branch to the reset vector address.

Figure 24-3. HALT Mode Released by $\overline{\text{RESET}}$ Input



- Remarks**
1. f_x : Main system clock oscillation frequency
 2. Figures in parentheses apply to operation with $f_x = 5.0 \text{ MHz}$.

Table 24-2. Operation after HALT Mode Release

Release Source	MKxx	PRxx	IE	ISP	Operation
Maskable interrupt request	0	0	0	x	Next address instruction execution
	0	0	1	x	Interrupt service execution
	0	1	0	1	Next address instruction execution
	0	1	x	0	
	0	1	1	1	Interrupt service execution
	1	x	x	x	HALT mode hold
Non-maskable interrupt request	–	–	x	x	Interrupt service execution
Test input	0	–	x	x	Next address instruction execution
	1	–	x	x	HALT mode hold
$\overline{\text{RESET}}$ input	–	–	x	x	Reset processing

x: Don't care

24.2.2 STOP mode

(1) STOP mode set and operating status

The STOP mode is set by executing the STOP instruction. It can be set only with the main system clock.

Cautions 1. When the STOP mode is set, the X2 pin is internally connected to V_{DD} via a pull-up resistor to minimize leakage current at the crystal oscillator. Thus, do not use the STOP mode in a system where an external clock is used for the main system clock.

2. Because the interrupt request signal is used to release the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared 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 24-3. STOP Mode Operating Status

STOP Mode Setting Item	With Subsystem Clock	Without Subsystem Clock
Clock Generator	Only main system clock stops oscillation.	
CPU	Operation stops.	
Port (output latch)	Status before STOP mode setting is held.	
16-bit timer/event counter	Operable when watch timer output is used as count clock (f _{XT} is selected as count clock for watch timer).	Operation stops.
8-bit timer/event counters 1 and 2	Operable when T11 or T12 is selected for the count clock.	
8-bit timer/event counters 5 and 6	Operable when T15 or T16 is selected for the count clock.	
Watch timer	Operable when f _{XT} is selected for the count clock.	Operation stops.
Watchdog timer	Operation stops.	
A/D converter	Operation stops.	
D/A converter	Operable.	
Real-time output port	Operable when external trigger is used, or T11 or T12 is selected for the 8-bit timer/event counter 1 or 2 count clock.	
Serial Interface		
When a function other than auto transmit/receive & UART is used	Operable only when externally supplied clock is specified as the serial clock.	
When auto transmit/receive function or UART is used	Operation stops.	
External interrupt		
INTP0	Operation disabled.	
INTP1 to INTP6	Operable.	
Bus lines in external expansion		
AD0 to AD7	Enters high-impedance state.	
A0 to A15	Holds the state before STOP mode is set.	
ASTB	Outputs low level.	
\overline{WR} , \overline{RD}	Outputs high level.	
\overline{WAIT}	Enters high-impedance state.	

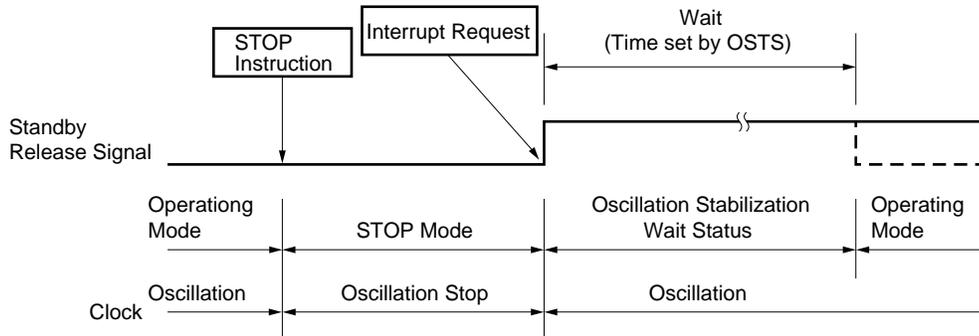
(2) STOP mode release

The STOP mode can be released with the following three types of sources.

(a) Release by unmasked interrupt request

An unmasked interrupt request is used to release the STOP mode. 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 24-4. STOP Mode Released by Interrupt Request Generation



Remark The broken line indicates the case when the interrupt request which has cleared the standby status is acknowledged.

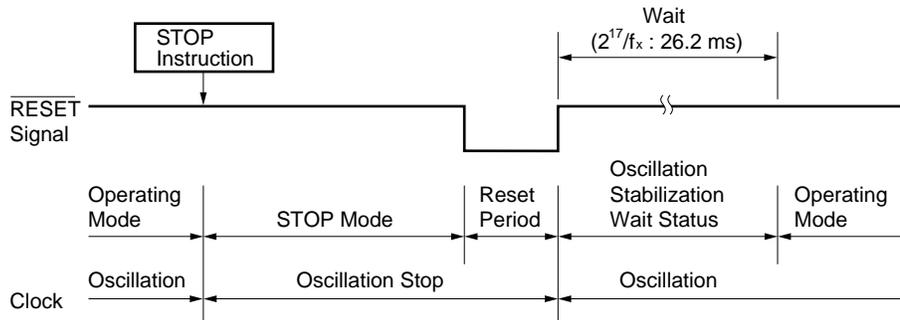
(b) Release by unmasked test input

The STOP mode is released by inputting an unmasked test signal. 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

The STOP mode is released and after the lapse of oscillation stabilization time, reset operation is carried out.

Figure 24-5. STOP Mode Released by $\overline{\text{RESET}}$ Input



- Remarks**
1. f_x : Main system clock oscillation frequency
 2. Figures in parentheses apply to operation with $f_x = 5.0$ MHz.

Table 24-4. Operation after STOP Mode Release

Release Source	MKxx	PRxx	IE	ISP	Operation
Maskable interrupt request	0	0	0	x	Next address instruction execution
	0	0	1	x	Interrupt service execution
	0	1	0	1	Next address instruction execution
	0	1	x	0	
	0	1	1	1	Interrupt service execution
	1	x	x	x	STOP mode hold
Test input	0	–	x	x	Next address instruction execution
	1	–	x	x	STOP mode hold
$\overline{\text{RESET}}$ input	–	–	x	x	Reset processing

x: Don't care

CHAPTER 25 RESET FUNCTION

25.1 Reset Function

The following two operations are available to generate the reset signal.

- (1) External reset input with $\overline{\text{RESET}}$ pin
- (2) Internal reset by watchdog timer overrun time detection

External reset and internal reset have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H by $\overline{\text{RESET}}$ input.

When a low level is input to the $\overline{\text{RESET}}$ pin or the watchdog timer overflows, a reset is applied and each hardware is set to the status as shown in Table 25-1. Each pin has high impedance during reset input or during oscillation stabilization time just after reset clear.

When a high level is input to the $\overline{\text{RESET}}$ input, the reset is cleared and program execution starts after the lapse of oscillation stabilization time ($2^{17}/f_x$). The reset applied by watchdog timer overflow is automatically cleared after a reset and program execution starts after the lapse of oscillation stabilization time ($2^{17}/f_x$) (see **Figures 25-2 to 25-4**).

- Cautions**
1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
 2. During reset input, main system clock oscillation remains stopped but subsystem clock oscillation continues.
 3. When the STOP mode is released by reset, the STOP mode contents are held during reset input. However, the port pin becomes high-impedance.

Figure 25-1. Block Diagram of Reset Function

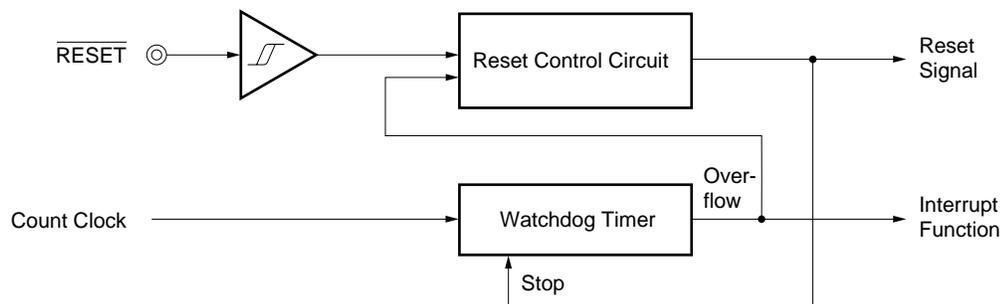


Figure 25-2. Timing of Reset by $\overline{\text{RESET}}$ Input

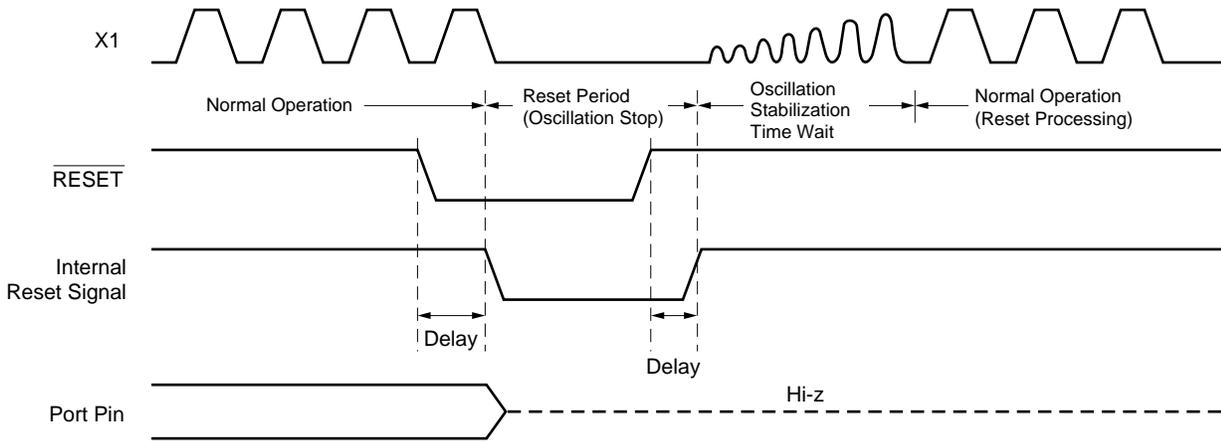


Figure 25-3. Timing of Reset due to Watchdog Timer Overflow

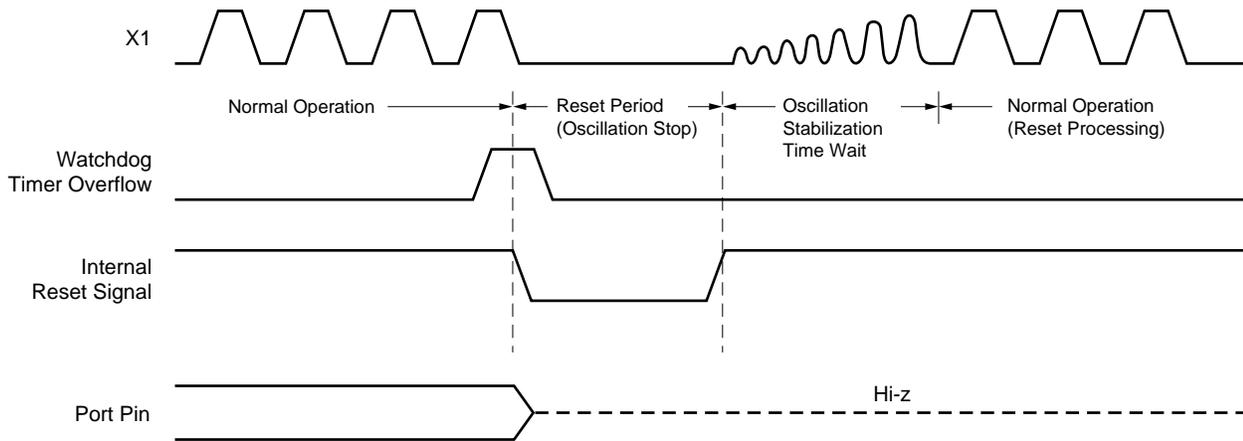


Figure 25-4. Timing of Reset by $\overline{\text{RESET}}$ Input in STOP Mode

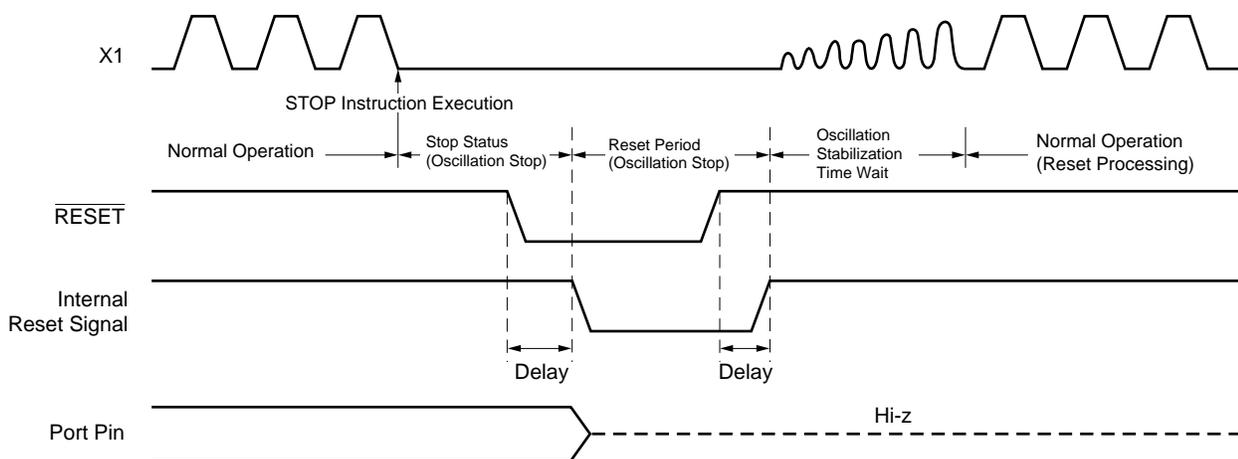


Table 25-1. Hardware Status after Reset (1/3)

Hardware		Status after Reset
Program counter (PC) ^{Note 1}		The contents of reset vector tables (0000H and 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined ^{Note 2}
	General register	Undefined ^{Note 2}
Port (Output latch)	Ports 0 to 3, Port 7, 9, 10, 12, 13 (P0 to P3, P7, P9, P10, P12, P13)	00H
	Port 4 to Port 6, Port 8 (P4 to P6, P8)	Undefined
Port mode register (PM0 to PM3, PM5 to PM10, PM12, PM13)		FFH
Pull-up resistor option register (PUOH, PUOL)		00H
Processor clock control register (PCC)		04H
Oscillation mode selection register (OSMS)		00H
Memory size switching register (IMS)		^{Note3}
Internal expansion RAM size switching register (IXS)		0AH
External bus type selection register (EBTS)		00H
Memory expansion mode register (MM)		10H
Oscillation stabilization time select register (OSTS)		04H
16-bit timer/event counter	Timer register (TM0)	00H
	Capture/compare register (CR00, CR01)	Undefined
	Clock selection register (TCL0)	00H
	Mode control register (TMC0)	00H
	Capture/compare control register 0 (CRC0)	04H
	Output control register (TOC0)	00H
8-bit timer/event counter 1 and 2	Timer register (TM1, TM2)	00H
	Compare registers (CR10, CR20)	Undefined
	Clock select register (TCL1)	00H
	Mode control registers (TMC1)	00H
	Output control register (TOC1)	00H

Notes 1. During reset input or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remains unchanged after reset.

2. In the standby mode, the status before the reset is held even after the reset.

3. The values after reset depend on the product.

μ PD78076, 78076Y : CCH μ PD78078, 78078Y : CFH

μ PD78P078, 78P078Y : CFH

Table 25-1. Hardware Status after Reset (2/3)

	Hardware	Status after Reset
8-bit timer/event counters 5 and 6	Timer register (TM5, TM6)	00H
	Compare register (CR50, CR60)	00H
	Clock select register (TCL5, TCL6)	00H
	Mode control register (TMC5, TMC6)	00H
Watch timer	Mode control register (TMC2)	00H
	Clock select register (TCL2)	00H
Watchdog timer	Mode register (WDTM)	00H
Serial interface	Clock select register (TCL3)	88H
	Shift registers (SIO0, SIO1)	Undefined
	Mode registers (CSIM0, CSIM1, CSIM2)	00H
	Serial bus interface control register (SBIC)	00H
	Slave address register (SVA)	Undefined
	Automatic data transmit/receive control register (ADTC)	00H
	Automatic data transmit/receive address pointer (ADTP)	00H
	Automatic data transmit/receive interval specify register (ADTI)	00H
	Asynchronous serial interface mode register (ASIM)	00H
	Asynchronous serial interface status register (ASIS)	00H
	Baud rate generator control register (BRGC)	00H
	Transmit shift register (TXS)	FFH
	Receive buffer register (RXB)	
	Interrupt timing specify register (SINT)	00H
A/D converter	Mode register (ADM)	01H
	Conversion result register (ADCR)	Undefined
	Input select register (ADIS)	00H
D/A converter	Mode register (DAM)	00H
	Conversion value setting register (DACS0, DACS1)	00H
Real-time output port	Mode register (RTPM)	00H
	Control register (RTPC)	00H
	Buffer register (RTBL, RTBH)	00H
ROM correction	Correction address register (CORAD0, CORAD1)	00H
	Correction control register (CORCN)	00H

Table 25-1. Hardware Status after Reset (3/3)

	Hardware	Status after Reset
Interrupt	Request flag register (IF0L, IF0H, IF1L)	00H
	Mask flag register (MK0L, MK0H, MK1L)	FFH
	Priority specify flag register (PR0L, PR0H, PR1L)	FFH
	External interrupt mode register (INTM0, INTM1)	00H
	Key return mode register (KRM)	02H
	Sampling clock select register (SCS)	00H

[MEMO]

CHAPTER 26 ROM CORRECTION

26.1 ROM Correction Functions

The μ PD78078, 78078Y Subseries can replace part of a program in the mask ROM with a program in the internal expansion RAM.

Instruction bugs found in the mask ROM can be avoided, and program flow can be changed by using the ROM correction.

The ROM correction can correct two places (max.) of the internal ROM (program).

Caution The ROM correction cannot be emulated by the in-circuit emulator (IE-78000-R, IE-78000-R-A IE-78001-R-A, IE-78K0-NS).

