NEC

User's Manual

μPD780208 Subseries

8-Bit Single-Chip Microcontrollers

μPD780204
μPD780204A
μPD780205
μPD780205A
μPD780206
μPD780208
μPD78P0208

Document No. U11302EJ4V0UM00 (4th edition)
Date Published July 2003 N CP(K)

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.

These commodities, technology or software, must be exported in accordance with the export administration regulations of the exporting country.
Diversion contrary to the law of that country is prohibited.

- **The information in this document is current as of January, 2003. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".
  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product 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": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)
(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.
(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02. 11-1

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC Electronics product in your application, please contact the NEC Electronics 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.

[GLOBAL SUPPORT]
http://www.necel.com/en/support/support.html

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

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 01

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87
- **Succursale Française**
  Vélizy-Villacoublay, France
  Tel: 01-30-67 58 00
- **Filiale Italiana**
  Milano, Italy
  Tel: 02-66 75 41
- **Branch The Netherlands**
  Eindhoven, The Netherlands
  Tel: 040-244 58 45
- **Tyskland Filial**
  Taeby, Sweden
  Tel: 08-63 80 820
- **United Kingdom Branch**
  Milton Keynes, UK
  Tel: 01908-691-133

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-558-3737

NEC Electronics Shanghai, Ltd.
Shanghai, P.R. China
Tel: 021-6841-1138

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377

NEC Electronics Singapore Pte. Ltd.
Novena Square, Singapore
Tel: 6253-8311

J03.4

Major Revisions in This Edition

Page	Description
Throughout	Addition of the following products to target products • μPD780204A • μPD780205A Deletion of the following package from target products • μPD78P0208KL-T (100-pin ceramic WQFN)
	CHAPTER 1 OUTLINE
p.29	• Update of **1.6 78K/0 Series Lineup**
p.32	• Addition of **Note** in **1.8 Overview of Functions**
p.33	• Addition of **Caution** in **Table 1-1 Mask Options in Mask ROM Versions**
	CHAPTER 2 PIN FUNCTIONS
p.42	• Addition of **2.2.12 V_{LOAD}**
p.43	• Modification of **Table 2-1 Types of Pin I/O Circuits**
	CHAPTER 3 CPU ARCHITECTURE
p.48	• Addition of **Caution** in **3.1 Memory Space**
p.67	• Modification of **Note** in **Table 3-3 Special-Function Register List**
	CHAPTER 4 PORT FUNCTIONS
p.90	• Addition of **Caution** in **4.2.6 Port 8**
p.91	• Addition of **Caution** in **4.2.7 Port 9**
p.92	• Addition of **Caution** in **4.2.8 Port 10**
p.93	• Addition of **Caution** in **4.2.9 Port 11**
p.94	• Addition of **Caution** in **4.2.10 Port 12**
	CHAPTER 5 CLOCK GENERATOR
p.103	• Addition of **Note** in **Figure 5-3 Format of Processor Clock Control Register**
	CHAPTER 6 16-BIT TIMER/EVENT COUNTER
p.133	• Modification of **Caution** in **Figure 6-8 Format of External Interrupt Mode Register**
p.144	• Modification of **6.6 (5) Valid edge setting**
	CHAPTER 8 WATCH TIMER
p.171	• Modification of **Caution** in **Figure 8-2 Format of Timer Clock Select Register 2**
	CHAPTER 9 WATCHDOG TIMER
p.178	• Modification of **Caution** in **Figure 9-2 Format of Timer Clock Select Register 2**
	CHAPTER 11 BUZZER OUTPUT CONTROLLER
p.188	• Modification of **Caution** in **Figure 11-2 Format of Timer Clock Select Register 2**
	CHAPTER 16 INTERRUPT AND TEST FUNCTIONS
p.340	• Addition of **Caution** in **Figure 16-2 Format of Interrupt Request Flag Register**
p.343	• Modification of **Caution** in **Figure 16-5 Format of External Interrupt Mode Register**
	CHAPTER 17 STANDBY FUNCTION
p.361	• Addition of description in **Table 17-1 HALT Mode Operating Status**
p.364	• Addition of description in **Table 17-3 STOP Mode Operating Status**
	CHAPTER 19 μPD78P0208
p.373	• Modification of **Table 19-2 Internal Memory Size Switching Register Setting Values**
	APPENDIX A DIFFERENCES BETWEEN μPD78044H, 780228, AND 780208 SUBSERIES
p.398	• Modification of description in **Table A-1 Major Differences Between μPD78044H, 780228, and 780208 Subseries**
	APPENDIX B DEVELOPMENT TOOLS
p.399	• Modification of description

The mark ★ shows major revised points.