26.2 ROM Correction Configuration

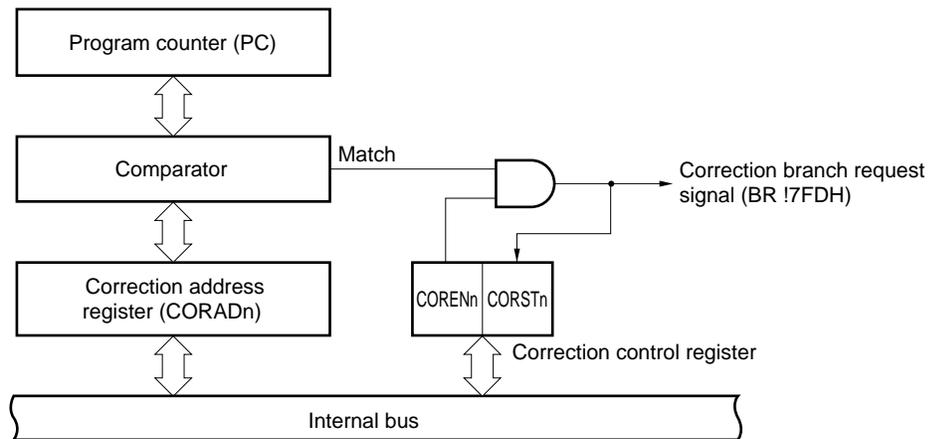
The ROM correction is executed by the following hardware.

Table 26-1. ROM Correction Configuration

Item	Configuration
Register	Correction address registers 0 and 1 (CORAD0, CORAD1)
Control register	Correction control register (CORCN)

Figure 26-1 shows a block diagram of the ROM correction.

Figure 26-1. Block Diagram of ROM Correction



Remark n = 0, 1

(1) Correction address registers 0 and 1 (CORAD0, CORAD1)

These registers set the start address (correction address) of the instruction(s) to be corrected in the mask ROM.

The ROM correction corrects two places (max.) of the program. Addresses are set to two registers, CORAD0 and CORAD1. If only one place needs to be corrected, set the address to either of the registers.

CORAD0 and CORAD1 are set with a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CORAD0 and CORAD1 to 0000H.

Figure 26-2. Correction Address Registers 0 and 1 Format

Symbol	15	0	Address	After reset	R/W
CORAD0			FF38H/FF39H	0000H	R/W
CORAD1			FF3AH/FF3BH	0000H	R/W

Cautions 1. Set the CORAD0 and CORAD1 when bit 1 (COREN0) and bit 3 (COREN1) of the correction control register (CORCN : see Figure 26-3) are 0.

2. Only addresses where operation codes are stored can be set in CORAD0 and CORAD1.

3. Do not set the following addresses to CORAD0 and CORAD1.

- Address value in table area of table reference instruction (CALLT instruction) : 0040H to 007FH
- Address value in vector table area : 0000H to 003FH

(2) Comparator

The comparator always compares the correction address value set in correction address registers 0 and 1 (CORAD0, CORAD1) with the fetch address value. When bit 1 (COREN0) or bit 3 (COREN1) of the correction control register (CORCN) is 1 and the correction address matches the fetch address value, the correction branch request signal (BR !F7FDH) is generated from the ROM correction circuit.

26.3 ROM Correction Control Registers

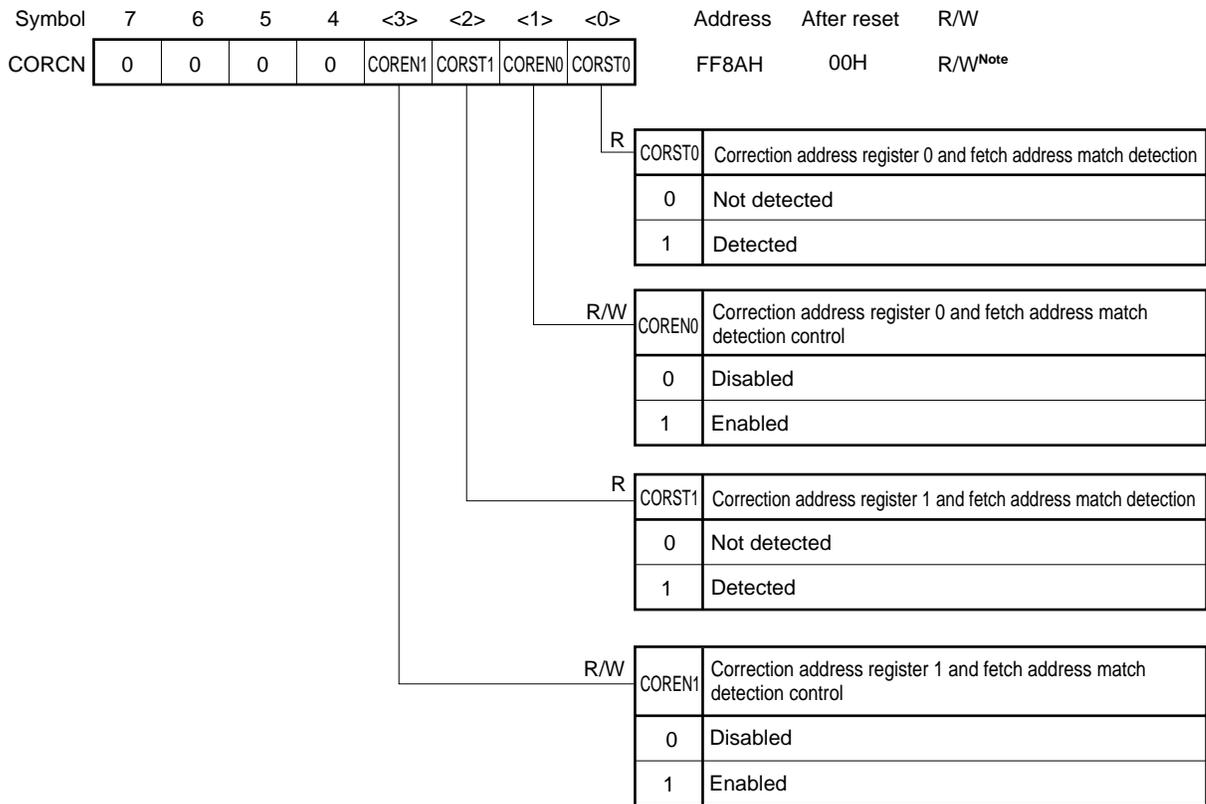
The ROM correction is controlled with the correction control register (CORCN).

(1) Correction control register (CORCN)

This register controls whether or not the correction branch request signal is generated when the fetch address matches the correction address set in correction address registers 0 and 1. The correction control register consists of correction enable flags (COREN0, COREN1) and correction status flags (CORST0, CORST1). The correction enable flags enable or disable the comparator match detection signal, and correction status flags show the values are matched. CORCN is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CORCN to 00H.

Figure 26-3. Correction Control Register Format



Note Bits 0 and 2 are read-only bits.

26.4 ROM Correction Application

- (1) Store the correction address and instruction after correction (patch program) to nonvolatile memory (such as EEPROM™) outside the microcontroller.

When two places should be corrected, store the branch destination judgment program as well. The branch destination judgment program checks which one of the addresses set to CORAD0 or CORAD1 generates the correction branch.

Figure 26-4. Storing Example to EEPROM (when One Place is Corrected)

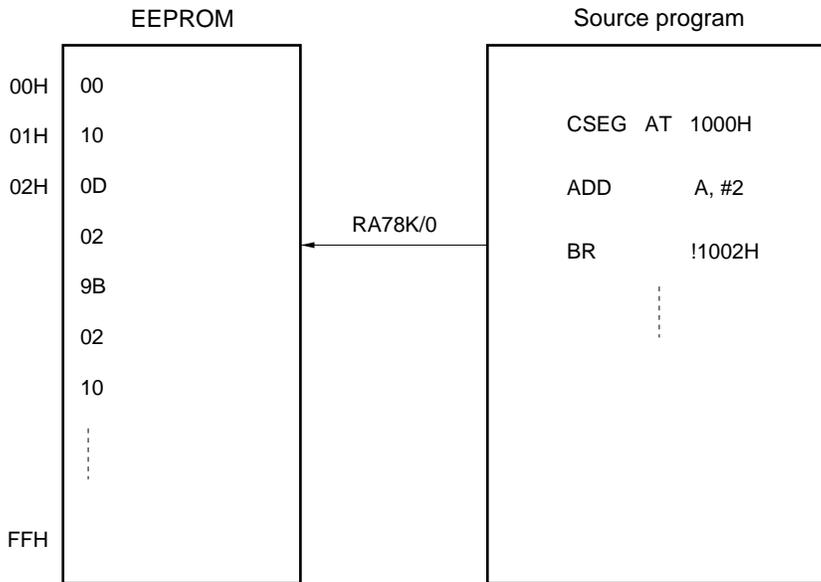
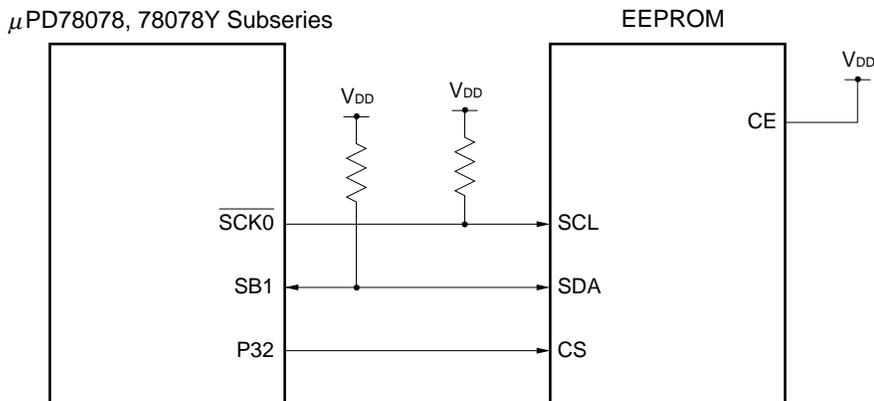
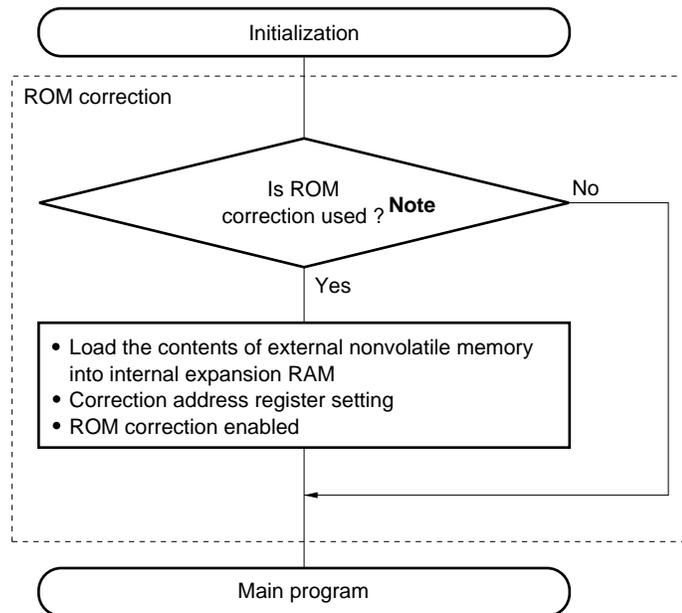


Figure 26-5. Connecting Example with EEPROM (Using 2-Wire Serial I/O Mode)



- (2) Assemble in advance the initialization routine as shown in Figure 26-6 to correct the program.

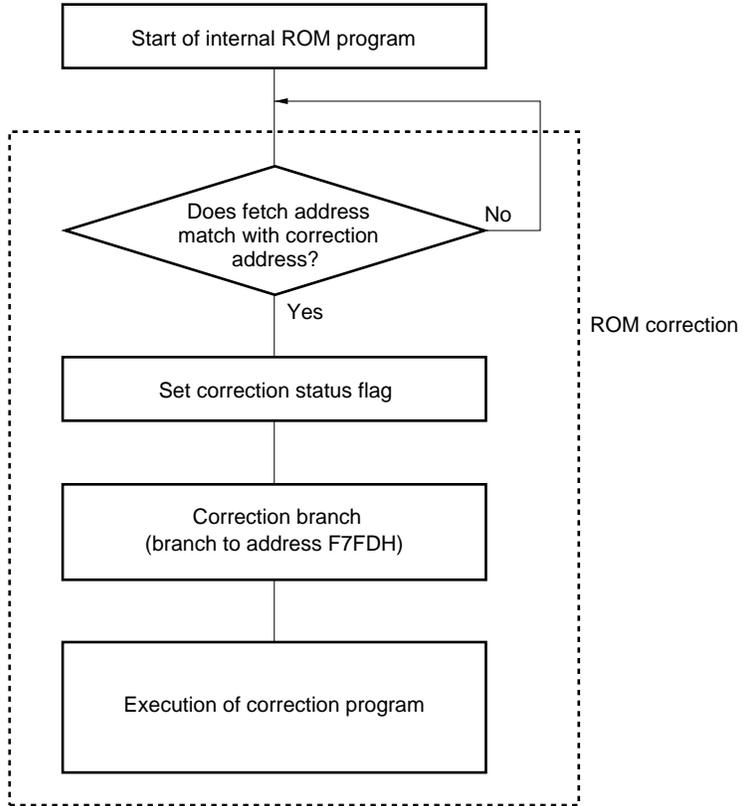
Figure 26-6. Initialization Routine



Note Whether the ROM correction is used or not should be judged by the port input level. For example, when the P20 input level is high, the ROM correction is used, otherwise, it is not used.

- (3) After reset, store the contents that have been previously stored in the external nonvolatile memory with initialization routine for ROM correction of the user to internal expansion RAM (see **Figure 26-6**). Set the start address of the instruction to be corrected to CORAD0 and CORAD1, and set bits 1 and 3 (COREN0, COREN1) of the correction control register (CORCN) to 1.
- (4) Set the entire-space branch instruction (BR !addr16) to the specified address (F7FDH) of the internal expansion RAM with the main program.
- (5) After the main program is started, the fetch address value and the values set in CORAD0 and CORAD1 are always compared by the comparator in the ROM correction circuit. When these values match, the correction branch request signal is generated. Simultaneously the corresponding correction status flag (CORST0 or CORST1) is set to 1.
- (6) Branch to the address F7FDH by the correction branch request signal.
- (7) Branch to the internal expansion RAM address set with the main program by the entire-space branch instruction of the address F7FDH.
- (8) When one place is corrected, the correction program is executed. When two places are corrected, the correction status flag is checked with the branch destination judgment program, and branches to the correction program.

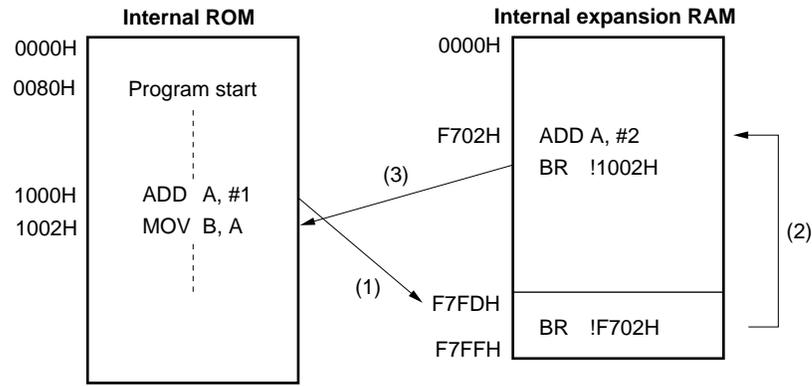
Figure 26-7. ROM Correction Operation



26.5 ROM Correction Example

The example of ROM correction when the instruction at address 1000H “ADD A, #1” is changed to “ADD A, #2” is as follows.

Figure 26-8. ROM Correction Example

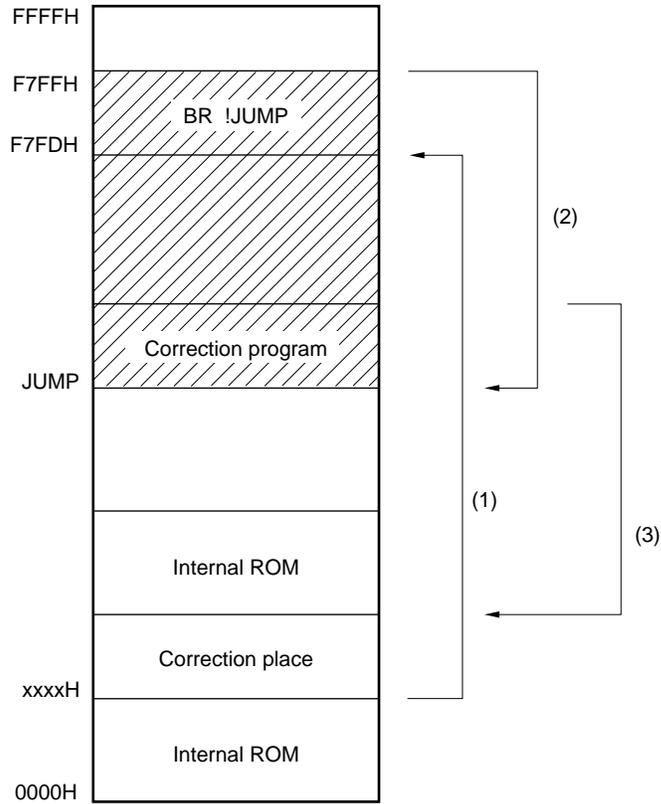


- (1) Branches to address F7FDH when the preset value 1000H in the correction address register matches the fetch address value after the main program is started.
- (2) Branches to any address (address F702H in this example) by setting the entire-space branch instruction (BR !addr16) to address F7FDH with the main program.
- (3) Returns to the internal ROM program after executing the substitute instruction ADD A, #2.

26.6 Program Execution Flow

Figures 26-9 and 26-10 show the program transition diagrams when the ROM correction is used.