INTRODUCTION

Readers This manual has been prepared for user engineers who wish to understand the functions of the μPD780208 Subseries and design and develop its application systems and programs.

Purpose This manual is intended to give users an understanding of the functions described in the **Organization** below.

Organization The μPD780208 Subseries manual consists of two parts: this manual and Instructions (common to the 78K/0 Series)

μPD780208 Subseries User's Manual (This manual)	78K/0 Series Instructions User's Manual
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions	• CPU functions • Instruction set • Explanation of each instruction

How to Read This Manual

It is assumed that the reader of this manual has general knowledge in the fields of electrical engineering, logic circuits, and microcontrollers.

- For an understanding of functions in general:
  → Read this manual in the order of the **CONTENTS**.
- For how to interpret the register format:
  → For a bit number enclosed in angle brackets, the bit name is defined as a reserved word in the RA78K0, and is defined in the header file named sfrbit.h in the CC78K0.
- To confirm the details of a register whose register name is known:
  → Refer to **APPENDIX C REGISTER INDEX**.
- For the details of μPD780208 Subseries instruction functions:
  → Refer to **78K/0 Series Instructions User's Manual (U12326E)**.
- For the electrical specifications of the μPD780208 Subseries:
  → Refer to the separate **μPD780204, 780205, 780206, 780208 Data Sheet (U10436E)** and **μPD78P0208 Data Sheet (U11295E)**.
- For application examples of the μPD780208 Subseries:
  → Refer to the separate **78K/0 Series Basics (II) Application Note (U10121E)**.

Conventions

Data significance:	Higher digits on the left and lower digits on the right
Active low representation:	$\overline{xxx}$ (overscore over pin or signal name)
Note:	Footnote for item marked with **Note** in the text
Caution:	Information requiring particular attention
Remark:	Supplementary information
Numerical representation:	Binary ..... xxxx or xxxxB
	Decimal xxxx
	Hexadecimal ..... xxxxH

Related Documents

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

Documents Related to Devices

Document Name	Document No.
μPD780204, 780205, 780206, 780208 Data Sheet	U10436E
μPD78P0208 Data Sheet	U11295E
μPD780208 Subseries User's Manual	This manual
78K/0 Series Instructions User's Manual	U12326E
78K/0 Series Basic (II) Application Note	U10121E

Documents Related to Software Development Tools (User's Manuals)

Document Name		Document No.
RA78K0 Assembler Package	Operation	U14445E
	Language	U14446E
	Structured Assembly Language	U11789E
CC78K0 C Compiler	Operation	U14297E
	Language	U14298E
SM78K Series System Simulator Ver. 2.30 or Later	Operation (Windows™ Based)	U15373E
	External Part User Open Interface Specification	U15802E
ID78K Series Integrated Debugger Ver. 2.30 or Later	Operation (Windows Based)	U15185E
RX78K0 Real-Time OS	Fundamentals	U11537E
	Installation	U11536E
Project Manager Ver. 3.12 or Later (Windows Based)		U14610E

Documents Related to Hardware Development Tools (User's Manuals)

Document Name	Document No.
IE-78K0-NS In-Circuit Emulator	U13731E
IE-78K0-NS-A In-Circuit Emulator	U14889E
IE-780208-NS-EM1 Emulation Board	U13691E
IE-78001-R-A In-Circuit Emulator	U14142E
IE-780208-R-EM Emulation Board	EEU-1501

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

Documents Related to PROM Writing (User's Manuals)

Document Name		Document No.
PG-1500 PROM Programmer		U11940E
PG-1500 Controller	PC-9800 Series (MS-DOS™ Based)	EEU-1291
	IBM PC Series (PC DOS™ Based)	U10540E

Other Related Documents

Document Name	Document No.
SEMICONDUCTOR SELECTION GUIDE - Products and Packages -	X13769X
Semiconductor Device Mount Manual	**Note**
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

Note See the "Semiconductor Device Mount Manual" website (http://www.necel.com/pkg/en/mount/index.html).

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

CONTENTS

LIST OF FIGURES (1/6)

LIST OF FIGURES (2/6)

LIST OF FIGURES (3/6)

LIST OF FIGURES (4/6)

LIST OF FIGURES (5/6)

LIST OF FIGURES (6/6)

LIST OF TABLES (1/2)

LIST OF TABLES (2/2)

CHAPTER 1 OUTLINE

1.1 Features

○ Internal high-capacity ROM and RAM

Item / Part Number	Program Memory		Data Memory			
	ROM	PROM	Internal High-Speed RAM	Buffer RAM	VFD Display RAM	Internal Expansion RAM
μPD780204 ★ μPD780204A	32 KB	—	1024 bytes	64 bytes	80 bytes	None
μPD780205 ★ μPD780205A	40 KB	—				
μPD780206	48 KB	—				1024 bytes
μPD780208	60 KB	—				
μPD78P0208	—	60 KB[Note 1]				1024 bytes[Note 2]

Notes 1. 32, 40, 48, or 60 KB can be selected by setting the internal memory size switching register (IMS).

2. 0 or 1024 bytes can be selected by setting the internal expansion RAM size switching register (IXS).

○ Minimum instruction execution time can be changed from high speed (0.4 μs: @ 5.0 MHz operation with main system clock) to ultra-low speed (122 μs: @ 32.768 kHz operation with subsystem clock)

○ 74 I/O ports

○ VFD controller/driver: 53 display outputs in total

- Segments: 9 to 40
- Digits: 2 to 16

○ 8-bit resolution A/D converter: 8 channels

- Power supply voltage (AV_{DD} = 4.0 to 5.5 V)

○ Serial interface: 2 channels

- 3-wire serial I/O/SBI/2-wire serial I/O mode: 1 channel
- 3-wire serial I/O mode (automatic transmit/receive function): 1 channel

○ Timer: 5 channels

- 16-bit timer/event counter: 1 channel
- 8-bit timer/event counter: 2 channels
- Watch timer: 1 channel
- Watchdog timer: 1 channel

○ 15 vectored interrupt sources

○ One test input

○ Two types of on-chip clock oscillators (for main and subsystem clocks)

○ Power supply voltage: V_{DD} = 2.7 to 5.5 V

1.2 Applications

Compact home stereo sets, cassette decks, tuners, CD players, VCRs, etc.

1.3 Ordering Information

Part Number	Package	Internal ROM
μPD780204GF-xxx-3BA	100-pin plastic QFP (14 x 20)	Mask ROM
★ μPD780204AGF-xxx-3BA	100-pin plastic QFP (14 x 20)	Mask ROM
μPD780205GF-xxx-3BA	100-pin plastic QFP (14 x 20)	Mask ROM
★ μPD780205AGF-xxx-3BA	100-pin plastic QFP (14 x 20)	Mask ROM
μPD780206GF-xxx-3BA	100-pin plastic QFP (14 x 20)	Mask ROM
μPD780208GF-xxx-3BA	100-pin plastic QFP (14 x 20)	Mask ROM
μPD78P0208GF-3BA	100-pin plastic QFP (14 x 20)	One-time PROM

Remark xxx indicates ROM code suffix.