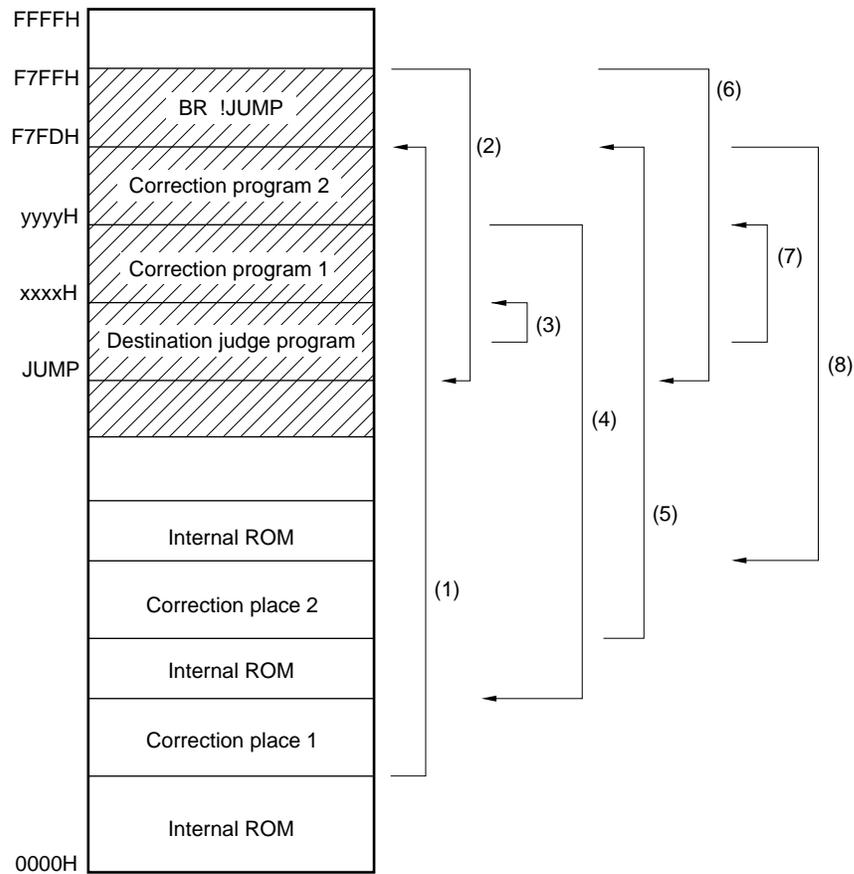
Figure 26-9. Program Transition Diagram (when One Place is Corrected)



- (1) Branches to address F7FDH when fetch address matches correction address
- (2) Branches to correction program
- (3) Returns to internal ROM program

Remark Area filled with diagonal lines : Internal expansion RAM
 JUMP : Correction program start address

Figure 26-10. Program Transition Diagram (when Two Places are Corrected)



- (1) Branches to address F7FDH when fetch address matches correction address
- (2) Branches to branch destination judgment program
- (3) Branches to correction program 1 by branch destination judgment program (BTCLR !CORST0, \$xxxxH)
- (4) Returns to internal ROM program
- (5) Branches to address F7FDH when fetch address matches correction address
- (6) Branches to branch destination judgment program
- (7) Branches to correction program 2 by branch destination judgment program (BTCLR !CORST1, \$yyyyH)
- (8) Returns to internal ROM program

Remark Area filled with diagonal lines : Internal expansion RAM
 JUMP : Correction program start address

26.7 Cautions on ROM Correction

- (1) Address values set in correction address registers 0 and 1 (CORAD0, CORAD1) must be addresses where instruction codes are stored.
- (2) Correction address registers 0 and 1 (CORAD0, CORAD1) should be set when the correction enable flag (COREN0, COREN1) is 0 (when the correction branch is in disabled state). If address is set to CORAD0 or CORAD1 when COREN0 or COREN1 is 1 (when the correction branch is in enabled state), the correction branch may start with the different address from the set address value.
- (3) Do not set the address value of instruction immediately after the instruction that sets the correction enable flag (COREN0, COREN1) to 1, to correction address register 0 or 1 (CORAD0, CORAD1) ; the correction branch may not start.
- (4) Do not set the address value in table area of table reference instruction (CALLT instruction) (0040H to 007FH), and the address value in vector table area (0000H to 003FH) to correction address registers 0 and 1 (CORAD0, CORAD1).
- (5) Do not set two addresses immediately after the instructions shown below to correction address registers 0 and 1 (CORAD0, CORAD1). (that is, when the mapped terminal address of these instructions is N, do not set the address values of N+1 and N+2.)
 - RET
 - RETI
 - RETB
 - BR \$addr16
 - STOP
 - HALT

CHAPTER 27 μ PD78P078, 78P078Y

The μ PD78P078 and 78P078Y (PROM versions) replace the internal mask ROM of the mask ROM versions (μ PD78074, 78075, 78076, 78078, and μ PD78074Y, 78075Y, 78076Y, 78078Y) with one-time programmable ROM or EPROM, which enable program writing, erasure, and rewriting. Table 27-1 lists differences between the PROM versions and mask ROM versions.

Table 27-1. Differences between PROM and Mask ROM Versions

Item	μ PD78P078, 78P078Y	Mask ROM Versions
Internal ROM format	One-time PROM/EPROM	Mask ROM
Internal ROM capacity	60 Kbytes	μ PD78076, 78076Y: 48 Kbytes μ PD78078, 78078Y: 60 Kbytes
Changing internal ROM and internal high-speed RAM capacities by the internal memory size switching register (IMS)	Yes ^{Note 1}	No
Changing internal expansion RAM capacity by the internal expansion RAM size switching register (IXS)	Yes ^{Note 2}	No
IC pin	None	Available
V _{PP} pin	Available	None
P60 to P63 pins pull-up resistor internal mask option	None	Available
Electrical specifications	Refer to the data sheet of individual product	

Notes 1. Internal PROM capacity becomes 60 Kbytes, and internal high-speed RAM capacity becomes 1024 bytes by $\overline{\text{RESET}}$ input.

2. Internal expansion RAM capacity becomes 1024 bytes by $\overline{\text{RESET}}$ input.

★ **Caution** There are differences in noise immunity and noise radiation between the PROM and mask ROM versions. When pre-producing an application set with the PROM version and then mass-producing it with the mask ROM version, be sure to conduct sufficient evaluations for the consumer samples (not engineering samples) of the mask ROM version.

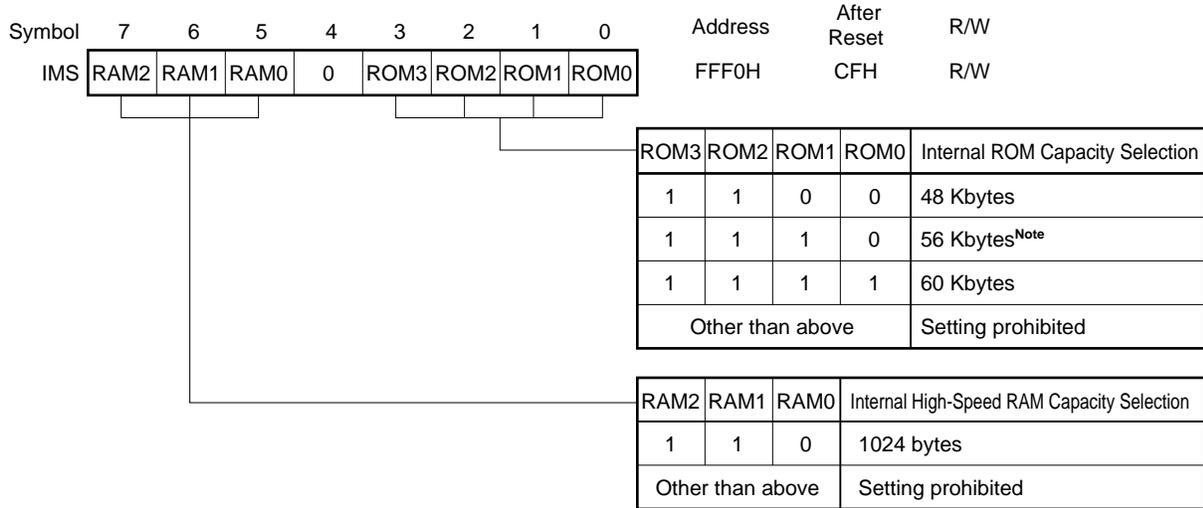
27.1 Internal Memory Size Switching Register

The μ PD78P078 and 78P078Y allow users to define its internal memory sizes using the internal memory size switching register (IMS), so that the same memory mapping as that of a mask ROM version with a different-size internal ROM is possible.

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

$\overline{\text{RESET}}$ input sets IMS to CFH.

Figure 27-1. Internal Memory Size Switching Register Format



Note When using the external device extension function on the following devices, the internal ROM capacity must be 56 Kbytes or less: μ PD78078, 78078Y, 78P078, 78P078Y.

Caution When using the mask ROM versions, do not set the values other than CCH and CFH to the IMS except for the case where the external device extension function is used on the μ PD78078 and 78078Y.

The IMS settings to give the same memory map as mask ROM versions are shown in Table 27-2.

Table 27-2. Examples of Internal Memory Size Switching Register Settings

Relevant Mask ROM Version	IMS Setting
μ PD78076, 78076Y	CCH
μ PD78078, 78078Y	CFH

27.2 Internal Extension RAM Size Switching Register

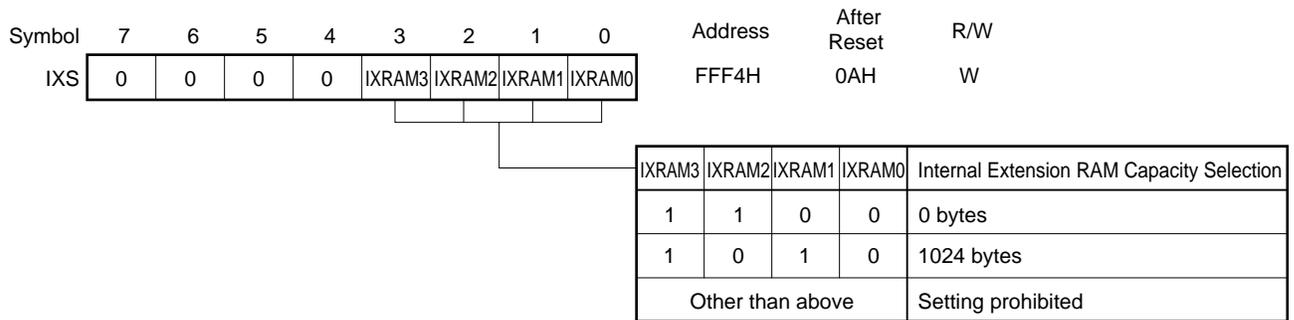
The μ PD78P078 and 78P078Y allow users to define its internal extension RAM size by using the internal extension RAM size switching register (IXS), so that the same memory mapping as that of a mask ROM version with a different internal extension RAM is possible.

The IXS is set by an 8-bit memory manipulation instruction.

RESET signal input sets IXS to 0AH.

Caution When the μ PD78076, 78078, 78076Y, or 78078Y is used, be sure to set the value specified in Table 27-3 to IXS. Other settings are prohibited.

Figure 27-2. Internal Extension RAM Size Switching Register Format



The value which is set in the IXS that has the identical memory map to the mask ROM versions is given in Table 27-3.

Table 27-3. Examples of Internal Extension RAM Size Switching Register Settings

Relevant Mask ROM Version	IXS Setting
μ PD78076, 78076Y	0AH
μ PD78078, 78078Y	

27.3 PROM Programming

The μ PD78P078 and 78P078Y each incorporate a 60-Kbyte PROM as program memory. To write a program into the PROM make the device enter the PROM programming mode by setting the levels of the V_{PP} and $\overline{\text{RESET}}$ pins as specified. For the connection of unused pins, refer to **1.5 (2) PROM programming mode** and **2.5 Pin Configuration (Top View)**.

Caution Write the program in the range of addresses 0000H to EFFFH (specify the last address as EFFFH). The program cannot be written by a PROM programmer in which the write address cannot be specified.

27.3.1 Operating modes

When +5 V or +12.5 V is applied to the V_{PP} pin and a low-level signal is applied to the $\overline{\text{RESET}}$ pin, the device is set to the PROM programming mode. This is one of the operating modes shown in Table 27-4 below according to the setting of the $\overline{\text{CE}}$, $\overline{\text{OE}}$, and $\overline{\text{PGM}}$ pins.

The PROM contents can be read by setting the read mode.

Table 27-4. PROM Programming Operating Modes

Operating mode	Pin	$\overline{\text{RESET}}$	V_{PP}	V_{DD}	$\overline{\text{CE}}$	$\overline{\text{OE}}$	$\overline{\text{PGM}}$	D0 to D7
	Page data latch	L		+12.5 V	+6.5 V	H	L	H
Page write	H					H	L	High impedance
Byte write	L					H	L	Data input
Program verify	L					L	H	Data output
Program inhibit	x					H	H	High impedance
	x			L	L			
Read	+5 V			+5 V	L	L	H	Data output
Output disabled					L	H	x	High impedance
Standby					H	x	x	High impedance

x: L or H

(1) Read mode

Read mode is set by setting $\overline{\text{CE}}$ to L and $\overline{\text{OE}}$ to L.

(2) Output disable mode

If $\overline{\text{OE}}$ is set to H, data output becomes high impedance and the output disable mode is set.

Therefore, if multiple μ PD78P078s or 78P078Ys are connected to the data bus, data can be read from any one device by controlling the $\overline{\text{OE}}$ pin.

(3) Standby mode

Setting \overline{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 ($X \leq 10$).

(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 ($X \leq 10$).

(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 μ PD78P078s or 78P078Ys are connected in parallel, and you wish to write to one of these devices.

The page write mode or byte write mode described above is used to perform a write. At this time, the write is not performed on the device which has the \overline{PGM} pin driven high.