1.4 Quality Grade

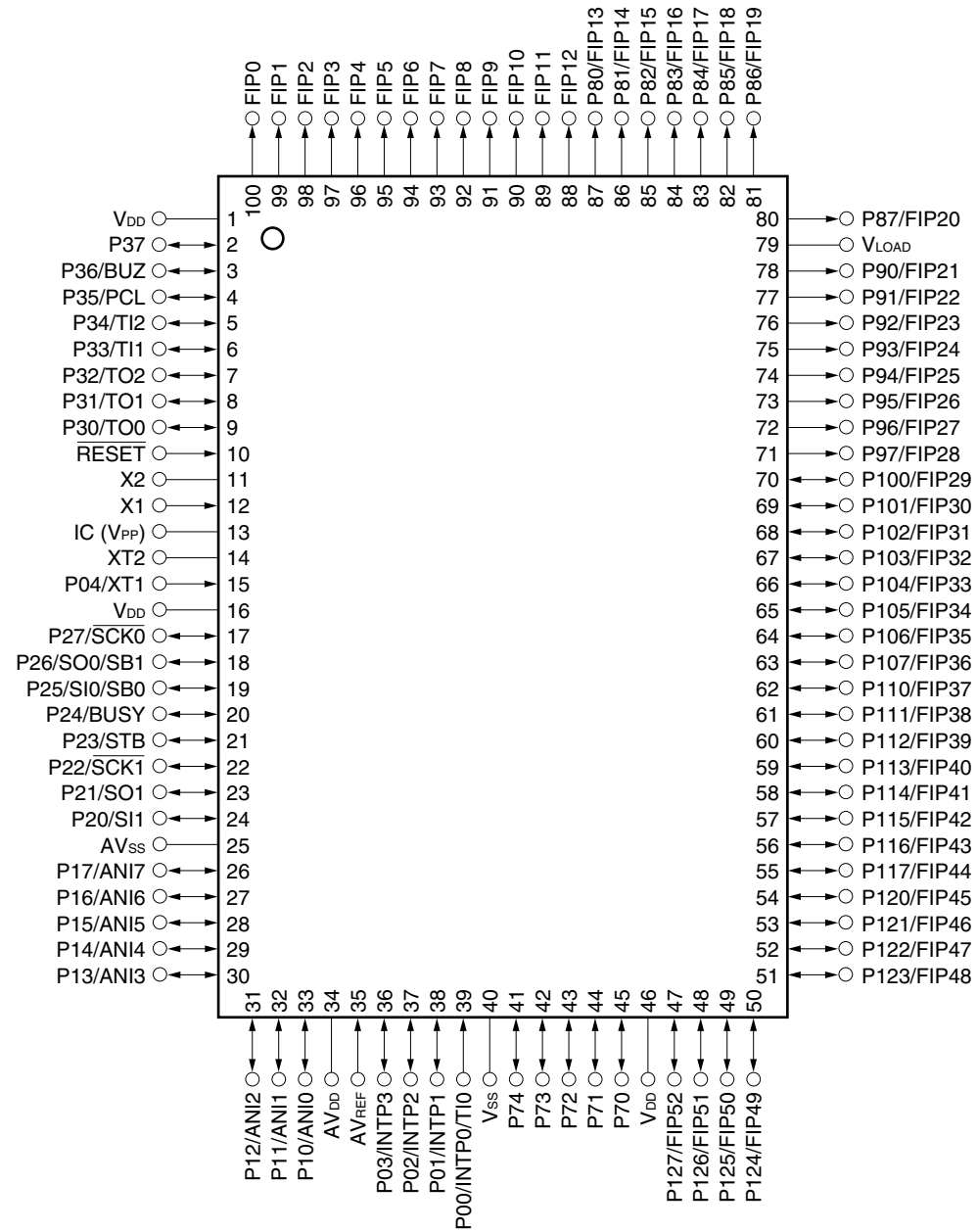
Part Number	Package	Quality Grade
μPD780204GF-xxx-3BA	100-pin plastic QFP (14 x 20)	Standard
★ μPD780204AGF-xxx-3BA	100-pin plastic QFP (14 x 20)	Standard
μPD780205GF-xxx-3BA	100-pin plastic QFP (14 x 20)	Standard
★ μPD780205AGF-xxx-3BA	100-pin plastic QFP (14 x 20)	Standard
μPD780206GF-xxx-3BA	100-pin plastic QFP (14 x 20)	Standard
μPD780208GF-xxx-3BA	100-pin plastic QFP (14 x 20)	Standard
μPD78P0208GF-3BA	100-pin plastic QFP (14 x 20)	Standard

Remark xxx indicates ROM code suffix.

1.5 Pin Configuration (Top View)

(1) Normal operating mode

- **100-pin plastic QFP (14 x 20)**
  μPD780204GF-xxx-3BA, 780204AGF-xxx-3BA, 780205GF-xxx-3BA, 780205AGF-xxx-3BA,
  μPD780206GF-xxx-3BA, 780208GF-xxx-3BA, 78P0208GF-3BA