27.3.2 PROM write procedure

Figure 27-3. Page Program Mode Flowchart

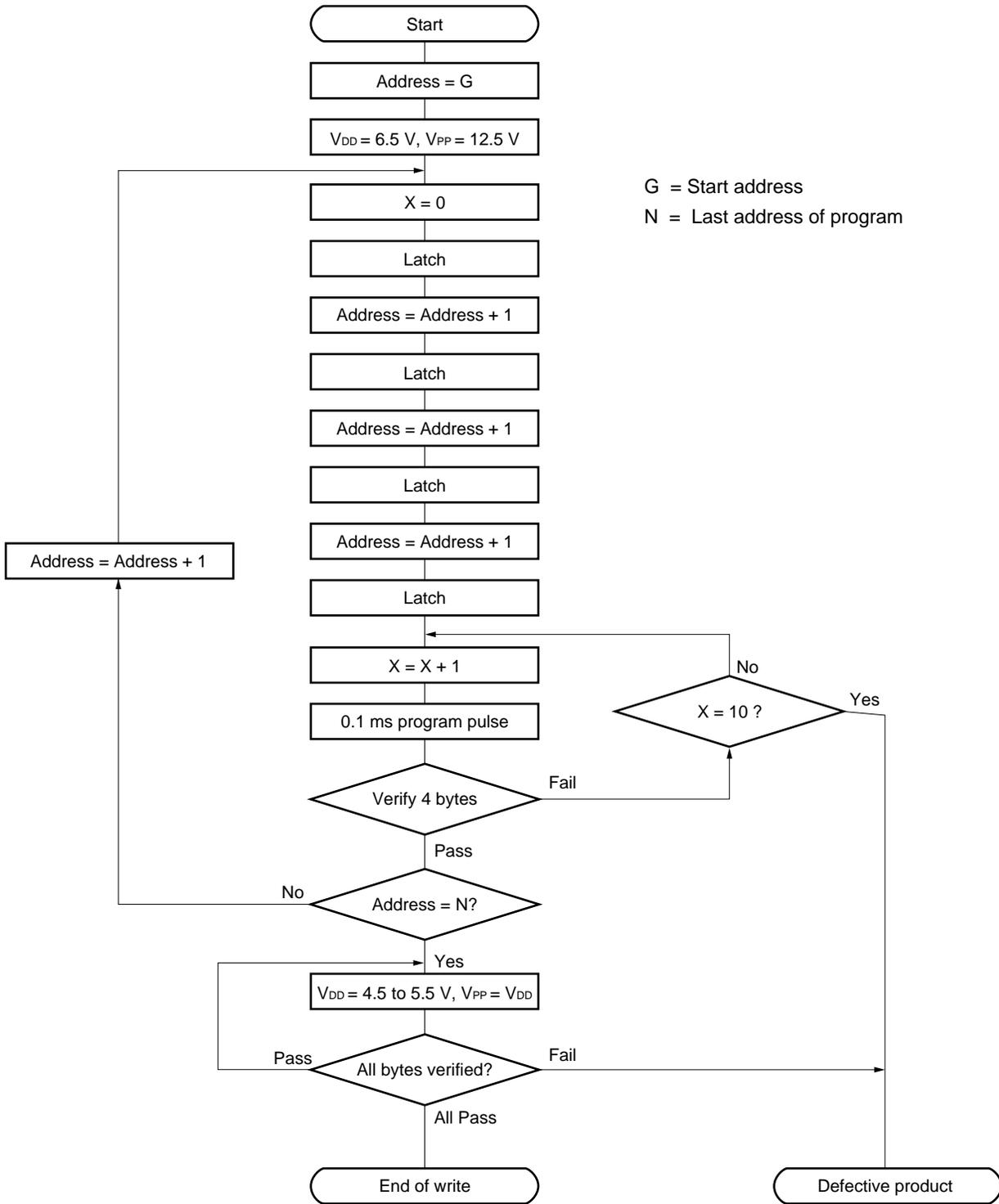


Figure 27-4. Page Program Mode Timing

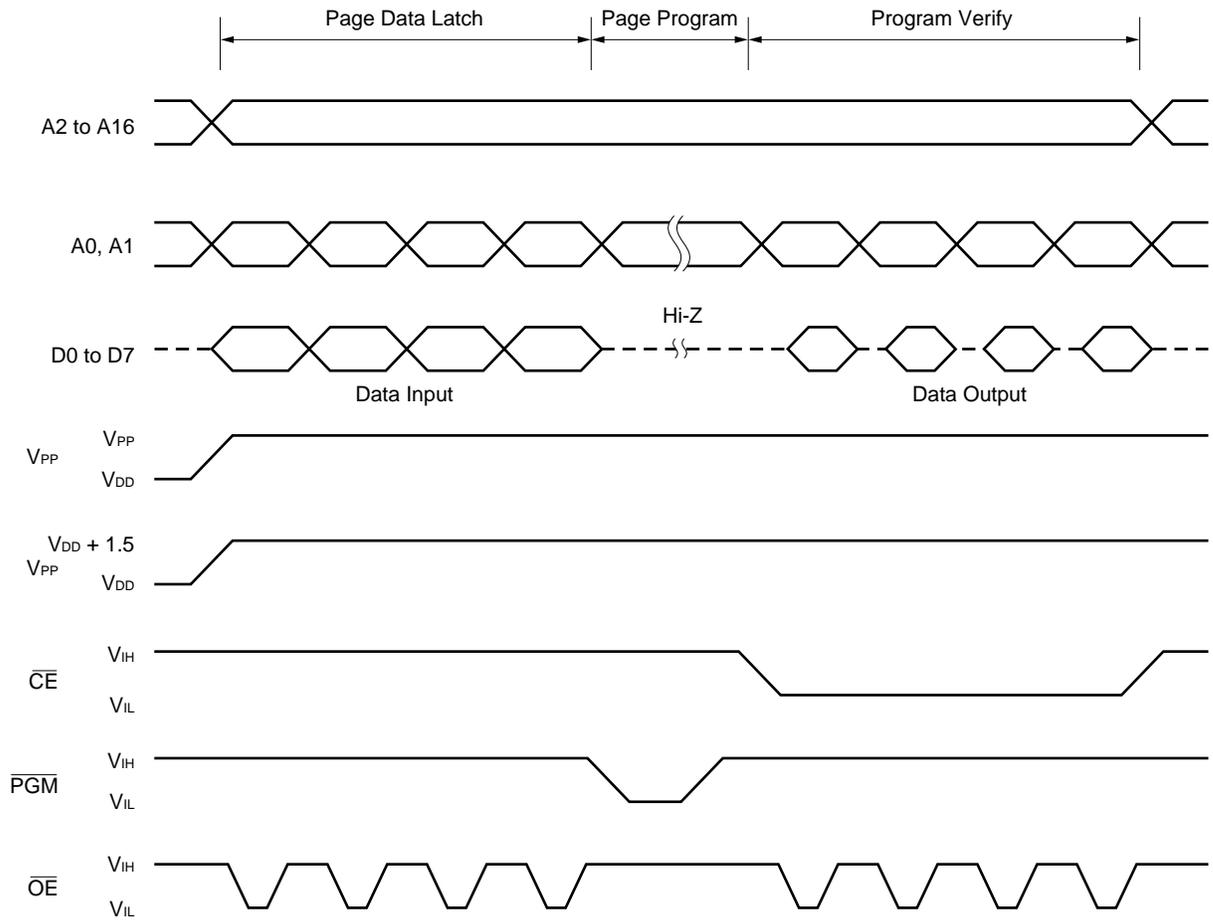


Figure 27-5. Byte Program Mode Flowchart

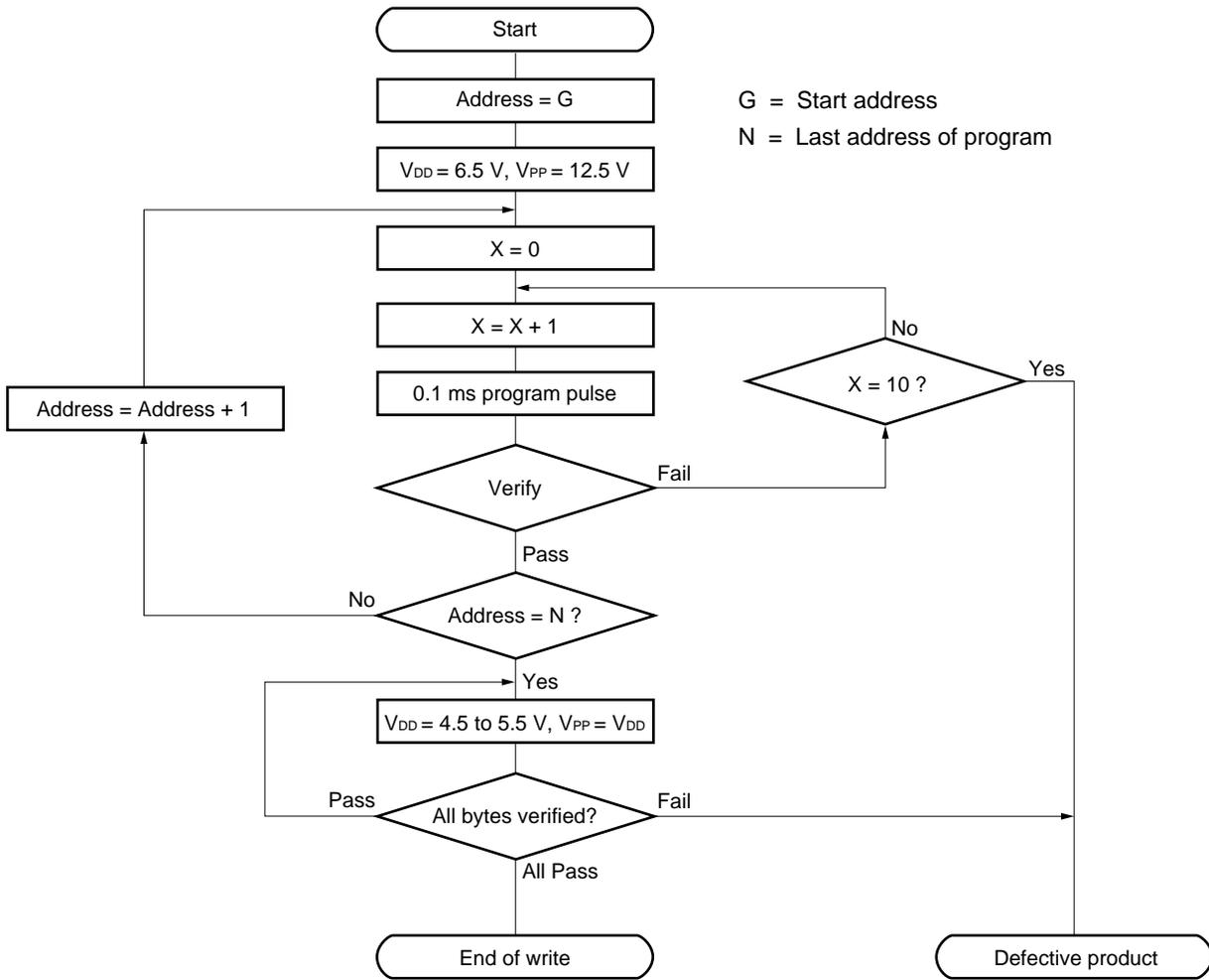
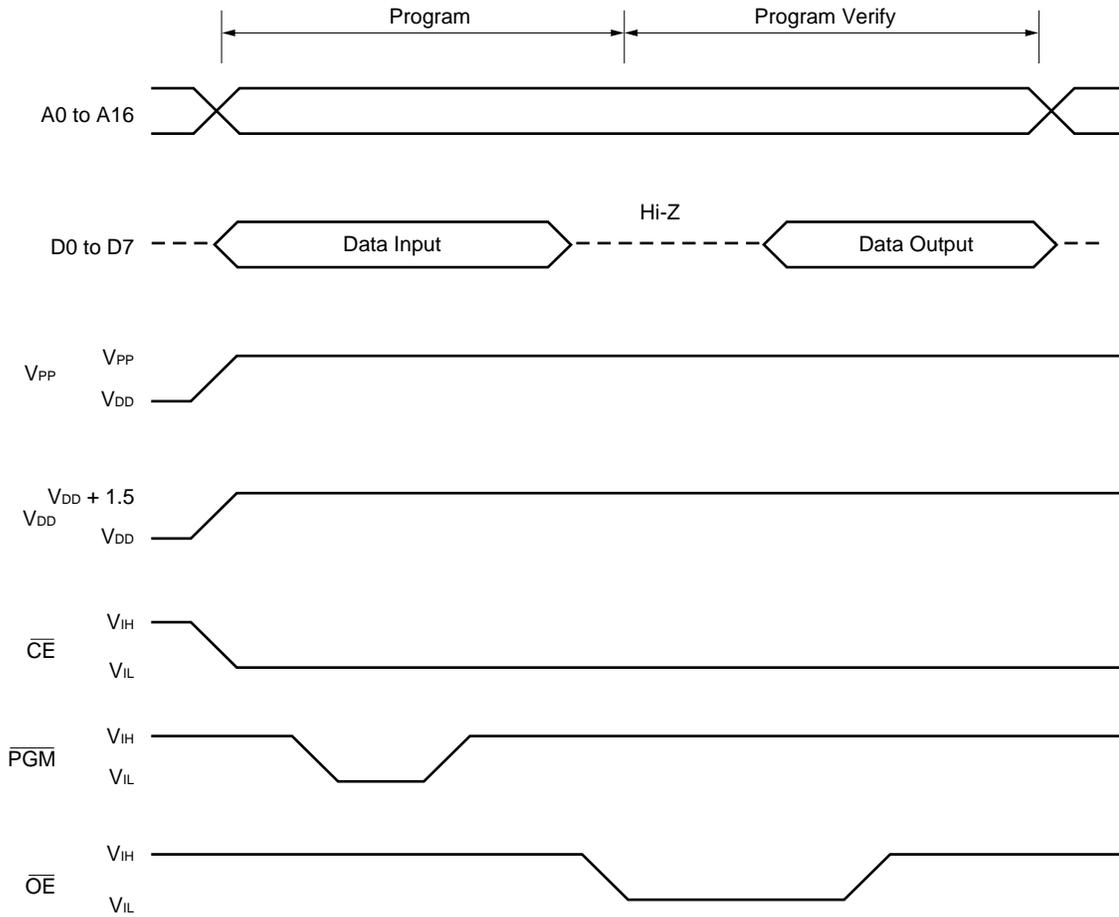


Figure 27-6. Byte Program Mode Timing



- Cautions**
1. 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 V_{PP} may have an adverse affect on reliability.

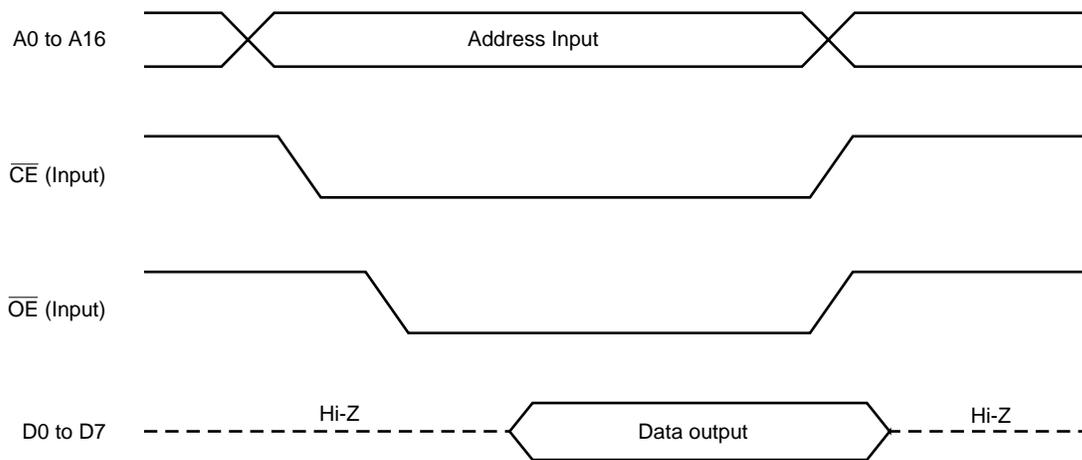
27.3.3 PROM reading procedure

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** and **2.5 Pin Configuration (Top View)**.
- (2) Supply +5 V to the V_{DD} and V_{PP} pins.
- (3) Input address of data to be read to pins A0 through A16.
- (4) Read mode.
- (5) Output data to pins D0 through D7.

The timing for steps (2) through (5) above is shown in Figure 27-7.

Figure 27-7. PROM Read Timing



27.4 Erasure Procedure (μ PD78P078KL-T and 78P078YKL-T Only)

With the μ PD78P078KL-T or 78P078YKL-T, it is possible to erase (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 x erasure time: 30 W·s/cm² or more
- Erasure time: 40 minutes (using a 12 mW/cm² ultraviolet lamp. A longer erasure 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.

27.5 Opaque Film Masking Window (μ PD78P078KL-T and 78P078YKL-T Only)

To prevent unintentional erasure of the EPROM contents by light and to prevent internal circuits from malfunction due to light coming in through the erasure window, mask the window with the attached opaque film after writing the EPROM.

27.6 Screening of One-Time PROM Versions

One-time PROM versions cannot be fully tested by NEC before shipment due to the structure of one-time PROM. Therefore, after users have written data to the PROM, screening should be implemented: that is, store devices under the following conditions. Then verify their contents after the device have returned to room temperature.

Storage Temperature	Storage Time
125°C	24 hours

For users who do not wish to implement screening by themselves, NEC provides such users with a charged service in which NEC performs a series of processes from writing one-time PROMs and screening them to verifying their contents for users by request. The PROM version devices which provide this service are called QTOP™ microcontrollers. This service will soon be available for the μ PD78P078Y. For details, please consult an NEC sales representative.

[MEMO]

CHAPTER 28 INSTRUCTION SET

This chapter describes each instruction set of the μ PD78078 and 78078Y Subseries as list table. For details of its operation and operation code, refer to the separate document “78K/0 Series USER’S MANUAL — Instructions (U12326E).”

28.1 Legends Used in Operation List

28.1.1 Operand identifiers and description methods

Operands are described in “Operand” column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for detail). When there are two or more description methods, select one of them. Alphabetic letters in capitals and symbols, #, !, \$ and [] are key words and must be described as they are. Each symbol has the following meaning.

- # : Immediate data specification
- ! : Absolute address specification
- \$: Relative address specification
- [] : Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, \$, and [] symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for description.

Table 28-1. Operand Identifiers and Description Methods

Identifier	Description Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7),
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special-function register symbol ^{Note}
sfrp	Special-function register symbol (16-bit manipulatable register even addresses only) ^{Note}
saddr	FE20H to FF1FH Immediate data or labels
saddrp	FE20H to FF1FH Immediate data or labels (even address only)
addr16	0000H to FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions)
addr11	0800H to 0FFFH Immediate data or labels
addr5	0040H to 007FH Immediate data or labels (even address only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special-function register symbols, refer to **Table 5-3 Special-Function Register List**.

28.1.2 Description of “operation” column

A	: A register; 8-bit accumulator
X	: X register
B	: B register
C	: C register
D	: D register
E	: E register
H	: H register
L	: L register
AX	: AX register pair; 16-bit accumulator
BC	: BC register pair
DE	: DE register pair
HL	: HL register pair
PC	: Program counter
SP	: Stack pointer
PSW	: Program status word
CY	: Carry flag
AC	: Auxiliary carry flag
Z	: Zero flag
RBS	: Register bank select flag
IE	: Interrupt request enable flag
NMIS	: Non-maskable interrupt servicing flag
()	: Memory contents indicated by address or register contents in parentheses
x _H , x _L	: Higher 8 bits and lower 8 bits of 16-bit register
∧	: Logical product (AND)
∨	: Logical sum (OR)
⊕	: Exclusive logical sum (exclusive OR)
—	: Inverted data
addr16	: 16-bit immediate data or label
jdisp8	: Signed 8-bit data (displacement value)

23.1.3 Description of “flag operation” column

(Blank)	: Not affected
0	: Cleared to 0
1	: Set to 1
x	: Set/cleared according to the result
R	: Previously saved value is restored

28.2 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 + n	A ← (addr16)				
		!addr16, A	3	8	9 + m	(addr16) ← A				
		PSW, #byte	3	—	7	PSW ← byte	x	x	x	
		A, PSW	2	—	5	A ← PSW				
		PSW, A	2	—	5	PSW ← A	x	x	x	
		A, [DE]	1	4	5 + n	A ← (DE)				
		[DE], A	1	4	5 + m	(DE) ← A				
		A, [HL]	1	4	5 + n	A ← (HL)				
		[HL], A	1	4	5 + m	(HL) ← A				
		A, [HL + byte]	2	8	9 + n	A ← (HL + byte)				
		[HL + byte], A	2	8	9 + m	(HL + byte) ← A				
	A, [HL + B]	1	6	7 + n	A ← (HL + B)					
	[HL + B], A	1	6	7 + m	(HL + B) ← A					
	A, [HL + C]	1	6	7 + n	A ← (HL + C)					
	[HL + C], A	1	6	7 + m	(HL + C) ← A					
	XCH	A, r ^{Note 3}	A, saddr	2	4	6	A ↔ (saddr)			
			A, sfr	2	—	6	A ↔ sfr			
			A, !addr16	3	8	10+n+m	A ↔ (addr16)			
			A, [DE]	1	4	6+n+m	A ↔ (DE)			
A, [HL]			1	4	6+n+m	A ↔ (HL)				
A, [HL + byte]			2	8	10+n+m	A ↔ (HL + byte)				
A, [HL + B]			2	8	10+n+m	A ↔ (HL + B)				
A, [HL + C]			2	8	10+n+m	A ↔ (HL + C)				

- Notes**
1. For instructions that access the internal high-speed RAM area or perform no data access
 2. For instructions that access an area other than the internal high-speed RAM area
 3. Except when “r = A”

- Remarks**
1. One clock in the “Clock” columns is equal to one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).
 2. The values in the “Clock” column assumes that the internal ROM area contains programs.
 3. n indicates wait cycles to be inserted when an external expansion memory area is read from.
 4. m indicates wait cycles to be inserted when an external expansion memory area is written to.

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 + 2n	AX ← (addr16)			
	!addr16, AX	3	10	12 + 2m	(addr16) ← AX				
XCHW	AX, rp ^{Note 3}	1	4	—	AX ↔ rp				
8-bit operation	ADD	A, #byte	2	4	—	A, CY ← A + byte	x	x	x
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte	x	x	x
		A, r ^{Note 4}	2	4	—	A, CY ← A + r	x	x	x
		r, A	2	4	—	r, CY ← r + A	x	x	x
		A, saddr	2	4	5	A, CY ← A + (saddr)	x	x	x
		A, !addr16	3	8	9 + n	A, CY ← A + (addr16)	x	x	x
		A, [HL]	1	4	5 + n	A, CY ← A + (HL)	x	x	x
		A, [HL + byte]	2	8	9 + n	A, CY ← A + (HL + byte)	x	x	x
		A, [HL + B]	2	8	9 + n	A, CY ← A + (HL + B)	x	x	x
		A, [HL + C]	2	8	9 + n	A, CY ← A + (HL + C)	x	x	x
	ADDC	A, #byte	2	4	—	A, CY ← A + byte + CY	x	x	x
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte + CY	x	x	x
		A, r ^{Note 4}	2	4	—	A, CY ← A + r + CY	x	x	x
		r, A	2	4	—	r, CY ← r + A + CY	x	x	x
		A, saddr	2	4	5	A, CY ← A + (saddr) + CY	x	x	x
		A, !addr16	3	8	9 + n	A, CY ← A + (addr16) + CY	x	x	x
		A, [HL]	1	4	5 + n	A, CY ← A + (HL) + CY	x	x	x
		A, [HL + byte]	2	8	9 + n	A, CY ← A + (HL + byte) + CY	x	x	x
		A, [HL + B]	2	8	9 + n	A, CY ← A + (HL + B) + CY	x	x	x
		A, [HL + C]	2	8	9 + n	A, CY ← A + (HL + C) + CY	x	x	x

- Notes**
1. For instructions that access the internal high-speed RAM area or perform no data access
 2. For instructions that access an area other than the internal high-speed RAM area
 3. Only when rp = BC, DE, or HL
 4. Except when “r = A”

- Remarks**
1. One clock in the “Clock” columns is equal to one cycle of the CPU clock (f_{cpu}) selected by the processor clock control register (PCC).
 2. The values in the “Clock” column assumes that the internal ROM area contains programs.
 3. n indicates wait cycles to be inserted when an external expansion memory area is read from.
 4. m indicates wait cycles to be inserted when an external expansion memory area is written to.

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	x	x	x
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte	x	x	x
		A, r ^{Note 3}	2	4	—	A, CY ← A – r	x	x	x
		r, A	2	4	—	r, CY ← r – A	x	x	x
		A, saddr	2	4	5	A, CY ← A – (saddr)	x	x	x
		A, !addr16	3	8	9 + n	A, CY ← A – (addr16)	x	x	x
		A, [HL]	1	4	5 + n	A, CY ← A – (HL)	x	x	x
		A, [HL + byte]	2	8	9 + n	A, CY ← A – (HL + byte)	x	x	x
		A, [HL + B]	2	8	9 + n	A, CY ← A – (HL + B)	x	x	x
		A, [HL + C]	2	8	9 + n	A, CY ← A – (HL + C)	x	x	x
	SUBC	A, #byte	2	4	—	A, CY ← A – byte – CY	x	x	x
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte – CY	x	x	x
		A, r ^{Note 3}	2	4	—	A, CY ← A – r – CY	x	x	x
		r, A	2	4	—	r, CY ← r – A – CY	x	x	x
		A, saddr	2	4	5	A, CY ← A – (saddr) – CY	x	x	x
		A, !addr16	3	8	9 + n	A, CY ← A – (addr16) – CY	x	x	x
		A, [HL]	1	4	5 + n	A, CY ← A – (HL) – CY	x	x	x
		A, [HL + byte]	2	8	9 + n	A, CY ← A – (HL + byte) – CY	x	x	x
		A, [HL + B]	2	8	9 + n	A, CY ← A – (HL + B) – CY	x	x	x
		A, [HL + C]	2	8	9 + n	A, CY ← A – (HL + C) – CY	x	x	x
	AND	A, #byte	2	4	—	A ← A ∧ byte	x		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∧ byte	x		
		A, r ^{Note 3}	2	4	—	A ← A ∧ r	x		
		r, A	2	4	—	r ← r ∧ A	x		
		A, saddr	2	4	5 + n	A ← A ∧ (saddr)	x		
		A, !addr16	3	8	9 + n	A ← A ∧ (addr16)	x		
		A, [HL]	1	4	5 + n	A ← A ∧ (HL)	x		
		A, [HL + byte]	2	8	9 + n	A ← A ∧ (HL + byte)	x		
		A, [HL + B]	2	8	9 + n	A ← A ∧ (HL + B)	x		
		A, [HL + C]	2	8	9 + n	A ← A ∧ (HL + C)	x		

- Notes**
1. For instructions that access the internal high-speed RAM area or perform no data access
 2. For instructions that access an area other than the internal high-speed RAM area
 3. Except when “r = A”

- Remarks**
1. One clock in the “Clock” columns is equal to one cycle of the CPU clock (f_{cpu}) selected by the processor clock control register (PCC).
 2. The values in the “Clock” column assumes that the internal ROM area contains programs.
 3. n indicates wait cycles to be inserted when an external expansion memory area is read from.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	OR	A, #byte	2	4	—	$A \leftarrow A \vee \text{byte}$	x		
		saddr, #byte	3	6	8	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$	x		
		A, r ^{Note 3}	2	4	—	$A \leftarrow A \vee r$	x		
		r, A	2	4	—	$r \leftarrow r \vee A$	x		
		A, saddr	2	4	5	$A \leftarrow A \vee (\text{saddr})$	x		
		A, !addr16	3	8	9 + n	$A \leftarrow A \vee (\text{addr16})$	x		
		A, [HL]	1	4	5 + n	$A \leftarrow A \vee (\text{HL})$	x		
		A, [HL + byte]	2	8	9 + n	$A \leftarrow A \vee (\text{HL} + \text{byte})$	x		
		A, [HL + B]	2	8	9 + n	$A \leftarrow A \vee (\text{HL} + B)$	x		
	A, [HL + C]	2	8	9 + n	$A \leftarrow A \vee (\text{HL} + C)$	x			
	XOR	A, #byte	2	4	—	$A \leftarrow A \vee \text{byte}$	x		
		saddr, #byte	3	6	8	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$	x		
		A, r ^{Note 3}	2	4	—	$A \leftarrow A \vee r$	x		
		r, A	2	4	—	$r \leftarrow r \vee A$	x		
		A, saddr	2	4	5 + n	$A \leftarrow A \vee (\text{saddr})$	x		
		A, !addr16	3	8	9 + n	$A \leftarrow A \vee (\text{addr16})$	x		
		A, [HL]	1	4	5 + n	$A \leftarrow A \vee (\text{HL})$	x		
		A, [HL + byte]	2	8	9 + n	$A \leftarrow A \vee (\text{HL} + \text{byte})$	x		
		A, [HL + B]	2	8	9 + n	$A \leftarrow A \vee (\text{HL} + B)$	x		
	A, [HL + C]	2	8	9 + n	$A \leftarrow A \vee (\text{HL} + C)$	x			
	CMP	A, #byte	2	4	—	$A - \text{byte}$	x	x	x
		saddr, #byte	3	6	8	$(\text{saddr}) - \text{byte}$	x	x	x
		A, r ^{Note 3}	2	4	—	$A - r$	x	x	x
		r, A	2	4	—	$r - A$	x	x	x
		A, saddr	2	4	5 + n	$A - (\text{saddr})$	x	x	x
		A, !addr16	3	8	9 + n	$A - (\text{addr16})$	x	x	x
		A, [HL]	1	4	5 + n	$A - (\text{HL})$	x	x	x
A, [HL + byte]		2	8	9 + n	$A - (\text{HL} + \text{byte})$	x	x	x	
A, [HL + B]		2	8	9 + n	$A - (\text{HL} + B)$	x	x	x	
A, [HL + C]	2	8	9 + n	$A - (\text{HL} + C)$	x	x	x		

- Notes**
1. For instructions that access the internal high-speed RAM area or perform no data access
 2. For instructions that access an area other than the internal high-speed RAM area
 3. Except when “r = A”

- Remarks**
1. One clock in the “Clock” columns is equal to one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).
 2. The values in the “Clock” column assumes that the internal ROM area contains programs.
 3. n indicates wait cycles to be inserted when an external expansion memory area is read from.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	ADDW	AX, #word	3	6	—	$AX, CY \leftarrow AX + \text{word}$	x	x	x
	SUBW	AX, #word	3	6	—	$AX, CY \leftarrow AX - \text{word}$	x	x	x
	CMPW	AX, #word	3	6	—	$AX - \text{word}$	x	x	x
Multiply divide	MULU	X	2	16	—	$AX \leftarrow A \times X$			
	DIVUW	C	2	25	—	$AX \text{ (Quotient), } C \text{ (Remainder)} \leftarrow AX/C$			
Increment decrement	INC	r	1	2	—	$r \leftarrow r + 1$	x	x	
		saddr	2	4	6	$(\text{saddr}) \leftarrow (\text{saddr}) + 1$	x	x	
	DEC	r	1	2	—	$r \leftarrow r - 1$	x	x	
		saddr	2	4	6	$(\text{saddr}) \leftarrow (\text{saddr}) - 1$	x	x	
	INCW	rp	1	4	—	$rp \leftarrow rp + 1$			
	DECW	rp	1	4	—	$rp \leftarrow rp - 1$			
Rotate	ROR	A, 1	1	2	—	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1 \text{ time}$			x
	ROL	A, 1	1	2	—	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1 \text{ time}$			x
	RORC	A, 1	1	2	—	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1 \text{ time}$			x
	ROLC	A, 1	1	2	—	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1 \text{ time}$			x
	ROR4	[HL]	2	10	12+n+m	$A_{3-0} \leftarrow (HL)_{3-0}, (HL)_{7-4} \leftarrow A_{3-0}, (HL)_{3-0} \leftarrow (HL)_{7-4}$			
	ROL4	[HL]	2	10	12+n+m	$A_{3-0} \leftarrow (HL)_{7-4}, (HL)_{3-0} \leftarrow A_{3-0}, (HL)_{7-4} \leftarrow (HL)_{3-0}$			
BCD adjust	ADJBA		2	4	—	Decimal Adjust Accumulator after Addition	x	x	x
	ADJBS		2	4	—	Decimal Adjust Accumulator after Subtract	x	x	x
Bit manipulation	MOV1	CY, saddr.bit	3	6	7	$CY \leftarrow (\text{saddr.bit})$			x
		CY, sfr.bit	3	—	7	$CY \leftarrow \text{sfr.bit}$			x
		CY, A.bit	2	4	—	$CY \leftarrow A.\text{bit}$			x
		CY, PSW.bit	3	—	7	$CY \leftarrow \text{PSW.bit}$			x
		CY, [HL].bit	2	6	7 + n	$CY \leftarrow (HL).\text{bit}$			x
		saddr.bit, CY	3	6	8	$(\text{saddr.bit}) \leftarrow CY$			
		sfr.bit, CY	3	—	8	$\text{sfr.bit} \leftarrow CY$			
		A.bit, CY	2	4	—	$A.\text{bit} \leftarrow CY$			
		PSW.bit, CY	3	—	8	$\text{PSW.bit} \leftarrow CY$			x
[HL].bit, CY	2	6	8+n+m	$(HL).\text{bit} \leftarrow CY$					

- Notes**
1. For instructions that access the internal high-speed RAM area or perform no data access
 2. For instructions that access an area other than the internal high-speed RAM area

- Remarks**
1. One clock in the “Clock” columns is equal to one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).
 2. The values in the “Clock” column assumes that the internal ROM area contains programs.
 3. n indicates wait cycles to be inserted when an external expansion memory area is read from.
 4. m indicates wait cycles to be inserted when an external expansion memory area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulation	AND1	CY, saddr.bit	3	6	7	$CY \leftarrow CY \wedge (\text{saddr.bit})$			x
		CY, sfr.bit	3	—	7	$CY \leftarrow CY \wedge \text{sfr.bit}$			x
		CY, A.bit	2	4	—	$CY \leftarrow CY \wedge A.\text{bit}$			x
		CY, PSW.bit	3	—	7	$CY \leftarrow CY \wedge \text{PSW.bit}$			x
		CY, [HL].bit	2	6	7 + n	$CY \leftarrow CY \wedge (\text{HL}).\text{bit}$			x
	OR1	CY, saddr.bit	3	6	7	$CY \leftarrow CY \vee (\text{saddr.bit})$			x
		CY, sfr.bit	3	—	7	$CY \leftarrow CY \vee \text{sfr.bit}$			x
		CY, A.bit	2	4	—	$CY \leftarrow CY \vee A.\text{bit}$			x
		CY, PSW.bit	3	—	7	$CY \leftarrow CY \vee \text{PSW.bit}$			x
		CY, [HL].bit	2	6	7 + n	$CY \leftarrow CY \vee (\text{HL}).\text{bit}$			x
	XOR1	CY, saddr.bit	3	6	7	$CY \leftarrow CY \vee (\text{saddr.bit})$			x
		CY, sfr.bit	3	—	7	$CY \leftarrow CY \vee \text{sfr.bit}$			x
		CY, A.bit	2	4	—	$CY \leftarrow CY \vee A.\text{bit}$			x
		CY, PSW. bit	3	—	7	$CY \leftarrow CY \vee \text{PSW.bit}$			x
		CY, [HL].bit	2	6	7 + n	$CY \leftarrow CY \vee (\text{HL}).\text{bit}$			x
	SET1	saddr.bit	2	4	6	$(\text{saddr.bit}) \leftarrow 1$			
		sfr.bit	3	—	8	$\text{sfr.bit} \leftarrow 1$			
		A.bit	2	4	—	$A.\text{bit} \leftarrow 1$			
		PSW.bit	2	—	6	$\text{PSW.bit} \leftarrow 1$	x	x	x
		[HL].bit	2	6	8+n+m	$(\text{HL}).\text{bit} \leftarrow 1$			
	CLR1	saddr.bit	2	4	6	$(\text{saddr.bit}) \leftarrow 0$			
		sfr.bit	3	—	8	$\text{sfr.bit} \leftarrow 0$			
		A.bit	2	4	—	$A.\text{bit} \leftarrow 0$			
		PSW.bit	2	—	6	$\text{PSW.bit} \leftarrow 0$	x	x	x
		[HL].bit	2	6	8+n+m	$(\text{HL}).\text{bit} \leftarrow 0$			
SET1	CY	1	2	—	$CY \leftarrow 1$			1	
CLR1	CY	1	2	—	$CY \leftarrow 0$			0	
NOT1	CY	1	2	—	$CY \leftarrow \overline{CY}$			x	

- Notes**
1. For instructions that access the internal high-speed RAM area or perform no data access
 2. For instructions that access an area other than the internal high-speed RAM area

- Remarks**
1. One clock in the “Clock” columns is equal to one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).
 2. The values in the “Clock” column assumes that the internal ROM area contains programs.
 3. n indicates wait cycles to be inserted when an external expansion memory area is read from.
 4. m indicates wait cycles to be inserted when an external expansion memory area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call / return	CALL	!addr16	3	7	—	$(SP - 1) \leftarrow (PC + 3)_H, (SP - 2) \leftarrow (PC + 3)_L,$ $PC \leftarrow \text{addr16}, SP \leftarrow SP - 2$			
	CALLF	!addr11	2	5	—	$(SP - 1) \leftarrow (PC + 2)_H, (SP - 2) \leftarrow (PC + 2)_L,$ $PC_{15-11} \leftarrow 00001, PC_{10-0} \leftarrow \text{addr11},$ $SP \leftarrow SP - 2$			
	CALLT	[addr5]	1	6	—	$(SP - 1) \leftarrow (PC + 1)_H, (SP - 2) \leftarrow (PC + 1)_L,$ $PC_H \leftarrow (00000000, \text{addr5} + 1),$ $PC_L \leftarrow (00000000, \text{addr5}),$ $SP \leftarrow SP - 2$			
	BRK		1	6	—	$(SP - 1) \leftarrow \text{PSW}, (SP - 2) \leftarrow (PC + 1)_H,$ $(SP - 3) \leftarrow (PC + 1)_L, PC_H \leftarrow (003FH),$ $PC_L \leftarrow (003EH), SP \leftarrow SP - 3, IE \leftarrow 0$			
	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	—	$(SP - 1) \leftarrow \text{PSW}, SP \leftarrow SP - 1$			
		rp	1	4	—	$(SP - 1) \leftarrow \text{rp}_H, (SP - 2) \leftarrow \text{rp}_L,$ $SP \leftarrow SP - 2$			
	POP	PSW	1	2	—	$\text{PSW} \leftarrow (SP), SP \leftarrow SP + 1$	R	R	R
		rp	1	4	—	$\text{rp}_H \leftarrow (SP + 1), \text{rp}_L \leftarrow (SP),$ $SP \leftarrow SP + 2$			
	MOVW	SP, #word	4	—	10	$SP \leftarrow \text{word}$			
		SP, AX	2	—	8	$SP \leftarrow \text{AX}$			
AX, SP		2	—	8	$\text{AX} \leftarrow \text{SP}$				
Unconditional branch	BR	!addr16	3	6	—	$PC \leftarrow \text{addr16}$			
		\$addr16	2	6	—	$PC \leftarrow PC + 2 + \text{jdisp8}$			
		AX	2	8	—	$PC_H \leftarrow A, PC_L \leftarrow X$			
Conditional branch	BC	\$addr16	2	6	—	$PC \leftarrow PC + 2 + \text{jdisp8}$ if CY = 1			
	BNC	\$addr16	2	6	—	$PC \leftarrow PC + 2 + \text{jdisp8}$ if CY = 0			
	BZ	\$addr16	2	6	—	$PC \leftarrow PC + 2 + \text{jdisp8}$ if Z = 1			
	BNZ	\$addr16	2	6	—	$PC \leftarrow PC + 2 + \text{jdisp8}$ if Z = 0			

- Notes**
1. For instructions that access the internal high-speed RAM area or perform no data access
 2. For instructions that access an area other than the internal high-speed RAM area

- Remarks**
1. One clock in the “Clock” columns is equal to one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).
 2. The values in the “Clock” column assumes that the internal ROM area contains programs.

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 + n	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 + n	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+n+m	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	x	x	x
		[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. For instructions that access the internal high-speed RAM area or perform no data access
 2. For instructions that access an area other than the internal high-speed RAM area

- Remarks**
1. One clock in the “Clock” columns is equal to one cycle of the CPU clock (f_{cpu}) selected by the processor clock control register (PCC).
 2. The values in the “Clock” column assumes that the internal ROM area contains programs.
 3. n indicates wait cycles to be inserted when an external expansion memory area is read from.
 4. m indicates wait cycles to be inserted when an external expansion memory area is written to.

28.3 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

Second Operand First Operand	#byte	A	r ^{Note}	sfr	saddr	laddr16	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 ADD SUB SUB SUBC SUBC AND AND OR OR XOR XOR CMP CMP	MOV XCH ADD ADD SUB SUB SUBC SUBC AND AND OR OR XOR XOR CMP CMP	MOV	MOV XCH	MOV XCH ADD ADD ADDC ADDC SUB SUB SUBC SUBC AND AND OR OR XOR XOR CMP 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
laddr16		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 when “r = A”

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

Second Operand 1st Operand	#word	AX	rp ^{Note}	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW ^{Note}						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

Second Operand First Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	\$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call/instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

Second Operand First Operand	AX	!addr16	!addr11	[addr5]	\$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

[MEMO]

★ **APPENDIX A DIFFERENCES BETWEEN μ PD78078, 78075B SUBSERIES, AND μ PD78070A**

The major differences between the μ PD78078, 78075B Subseries, and μ PD78070A are shown in Table A-1.

Table A-1. Major Differences between μ PD78078, 78075B Subseries, and μ PD78070A

Part Number		μ PD78078 Subseries	μ PD78075B Subseries	μ PD78070A	
Item					
Anti-EMI noise measure		Not provided	Provided	Not provided	
Product equipped with I ² C bus		Available	Not available	Available	
Supply voltage		V _{DD} = 1.8 to 5.5 V		V _{DD} = 2.7 to 5.5 V	
Internal ROM size		μ PD78076 : 48 Kbytes μ PD78078, 78P078 : 60 Kbytes	μ PD78074B : 32 Kbytes μ PD78075B : 40 Kbytes	Not provided	
Internal expansion RAM size		1024 bytes	None		
I/O port	Total	88		61	
	CMOS input	2			
	CMOS I/O	78		51	
	N-ch open-drain I/O	8			
Pins with ^{Note 1} additional functions	Pin with pull-up resistor	86 (78 for μ PD78P078)		51	
	Medium-voltage pin	8		Not provided	
	LED direct drive output	16	Not provided	4	
AV _{DD} pin		Provided	Not provided (AV _{REF0} pin functions alternately)	Provided	
External expansion function	Bus mode	Selectable from multiplexed bus mode or separate bus mode		Only separate bus mode	
	Memory expansion mode	Selectable from four types of memory expansion modes		Only full-address mode	
ROM correction function		Provided	Not provided		
Package		<ul style="list-style-type: none"> • 100-pin plastic QFP (Fine pitch) (14 x 14 mm)^{Note 2} • 100-pin plastic LQFP (Fine pitch) (14 x 14 mm) • 100-pin plastic QFP (14 x 20 mm) • 100-pin ceramic WQFN^{Note 3} 	<ul style="list-style-type: none"> • 100-pin plastic QFP (Fine pitch) (14 x 14 mm) • 100-pin plastic LQFP (Fine pitch) (14 x 14 mm) • 100-pin plastic QFP (14 x 20 mm) 	<ul style="list-style-type: none"> • 100-pin plastic QFP (Fine pitch) (14 x 14 mm) • 100-pin plastic LQFP (Fine pitch) (14 x 14 mm)^{Note 4} • 100-pin plastic QFP (14 x 20 mm) 	
Electrical specifications Recommended soldering conditions		Refer to the data sheet of individual product.			

- Notes**
1. The number of I/O ports includes the pins with additional functions.
 2. Not available in the Y subseries
 3. PROM version only
 4. Under development

[MEMO]

APPENDIX B DEVELOPMENT TOOLS

The following development tools are available for the development of systems which employ the μ PD78078 and 78078Y Subseries. Figure B-1 shows the configuration example of the tools.

Figure B-1. Development Tool Configuration (1/2)

(1) When using in-circuit emulator IE-78K0-NS

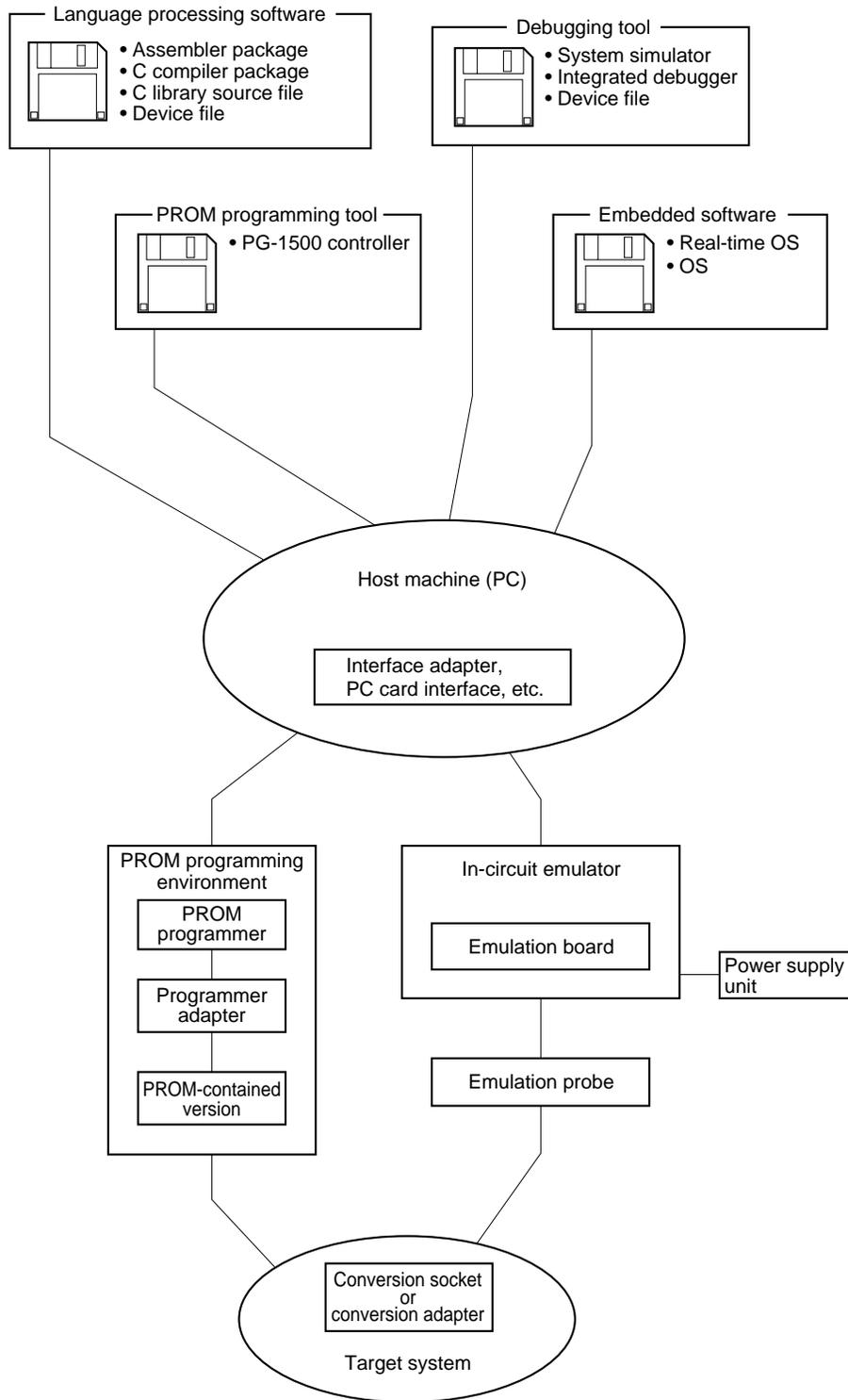
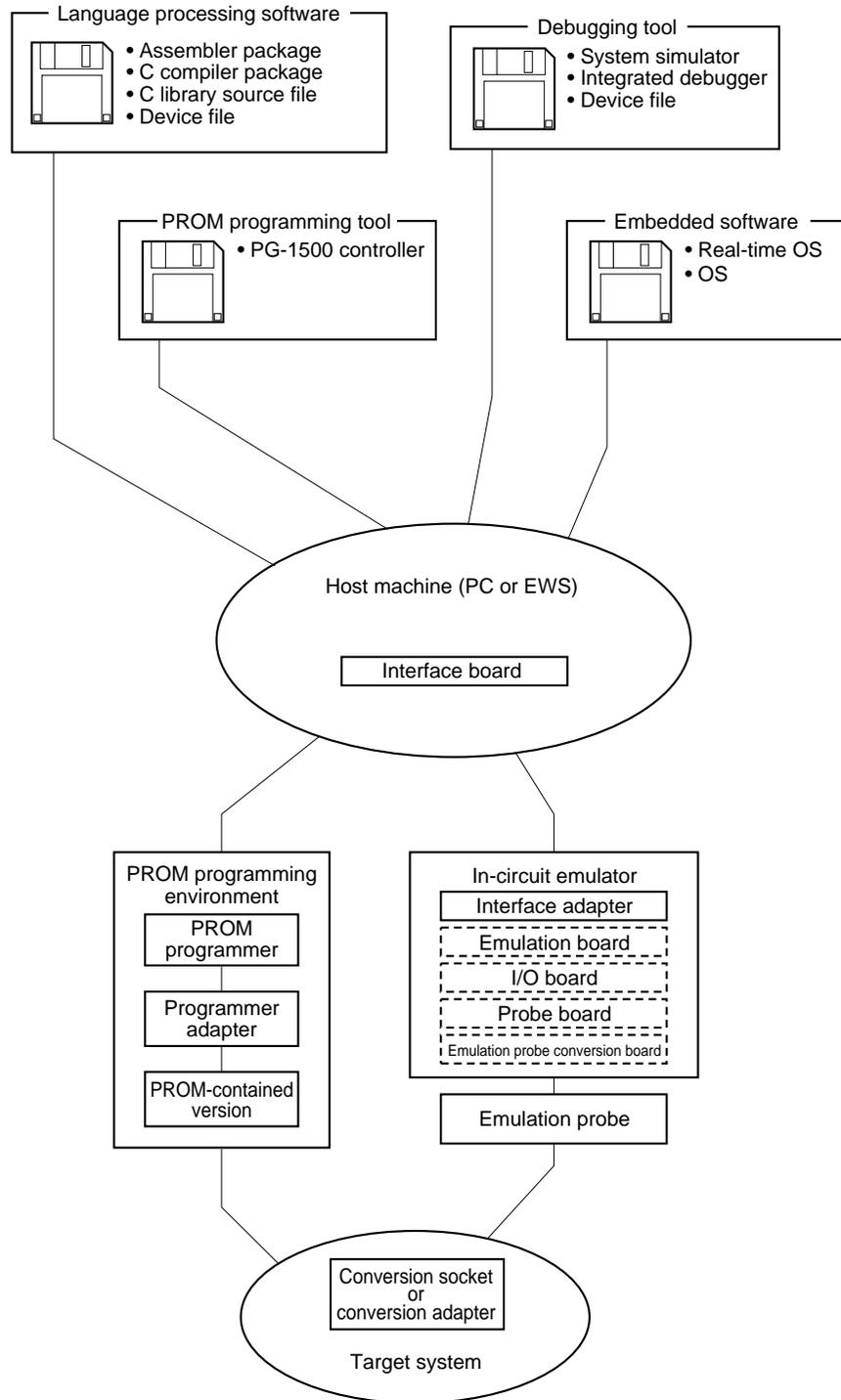


Figure B-1. Development Tool Configuration (2/2)

(2) When using in-circuit emulator IE-78001-R-A



Remark The parts shown within broken lines differ depending on the developing environment. Refer to **B.3.1 Hardware**.

B.1 Language Processing Software

<p>RA78K/0 Assembler Package</p>	<p>A program that converts a program written in mnemonic into object codes that microcomputers can process. Provided with functions to automatically perform generation of symbol table, optimizing processing of branch instructions, etc. Used in combination with separately available Device File (DF78078).</p> <p><Precautions for the use in PC environment> Although Assembler Package is a DOS-based application, it can be used in a Windows environment through the use of Project Manager (included in Assembler Package) on Windows.</p> <p>Part number: μSxxxxRA78K0</p>
<p>CC78K/0 C Compiler Package</p>	<p>A program which converts a program written in C language into object codes that microcomputers can process. Used in combination with separately available Assembler Package and Device File.</p> <p><Precautions for the use in PC environment> Although C Compiler Package is a DOS-based application, it can be used in Windows environment through the use of Project Manager (included in Assembler Package) on Windows.</p> <p>Part number: μSxxxxCC78K0</p>
<p>DF78078^{Note} Device File</p>	<p>A file which contains information peculiar to the device. Used in combination with separately available tools (RA78K/0, CC78K/0, SM78K0, ID78K0-NS, ID78K0). Supporting OS and host machines are dependent on the tool to be combined with.</p> <p>Part number: μSxxxxDF78078</p>
<p>CC78K/0-L C Library Source File</p>	<p>A source file of functions which configure the object library included in C Compiler package. Required when modifying the object library included in C Compiler Package for customization. Since this is a source file, its operation environment is independent from OS.</p> <p>Part number: μSxxxxCC78K0-L</p>

Note The DF78078 can commonly be used for all the products of the RA78K/0, CC78K/0, SM78K0, ID78K0-NS, and ID78K0.

Remark xxxx in the part number differs depending on the host machine and OS used.

μ Sxxxx RA78K0
 μ Sxxxx CC78K0
 μ Sxxxx DF78078
 μ Sxxxx CC78K0-L

xxxx	Host Machine	OS	Supply Media
AA13	PC-9800 series	Japanese Windows ^{Notes 1, 2}	3.5-inch 2HD FD
AB13	IBM PC/AT™ and compatibles	Japanese Windows ^{Notes 1, 2}	3.5-inch 2HC FD
BB13		English Windows ^{Notes 1, 2}	
3P16	HP9000 series 700™	HP-UX™ (rel. 9.05)	DAT (DDS)
3K13	SPARCstation™	SunOS™ (rel. 4.1.4)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS™ (RISC)	NEWS-OS™ (rel. 6.1)	3.5-inch 2HC FD

- Notes**
1. Operates also in DOS environment.
 2. Does not support WindowsNT™

B.2 PROM Writing Tools

B.2.1 Hardware

PG-1500 PROM Programmer	A PROM programmer that, by connecting the attached board and separately available PROM programmer adapter, is capable of programming single-chip microcomputers incorporating a PROM on stand-alone basis or through operation from the host machine. Also capable of programming typical 256-Kbit to 4-Mbit PROM.
PA-78P078GC PA-78P078GF PA-78P078KL-T PROM Programmer Adapter	A PROM programmer adapter for the μ PD78P078 and μ PD78P078Y. Used connected to the PG-1500. PA-78P078GC : 100-pin plastic QFP (GC-7EA, GC-8EU type) PA-78P078GF : 100-pin plastic QFP (GF-3BA type) PA-78P078KL-T : 100-pin ceramic WQFN (KL-T type)

B.2.2 Software

PG-1500 Controller	Connects PG-1500 and the host machine with serial and parallel interface, and controls the PG-1500 on the host machine. The PG-1500 controller is a DOS-based application. Use it with the DOS prompt on Windows.
	Part number: μ SxxxxPG1500

Remark xxxx in the part number differs depending on the host machine and OS used.

μ Sxxxx PG1500

xxxx	Host Machine	OS	Supply Media
5A13	PC-9800 series	MS-DOS (ver. 3.30 to ver. 6.2 ^{Note})	3.5-inch 2HD FD
5B13	IBM PC/AT and compatibles	Refer to B.4	3.5-inch 2HC FD

Note MS-DOS ver. 5.0 or later has a task swap function, but it cannot be used with the above software.

B.3 Debugging Tools

B.3.1 Hardware (1/2)

(1) When using in-circuit emulator IE-78K0-NS

IE-78K0-NS ^{Note} In-circuit Emulator	An in-circuit emulator to debug hardware and software when developing application systems that use the 78K/0 Series. Supports integrated debugger (ID78K0-NS). Used in combination with a power supply unit, emulation probe, and interface adapter to connect to the host machine.
IE-70000-MC-PS-B Power Supply Adapter	An adapter to supply voltage from AC100 to 240-V outlet.
IE-70000-98-IF-C ^{Note} Interface Adapter	An adapter required for using a PC-9800 series computer (except notebook-type personal computer) as the host machine for the IE-78K0-NS.
IE-70000-CD-IF ^{Note} PC Card Interface	A PC card and an interface cable required for using PC-9800 series notebook-type personal computer as the host machine for the IE-78K0-NS.
IE-70000-PC-IF-C ^{Note} Interface Adapter	An adapter required when using an IBM PC/AT and compatible as the host machine for the IE-78K0-NS.
IE-78078-NS-EM1 ^{Note} Emulation Board	A board to emulate peripheral hardware peculiar to the device. Used in combination with an in-circuit emulator.
NP-100GC Emulation Probe	A probe to connect an in-circuit emulator and a target system. For 100-pin plastic QFP (GC-7EA, GC-8EU type)
TGC-100SDW Conversion Adapter (refer to Figure B-2)	A conversion adapter to connect the board of a target system that is designed to mount 100-pin plastic QFP (GC-7EA, GC-8EU type) and the NP-100GC. The μ PD78P078KL-T and 78P078YKL-T (ceramic WQFN) can be mounted instead of connecting NP-100GC.
NP-100GF Emulation Probe	A probe to connect an in-circuit emulator and the target system. For 100-pin plastic QFP (GF-3BA type).
EV-9200GF-100 Conversion Socket (refer to Figure B-3)	A conversion socket to connect the board of a target system designed to mount 100-pin plastic QFP (GF-3BA type) to the NP-100GF.

Note Under development

- Remarks**
1. The NP-100GC and NP-100GF are products of Naito Densai Machidaseisakusho Co., Ltd.
Contact: Naito Densai Machidaseisakusho Co., Ltd (Tel: (044)822-3813)
 2. The TGC-100SDW is a product of TOKYO ELETECH Corporation.
Contact: Daimaru Kogyo Co., Ltd.
Tokyo Electronic Component Department (Tel: (03)3820-7112)
Osaka Electronic Component Department (Tel: (06)244-6672)
 3. The TGC-100SDW is sold singly.
 4. The EV-9200GF-100 is sold in a set of five.

B.3.1 Hardware (2/2)

(2) When using in-circuit emulator IE-78001-R-A

IE-78001-R-A ^{Note} In-circuit Emulator	An in-circuit emulator to debug hardware and software when developing application systems that use the 78K/0 Series. Supports integrated debugger (ID78K0). Used in combination with an interface adapter to connect to an emulation probe and the host machine.
IE-70000-98-IF-B or IE-70000-98-IF-C ^{Note} Interface Adapter	An adapter required for using a PC-9800 series (except notebook-type personal computer) as the host machine for the IE-78001-R-A.
IE-70000-PC-IF-B or IE-70000-PC-IF-C ^{Note} Interface adapter	An adapter required for using an IBM PC/AT or compatible as the host machine for the IE-78001-R-A.
IE-78000-R-SV3 Interface Adapter	An adapter and a cable required for using EWS as the host machine for the IE-78001-R-A. Used connected to the board in the IE-78001-R-A. Supports 10Base-5 for Ethernet™. A separately available adapter required for other systems.
IE-78078-NS-EM1 ^{Note} Emulation Board	A board to emulate peripheral hardware peculiar to the device. Used in combination with an in-circuit emulator and emulation probe conversion board.
IE-78K0-R-EX1 ^{Note} Emulation Probe Conversion Board	A board required for using the IE-78078-NS-EM1 with the IE-78001-R-A
IE-78078-R-EM Emulation board	A board to emulate peripheral hardware peculiar to the device (supports 3.0 to 5.5 V). Used in combination with the IE-78001-R-A.
EP-78064GC-R Emulation Probe	A probe to connect an in-circuit emulator and the target system. For 100-pin plastic QFP (GC-7EA, GC-8EU type).
TGC-100SDW Conversion Adapter (refer to Figure B-2)	A conversion adapter to connect the board of a target system designed to mount 100-pin plastic QFP (GC-7EA, GC-8EU type) and the EP-78064GC-R. The μ PD78P078KL-T or 78P078YKL-T (ceramic WQFN) can be mounted instead of connecting the EP-78064GC-R.
EP-78064GF-R Emulation Probe	A probe to connect an in-circuit emulator and the target system. For 100-pin plastic QFP (GF-3BA type).
EV-9200GF-100 Conversion Socket (refer to Figure B-3)	A conversion socket to connect the board of a target system designed to mount 100-pin plastic QFP (GF-3BA type) to the EP-78064GF-R.

Note Under development

- Remarks 1.** The TGC-100SDW is a product of TOKYO ELETECH Corporation.
 Contact: Daimaru Kogyo Co., Ltd.
 Tokyo Electronic Component Department (Tel: (03)3820-7112)
 Osaka Electronic Component Department (Tel: (06)244-6672)
- 2.** The TGC-100SDW is sold singly.
- 3.** The EV-9200GF-100 is sold in a set of five.

B.3.2 Software (1/2)

SM78K0 System Simulator	Capable of debugging in C source level or assembler level while simulating the operation of the target system on the host machine. The SM78K0 operates on Windows. The use of the SM78K0 enables the verification of logic and performance of applications independently from hardware development without using in-circuit emulator and improves the development efficiency and the software quality. Used in combination with separately available Device File (DF78078). Part number: μ SxxxxSM78K0
----------------------------	--

Remark xxxx in the part number differs depending on the host machine and OS used.

μ Sxxxx SM78K0

xxxx	Host Machine	OS	Supply Media
AA13	PC-9800 series	Japanese Windows ^{Notes 1, 2}	3.5-inch 2HD FD
AB13	IBM PC/AT and	Japanese Windows ^{Notes 1, 2}	3.5-inch 2HC FD
BB13	compatible	English Windows ^{Note}	

Note Does not support WindowsNT.

B.3.2 Software (2/2)

ID78K0-NS ^{Note} Integrated debugger (supporting in-circuit emulator IE-78K0-NS)	A control program to debug the 78K/0 Series. Adopting Windows on personal computers and OSF/Motif™ on EWS as graphical user interface, presents the appearance and the operability conforming to them. Enhancing the debugging function that supports C language, the trace result can be displayed in the C language level by using window integration function which correlates the source program, disassembly display, and memory display to the trace result. In addition, the debugging efficiency of programs using real-time OS can be improved by integrating function extension modules such as task debuggers and system performance analyzers. Used in combination with separately available Device File (DF78078). Part number: μ SxxxID78K0-NS, μ SxxxID78K0.
ID78K0 Integrated Debugger (supporting in-circuit emulator IE-78001-R-A)	

Note Under development

Remark xxxx in the part number differs depending on the host machine and OS used.

μ Sxxx ID78K0-NS

xxxx	Host Machine	OS	Supply Media
AA13	PC-9800 series	Japanese Windows ^{Note}	3.5-inch 2HD FD
AB13	IBM PC/AT and compatible	Japanese Windows ^{Note}	3.5-inch 2HC FD
BB13		English Windows ^{Note}	

Note Does not support WindowsNT.

μ Sxxx ID78K0

xxxx	Host Machine	OS	Supply Media
AA13	PC-9800 series	Japanese Windows ^{Note}	3.5-inch 2HD FD
AB13	IBM PC/AT and compatible	Japanese Windows ^{Note}	3.5-inch 2HC FD
BB13		English Windows ^{Note}	
3P16	HP9000 series 700	HP-UX (Rel. 9.05)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel. 4.1.4)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS (RISC)	NEWS-OS (Rel. 6.1)	3.5-inch 2HC FD

Note Does not support WindowsNT.

B.4 OS for IBM PC

The following OSs are supported for IBM PC.

Table B-1. OS for IBM PC

OS	Version
PC DOS	Ver. 5.02 to Ver. 6.3
	J6.1/V ^{Note} to J6.3/V ^{Note}
IBM DOS™	J5.02/V ^{Note}
MS-DOS	Ver. 5.0 to Ver. 6.22
	5.0/V ^{Note} to 6.2/V ^{Note}

Note Only English mode is supported.

Caution MS-DOS ver. 5.0 or later has a task swap function, but it cannot be used with the above software.

B.5 System Upgrading from Former-type In-circuit Emulator for 78K/0 Series to IE-78001-R-A

If the user already owns a former-type in-circuit emulator for the 78K/0 Series (IE-78000-R or IE-78000-R-A), it can be used in the same manner as the IE-78001-R-A by replacing the break board inside the main unit with the IE-78001-R-BK (under development).

Table B-2. System Upgrading from Former-type In-circuit Emulator for 78K/0 Series to IE-78001-R-A

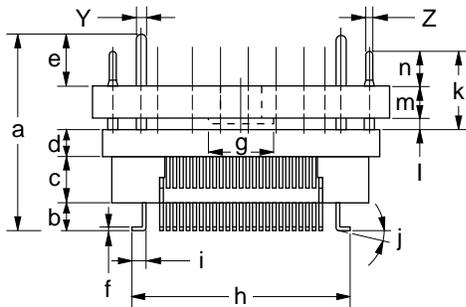
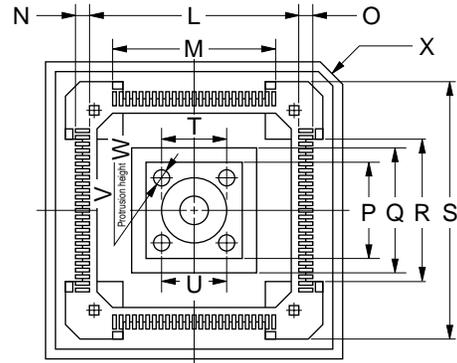
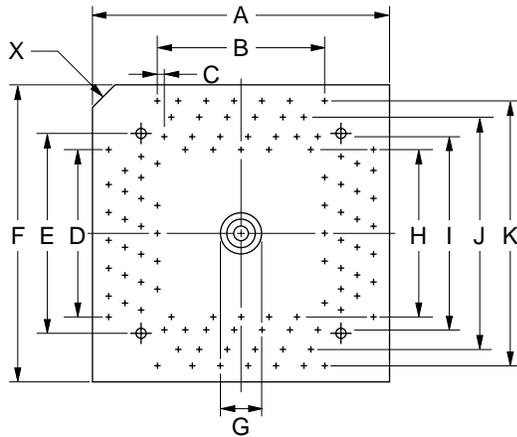
In-circuit Unit Owned	System Upgrading of Enclosure ^{Note}	Board to be Purchased
IE-78000-R	Required	IE-78001-R-BK
IE-78000-R-A	Not required	

Note To system up-grade the enclosure, the unit must be brought to NEC.

Drawing for Conversion Adapter (TGC-100SDW)

Figure B-2. TGC-100SDW Drawing (For Reference Only)

TGC-100SDW (TQPACK100SD + TQSOCKET100SDW)
 Package dimension (unit: mm)

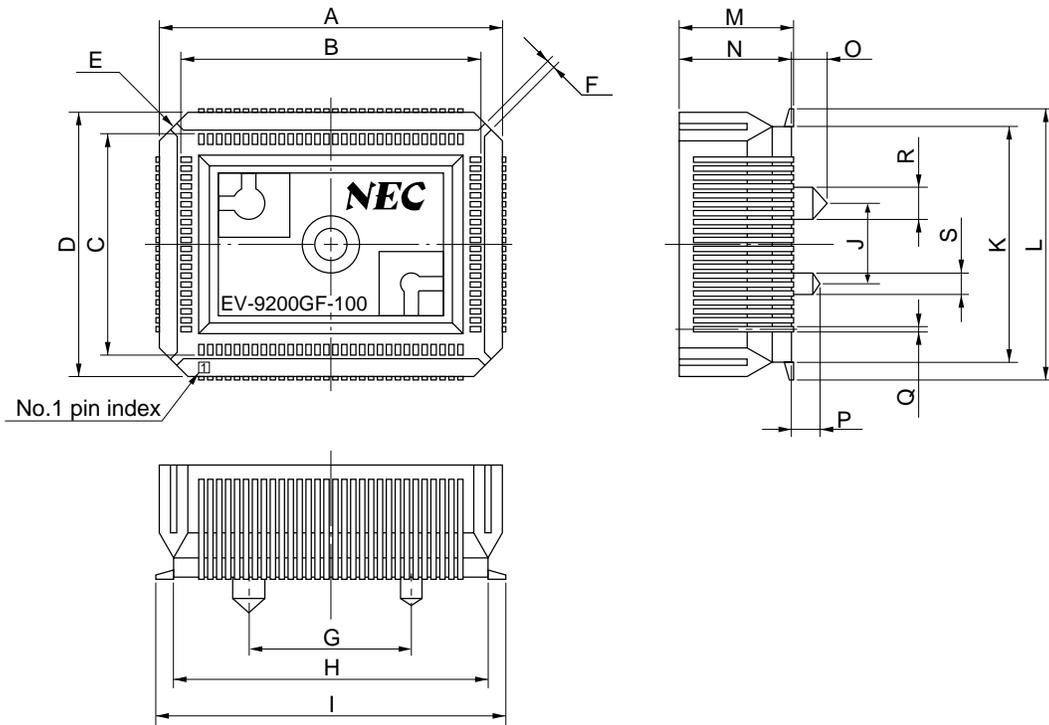


ITEM	MILLIMETERS	INCHES	ITEM	MILLIMETERS	INCHES
A	21.55	0.848	a	14.45	0.569
B	0.5x24=12	0.020x0.945=0.472	b	1.85±0.25	0.073±0.010
C	0.5	0.020	c	3.5	0.138
D	0.5x24=12	0.020x0.945=0.472	d	2.0	0.079
E	15.0	0.591	e	3.9	0.154
F	21.55	0.848	f	0.25	0.010
G	φ3.55	φ0.140	g	φ4.5	φ0.177
H	10.9	0.429	h	16.0	0.630
I	13.3	0.524	i	1.125±0.3	0.044±0.012
J	15.7	0.618	j	0-5°	0.000-0.197°
K	18.1	0.713	k	5.9	0.232
L	13.75	0.541	l	0.8	0.031
M	0.5x24=12.0	0.020x0.945=0.472	m	2.4	0.094
N	1.125±0.3	0.044±0.012	n	2.7	0.106
O	1.125±0.2	0.044±0.008	TGC-100SDW-G1E		
P	7.5	0.295			
Q	10.0	0.394			
R	11.3	0.445			
S	18.1	0.713			
T	φ5.0	φ0.197			
U	5.0	0.197			
V	4-φ1.3	4-φ0.051			
W	1.8	0.071			
X	C 2.0	C 0.079			
Y	φ0.9	φ0.035			
Z	φ0.3	φ0.012			

note: Product by TOKYO ELETECH CORPORATION.

Socket Drawing and Recommended Footprints (EV-9200GF-100)

Figure B-3. EV-9200GF-100 Drawing (For Reference Only)

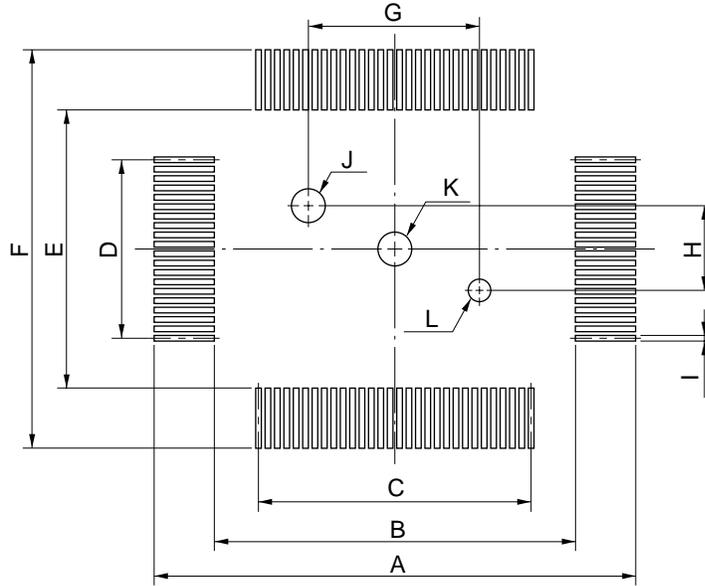


EV-9200GF-100-G0

ITEM	MILLIMETERS	INCHES
A	24.6	0.969
B	21	0.827
C	15	0.591
D	18.6	0.732
E	4-C 2	4-C 0.079
F	0.8	0.031
G	12.0	0.472
H	22.6	0.89
I	25.3	0.996
J	6.0	0.236
K	16.6	0.654
L	19.3	0.76
M	8.2	0.323
N	8.0	0.315
O	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 B-4. EV-9200GF-100 Recommended Footprints (For Reference Only)

Based on EV-9200GF-100
(2) Pad drawing (in mm)



EV-9200GF-100-P1

ITEM	MILLIMETERS	INCHES
A	26.3	1.035
B	21.6	0.85
C	$0.65 \pm 0.02 \times 29 = 18.85 \pm 0.05$	$0.026^{+0.001}_{-0.002} \times 1.142 = 0.742^{+0.002}_{-0.002}$
D	$0.65 \pm 0.02 \times 19 = 12.35 \pm 0.05$	$0.026^{+0.001}_{-0.002} \times 0.748 = 0.486^{+0.003}_{-0.002}$
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	$\phi 2.36 \pm 0.03$	$\phi 0.093^{+0.001}_{-0.002}$
K	$\phi 2.3$	$\phi 0.091$
L	$\phi 1.57 \pm 0.03$	$\phi 0.062^{+0.001}_{-0.002}$

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 C EMBEDDED SOFTWARE

For efficient program development and maintenance of the μ PD78078, 78078Y Subseries, the following embedded software is available.

Real-time OS (1/2)

RX78K/0 Real-time OS	<p>A real-time OS conforming to μITRON specifications.</p> <p>Added with the tool (configurator) to create the RX78K/0 nucleus and multiple information table. Used in combination with separately available Assembler Package (RA78K/0) and Device File (DF78078).</p> <p><Precautions for the use in PC environment></p> <p>Real-time OS is a DOS-based application. Use it with DOS prompt on Windows.</p> <p>Part number: μSxxxxRX78013-$\Delta\Delta\Delta\Delta$</p>
-------------------------	--

Caution When purchasing the RX78K/0, fill in the purchase application form in advance, and sign the License Agreement.

Remark xxxx and $\Delta\Delta\Delta\Delta$ in the part number differs depending on the host machine and OS used.

μ SxxxxRX78013- $\Delta\Delta\Delta\Delta$

$\Delta\Delta\Delta\Delta$	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

xxxx	Host Machine	OS	Supply Media
AA13	PC-9800 series	Japanese Windows ^{Notes1, 2}	3.5-inch 2HD FD
AB13	IBM PC/AT and compatibles	Japanese Windows ^{Notes1, 2}	3.5-inch 2HC FD
BB13		English Windows ^{Notes1, 2}	
3P16	HP9000 series 700	HP-UX (Rel. 9.05)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel. 4.1.4)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS (RISC)	NEWS-OS (Rel. 6.1)	3.5-inch 2HC FD
3R15			1/4-inch CGMT

Notes 1. Operates also in DOS environment.
2. Does not support WindowsNT.

Real-time OS (2/2)

MX78K0 OS	A μ TRON specification subset OS. Added with MX78K0 nucleus. Performs task management, event management, and time management. In task management, controls the execution order of tasks and performs processing to change the task to the one executed next. <Precautions for the use in PC environment> The MX78K0 is a DOS-based application. Use it with DOS prompt on Windows.
	Part number: μ SxxxxMX78K0- $\Delta\Delta\Delta$

Remark xxxx and $\Delta\Delta\Delta$ in the part number differs depending on the host machine and OS used.

μ SxxxxMX78K0- $\Delta\Delta\Delta$

$\Delta\Delta\Delta$	Product outline	Max. No. for Use in Mass Production
001	Evaluation object	Use for preproduction.
xx	Mass-production object	Use for mass production.
S01	Source program	Can be purchased only when purchasing mass-produced object.

xxxx	Host Machine	OS	Supply Media
AA13	PC-9800 series	Japanese Windows ^{Notes1, 2}	3.5-inch 2HD FD
AB13	IBM PC/AT and compatibles	Japanese Windows ^{Notes1, 2}	3.5-inch 2HC FD
BB13		English Windows ^{Notes1, 2}	
3P16	HP9000 series 700	HP-UX (Rel. 9.05)	DAT (DOS)
3K13	SPARCstation	SunOS (Rel. 4.1.4)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS (RISC)	NEWS-OS (Rel. 6.1)	3.5-inch 2HC FD

- Notes**
1. Operates also in DOS environment.
 2. Does not support WindowsNT.

APPENDIX D REGISTER INDEX

D.1 Register Name Index

[A]

- A/D conversion result register (ADCR) ... 295
- A/D converter input select register (ADIS) ... 298
- A/D converter mode register (ADM) ... 296
- Asynchronous serial interface mode register (ASIM) ... 462
- Asynchronous serial interface status register (ASIS) ... 464
- Automatic data transmit/receive address pointer (ADTP) ... 417
- Automatic data transmit/receive interval specify register (ADTI) ... 421
- Automatic data transmit/receive control register (ADTC) ... 420

[B]

- Baud rate generator control register (BRGC) ... 465

[C]

- Capture/compare control register 0 (CRC0) ... 193
- Capture/compare register 00 (CR00) ... 188
- Capture/compare register 01 (CR01) ... 188
- Compare register 10 (CR10) ... 230
- Compare register 20 (CR20) ... 230
- Compare register 50 (CR50) ... 253
- Compare register 60 (CR60) ... 253
- Correction address register 0 (CORAD0) ... 560
- Correction address register 1 (CORAD1) ... 560
- Correction control register (CORCN) ... 561

[D]

- D/A conversion value set register 0 (DACS0) ... 311
- D/A conversion value set register 1 (DACS1) ... 311
- D/A converter mode register (DAM) ... 312

[E]

8-bit timer mode control register 1 (TMC1) ... 233
8-bit timer mode control register 5 (TMC5) ... 256
8-bit timer mode control register 6 (TMC6) ... 257
8-bit timer output control register (TOC1) ... 234
8-bit timer register 1 (TM1) ... 230
8-bit timer register 2 (TM2) ... 230
8-bit timer register 5 (TM5) ... 253
8-bit timer register 6 (TM6) ... 253
External bus type select register (EBTS) ... 533
External interrupt mode register 0 (INTM0) ... 196, 508
External interrupt mode register 1 (INTM1) ... 299, 508

[I]

Internal extension RAM size switching register (IXS) ... 571
Internal memory size switching register (IMS) ... 532, 570
Interrupt mask flag register 0H (MK0H) ... 506
Interrupt mask flag register 0L (MK0L) ... 506
Interrupt mask flag register 1L (MK1L) ... 506, 524
Interrupt request flag register 0H (IF0H) ... 505
Interrupt request flag register 0L (IF0L) ... 505
Interrupt request flag register 1L (IF1L) ... 505, 524
Interrupt timing specify register (SINT) ... 327, 378

[K]

Key return mode register (KRM) ... 162, 525

[M]

Memory expansion mode register (MM) ... 161, 531

[O]

Oscillation mode selection register (OSMS) ... 170
Oscillation stabilization time select register (OSTS) ... 546

[P]

Port 0 (P0) ... 136
Port 1 (P1) ... 138
Port 2 (P2) ... 139, 141
Port 3 (P3) ... 143
Port 4 (P4) ... 144
Port 5 (P5) ... 145
Port 6 (P6) ... 146
Port 7 (P7) ... 148
Port 8 (P8) ... 150
Port 9 (P9) ... 151
Port 10 (P10) ... 153
Port 12 (P12) ... 155
Port 13 (P13) ... 156
Port mode register 0 (PM0) ... 157
Port mode register 1 (PM1) ... 157
Port mode register 2 (PM2) ... 157
Port mode register 3 (PM3) ... 157, 195, 235, 287, 292
Port mode register 5 (PM5) ... 157
Port mode register 6 (PM6) ... 157
Port mode register 7 (PM7) ... 157
Port mode register 8 (PM8) ... 157
Port mode register 9 (PM9) ... 157
Port mode register 10 (PM10) ... 157, 258
Port mode register 12 (PM12) ... 157, 497
Port mode register 13 (PM13) ... 157
Priority specify flag register 0H (PR0H) ... 507
Priority specify flag register 0L (PR0L) ... 507
Priority specify flag register 1L (PR1L) ... 507
Processor clock control register (PCC) ... 167
Pull-up resistor option register H (PUOH) ... 160
Pull-up resistor option register L (PUOL) ... 160

[R]

Real-time output buffer register H (RTBH) ... 496
Real-time output buffer register L (RTBL) ... 496
Real-time output port control register (RTPC) ... 498
Real-time output port mode register (RTPM) ... 497
Receive buffer register (RXB) ... 460
Receive shift register (RXS) ... 460

[S]

Sampling clock select register (SCS) ... 197, 510
Serial bus interface control register (SBIC) ... 325, 376
Serial I/O shift register 0 (SIO0) ... 319, 370
Serial I/O shift register 1 (SIO1) ... 417
Serial operating mode register 0 (CSIM0) ... 323, 374
Serial operating mode register 1 (CSIM1) ... 419
Serial operating mode register 2 (CSIM2) ... 461
16-bit timer mode control register (TMC0) ... 192
16-bit timer output control register (TOC0) ... 194
16-bit timer register (TM0) ... 189
16-bit timer register (TMS) ... 230
Slave address register (SVA) ... 319, 370
Successive approximation register (SAR) ... 295

[T]

Timer clock select register 0 (TCL0) ... 190, 285
Timer clock select register 1 (TCL1) ... 231
Timer clock select register 2 (TCL2) ... 271, 278, 290
Timer clock select register 3 (TCL3) ... 321, 373, 418
Timer clock select register 5 (TCL5) ... 254
Timer clock select register 6 (TCL6) ... 255
Transmit shift register (TXS) ... 460

[W]

Watchdog timer mode register (WDTM) ... 280
Watch timer mode control register (TMC2) ... 273

D.2 Register Symbol Index

[A]

ADCR: A/D conversion result register ... 295
ADIS: A/D converter input select register ... 298
ADM: A/D converter mode register ... 296
ADTC: Automatic data transmit/receive control register ... 420
ADTI: Automatic data transmit/receive interval specify register ... 421
ADTP: Automatic data transmit/receive address pointer ... 417
ASIM: Asynchronous serial interface mode register ... 462
ASIS: Asynchronous serial interface status register ... 464

[B]

BRGC: Baud rate generator control register ... 465

[C]

CORAD0: Correction address register 0 ... 560
CORAD1: Correction address register 1 ... 560
CORCN: Correction control register ... 561
CR00: Capture/compare register 00 ... 188
CR01: Capture/compare register 01 ... 188
CR10: Compare register 10 ... 230
CR20: Compare register 20 ... 230
CR50: Compare register 50 ... 253
CR60: Compare register 60 ... 253
CRC0: Capture/compare control register 0 ... 193
CSIM0: Serial operating mode register 0 ... 323, 374
CSIM1: Serial operating mode register 1 ... 419
CSIM2: Serial operating mode register 2 ... 461

[D]

DACS0: D/A conversion value set register 0 ... 311
DACS1: D/A conversion value set register 1 ... 311
DAM: D/A converter mode register ... 312

[E]

EBTS: External bus type select register ... 533

[I]

IF0H: Interrupt request flag register 0H ... 505
IF0L: Interrupt request flag register 0L ... 505
IF1L: Interrupt request flag register 1L ... 505, 524

IMS: Internal memory size switching register ... 532, 570
INTM0: External interrupt mode register 0 ... 196, 508
INTM1: External interrupt mode register 1 ... 299, 508
IXS: Internal extension RAM size switching register ... 571

[K]

KRM: Key return mode register ... 162, 525

[M]

MK0H: Interrupt mask flag register 0H ... 506
MK0L: Interrupt mask flag register 0L ... 506
MK1L: Interrupt mask flag register 1L ... 506, 524
MM: Memory expansion mode register ... 161, 531

[O]

OSMS: Oscillation mode selection register ... 170
OSTS: Oscillation stabilization time select register ... 546

[P]

P0: Port 0 ... 136
P1: Port 1 ... 138
P2: Port 2 ... 139, 141
P3: Port 3 ... 143
P4: Port 4 ... 144
P5: Port 5 ... 145
P6: Port 6 ... 146
P7: Port 7 ... 148
P8: Port 8 ... 150
P9: Port 9 ... 151
P10: Port 10 ... 153
P12: Port 12 ... 155
P13: Port 13 ... 156
PCC: Processor clock control register ... 167
PM0: Port mode register 0 ... 157
PM1: Port mode register 1 ... 157
PM2: Port mode register 2 ... 157
PM3: Port mode register 3 ... 157, 195, 235, 287, 292
PM5: Port mode register 5 ... 157
PM6: Port mode register 6 ... 157
PM7: Port mode register 7 ... 157
PM8: Port mode register 8 ... 157
PM9: Port mode register 9 ... 157

PM10: Port mode register 10 ... 157, 258
PM12: Port mode register 12 ... 157, 497
PM13: Port mode register 13 ... 157
PROH: Priority specify flag register 0H ... 507
PROL: Priority specify flag register 0L ... 507
PR1L: Priority specify flag register 1L ... 507
PUOH: Pull-up resistor option register H ... 160
PUOL: Pull-up resistor option register L ... 160

[R]

RTBH: Real-time output buffer register H ... 496
RTBL: Real-time output buffer register L ... 496
RTPC: Real-time output port control register ... 498
RTPM: Real-time output port mode register ... 497
RXB: Receive buffer register ... 460
RXS: Receive shift register ... 460

[S]

SAR: Successive approximation register ... 295
SBIC: Serial bus interface control register ... 325, 376
SCS: Sampling clock select register ... 197, 510
SINT: Interrupt timing specify register ... 327, 378
SIO0: Serial I/O shift register 0 ... 319, 370
SIO1: Serial I/O shift register 1 ... 417
SVA: Slave address register ... 319, 370

[T]

TCL0: Timer clock select register 0 ... 190, 285
TCL1: Timer clock select register 1 ... 231
TCL2: Timer clock select register 2 ... 271, 278, 290
TCL3: Timer clock select register 3 ... 321, 373, 418
TCL5: Timer clock select register 5 ... 254
TCL6: Timer clock select register 6 ... 255
TM0: 16-bit timer register ... 189
TM1: 8-bit timer register 1 ... 230
TM2: 8-bit timer register 2 ... 230
TM5: 8-bit timer register 5 ... 253
TM6: 8-bit timer register 6 ... 253
TMC0: 16-bit timer mode control register ... 192
TMC1: 8-bit timer mode control register 1 ... 233
TMC2: Watch timer mode control register ... 273
TMC5: 8-bit timer mode control register 5 ... 256

TMC6: 8-bit timer mode control register 6 ... 257
TMS: 16-bit timer register ... 230
TOC0: 16-bit timer output control register ... 194
TOC1: 8-bit timer output control register ... 234
TXS: Transmit shift register ... 460

[W]

WDTM: Watchdog timer mode register ... 280

APPENDIX E REVISION HISTORY

The revision history is shown below. The chapters appearing in the chapter column indicate those of the corresponding edition.

Version	Major revisions from previous version	Chapter
Second	μ PD78076, 78078, 78P078: Under development → Developed	Throughout this document
	μ PD78074, 78075, 78074Y, 78075Y, 78076Y, 78078Y, 78P078Y have been added as new members of this subseries.	
	Supply voltage (V_{DD}) range has been changed: 2.0 to 6.0 V → 1.8 to 5.5 V	
	1.9 Mask Options has been added.	CHAPTER 1 OUTLINE (μ PD78078 SUBSERIES)
	1.10 Differences with μPD78054 Subseries has been added.	
	Cautions on using the alternate function of port 13 pins have been added.	CHAPTER 3 PIN FUNCTION (μ PD78078 SUBSERIES)
		CHAPTER 6 PORT FUNCTIONS
		CHAPTER 16 D/A CONVERTER
	Change has been made to recommended connection of unused pins.	CHAPTER 3 PIN FUNCTION (μ PD78078 SUBSERIES)
	Change has been made to recommended connection of unused pins for P130/ANO0 and P131/ANO1 pins.	
	Caution when mask option pull-up resistor is not used has been added.	CHAPTER 6 PORT FUNCTIONS
	Caution has been added to Figure 7-4. Oscillation Mode Selection Register Format .	CHAPTER 7 CLOCK GENERATER
	Caution when an external clock is used as the main system clock source has been corrected.	CHAPTER 24 STANDBY FUNCTION
	The count clocks of 8-bit timer/event counters 5 and 6 have been corrected.	CHAPTER 10 8-BIT TIMER/ EVENT COUNTERS 5 AND 6
	A/D converter mode register format has been modified.	CHAPTER 15 A/D CONVERTER
	15.5 (7) AV_{DD} pin has been modified, and Figure 15-12. Handling of AV_{DD} Pin has been added.	
	Serial operating mode register 0 format has been modified.	CHAPTER 17 SERIAL INTERFACE CHANNEL 0 (μ PD78078 SUBSERIES)
	Figure 17-34. $\overline{SCK0}$/P27 Pin Configuration has been corrected.	
	The baud rate transmit/receive clock range that can be generated using the main system clock has been modified: 75 bps to 38400 → 75 bps to 76800 bps	CHAPTER 20 SERIAL INTERFACE CHANNEL 2
	CHAPTER 23 EXTERNAL DEVICE EXPANSION FUNCTION has been re-arranged so that the multiplexed bus mode and the separate bus mode are explained separately.	CHAPTER 23 EXTERNAL DEVICE EXPANSION FUNCTION
23.4 Example of Connection with Memory has been modified.		

Version	Major revisions from previous version	Chapter
Second	Table 24-1. HALT Mode Operating Status has been modified.	CHAPTER 24 STANDBY FUNCTION
	Table 24-3. STOP Mode Operating Status has been modified.	
	CHAPTER 26 ROM CORRECTION has been added.	—
	The development statuses for the following products have been changed from “Under Development” to “Developed”. PA-78P078GC, PA-78P078GF, PA-78P078KL-T, IE-78078-R-EM, DF78078	APPENDIX A DEVELOPMENT TOOLS
	The following products have been defined as products for maintenance purposes only. IE-75000-R, IE-78230-R, IE-78240-R, IE-78320-R, IE-78330-R	
	A system simulator (SM78K0) has been added.	
	A.4 Operating Systems for IBM PC has been added.	
	HP9000 Series 700 has been added as a host machine on which the software can run.	APPENDIX B EMBEDDED SOFTWARE
	An operating system (MX78K0) has been added.	
Third	The μ PD78074 and 78075 have been deleted.	Throughout this document
	The following package has been added to the μ PD78078 subseries. 100-pin plastic LQFP (fine pitch) (14 x 14 mm, resin thickness 1.4 mm) (planned)	
	Description has been added to the caution of Figure 8-6. 16-Bit Timer Output Control Register Format .	CHAPTER 8 16-BIT TIMER/ EVENT COUNTER
	Figure 12-3. Watchdog Timer Mode Register Format has been changed, and caution has been added.	CHAPTER 12 WATCHDOG TIMER
	Caution has been added to Serial I/O shift register 0 (SIO0) of μ PD78078Y subseries.	CHAPTER 18 SERIAL INTERFACE CHANNEL 0 (μ PD78078Y SUBSERIES)
	Figure 18-22. Example of Communication from Master to Slave (with 9-clock wait selected for both master and slave) has been corrected.	
	Figure 18-23. Example of Communication from Slave to Master (with 9-clock wait selected for both master and slave) has been corrected.	
	18.4.6 Restrictions in I²C bus mode has been added.	
	Caution has been added to Figure 19-5. Automatic Data Transmit/Receive Interval Specify Register Format .	CHAPTER 19 SERIAL INTERFACE CHANNEL 1
	Caution has been added to 19.4.3 (3) (d) Busy control option .	
	Description related to the port mode register 12 (PM12) has been added.	CHAPTER 21 REAL-TIME OUTPUT PORT
	The following products have been added. IE-78000-R-A, IE-70000-98-IF-B, IE-70000-98N-IF-B, IE-70000-PC-IF-B, IE-78000-R-SV3, ID78K0 The versions of the supported OS have been changed.	APPENDIX A DEVELOPMENT TOOLS, APPENDIX B EMBEDDED SOFTWARE

Edition	Major revisions from previous edition	Chapter
Fourth	<p>The following products have been changed from “under development” to “already developed”.</p> <p>μPD78078Y Subseries: μPD78076Y, 78078Y, 78P078Y</p>	Throughout
	<p>The following package has been added to the μPD78078Y Subseries.</p> <p>100-pin plastic LQFP (Fine pitch) (14 x 14 mm, resin thickness 1.4 mm)</p>	
	<p>Block diagrams of ports have been changed.</p> <p>Figure 6-5. Block Diagram of P20, P21, P23 to P26, Figure 6-6. Block Diagram of P22 and P27, Figure 6-7. Block Diagram of P20, P21, P23 to P26, Figure 6-8. Block Diagram of P22 and P27, Figure 6-9. Block Diagram of P30 to P37, Figure 6-16. Block Diagram of P71 and P72, Figure 6-20. Block Diagram of P100 and P101</p>	CHAPTER 6 PORT FUNCTION
	<p>Table 7-2. Relationship between CPU clock and Minimum Instruction Execution Time has been added.</p>	CHAPTER 7 CLOCK GENERATION CIRCUIT
	<p>Precautions for changing the operation mode of the serial interface channel 0 have been added.</p>	CHAPTER 17 SERIAL INTERFACE CHANNEL 0 (μ PD78078 SUBSERIES), CHAPTER 18 SERIAL INTERFACE CHANNEL 0 (μ PD78078Y SUBSERIES)
	<p>Release conditions of busy mode of the serial interface channel 0 (in SBI mode) have been changed.</p>	CHAPTER 17 SERIAL INTERFACE CHANNEL 0 (μ PD78078 SUBSERIES)
	<p>Cautions for the case that bus release signal (REL) and command signal (CMD) are misrecognized due to the bus line changing timing has been added.</p>	
	<p>Conditions and timings for the generation of interrupt request (INTSR, INTSER) when a reception error is generated have been corrected.</p>	CHAPTER 20 SERIAL INTERFACE CHANNEL 2
	<p>Precautions for using UART mode have been added.</p>	
	<p>Precautions for replacing PROM versions with mask ROM versions have been added.</p>	CHAPTER 27 μ PD78P078, 78P078Y
	<p>APPENDIX A DIFFERENCES BETWEEN μPD78078, 78075B SUBSERIES and μPD78070A has been added.</p>	APPENDIX A DIFFERENCES BETWEEN μ PD78078, 78075B SUBSERIES AND μ PD78070A
	<p>Entirely revised: Supports in-circuit emulator IE-78K0-NS</p>	APPENDIX B DEVELOPMENT TOOLS
	<p>Entirely revised: Fuzzy inference developing support system has been deleted.</p>	APPENDIX C EMBEDDED SOFTWARE

[MEMO]

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

Hong Kong, Philippines, Oceania

NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Asian Nations except Philippines

NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Europe

NEC Electronics (Europe) GmbH
Technical Documentation Dept.
Fax: +49-211-6503-274

Korea

NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: 02-528-4411

Japan

NEC Corporation
Semiconductor Solution Engineering Division
Technical Information Support Dept.
Fax: 044-548-7900

South America

NEC do Brasil S.A.
Fax: +55-11-6465-6829

Taiwan

NEC Electronics Taiwan Ltd.
Fax: 02-719-5951

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	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Technical Accuracy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>