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

2. Connect the AVDD pin to VDD.

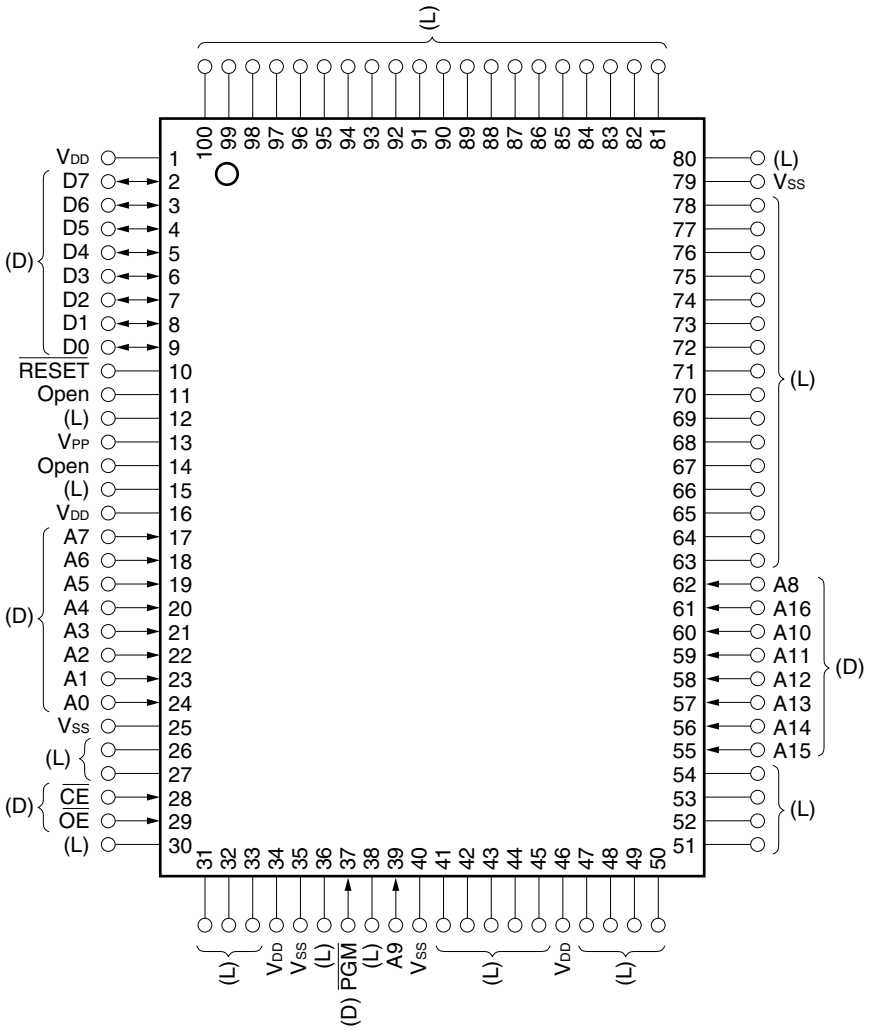
3. Connect the AVSS pin to VSS.

Remark The pin connection in parentheses is intended for the μPD78P0208.

ANI0 to ANI7:	Analog input
AV_{DD}:	Analog power supply
AV_{REF}:	Analog reference voltage
AV_{SS}:	Analog ground
BUSY:	Busy
BUZ:	Buzzer clock
FIP0 to FIP52:	Fluorescent indicator panel
IC:	Internally connected
INTP0 to INTP3:	External interrupt input
P00 to P04:	Port 0
P10 to P17:	Port 1
P20 to P27:	Port 2
P30 to P37:	Port 3
P70 to P74:	Port 7
P80 to P87:	Port 8
P90 to P97:	Port 9
P100 to P107:	Port 10
P110 to P117:	Port 11
P120 to P127:	Port 12
PCL:	Programmable clock
$\overline{\text{RESET}}$:	Reset
SB0, SB1:	Serial bus
$\overline{\text{SCK0}}$, $\overline{\text{SCK1}}$:	Serial clock
SI0, SI1:	Serial input
SO0, SO1:	Serial output
STB:	Strobe
TI0 to TI2:	Timer input
TO0 to TO2:	Timer output
V_{DD}:	Power supply
V_{LOAD}:	Negative power supply
V_{PP}:	Programming power supply
V_{SS}:	Ground
X1, X2:	Crystal (main system clock)
XT1, XT2:	Crystal (subsystem clock)

(2) PROM programming mode

- **100-pin plastic QFP (14 x 20)**
  μPD78P0208GF-3BA

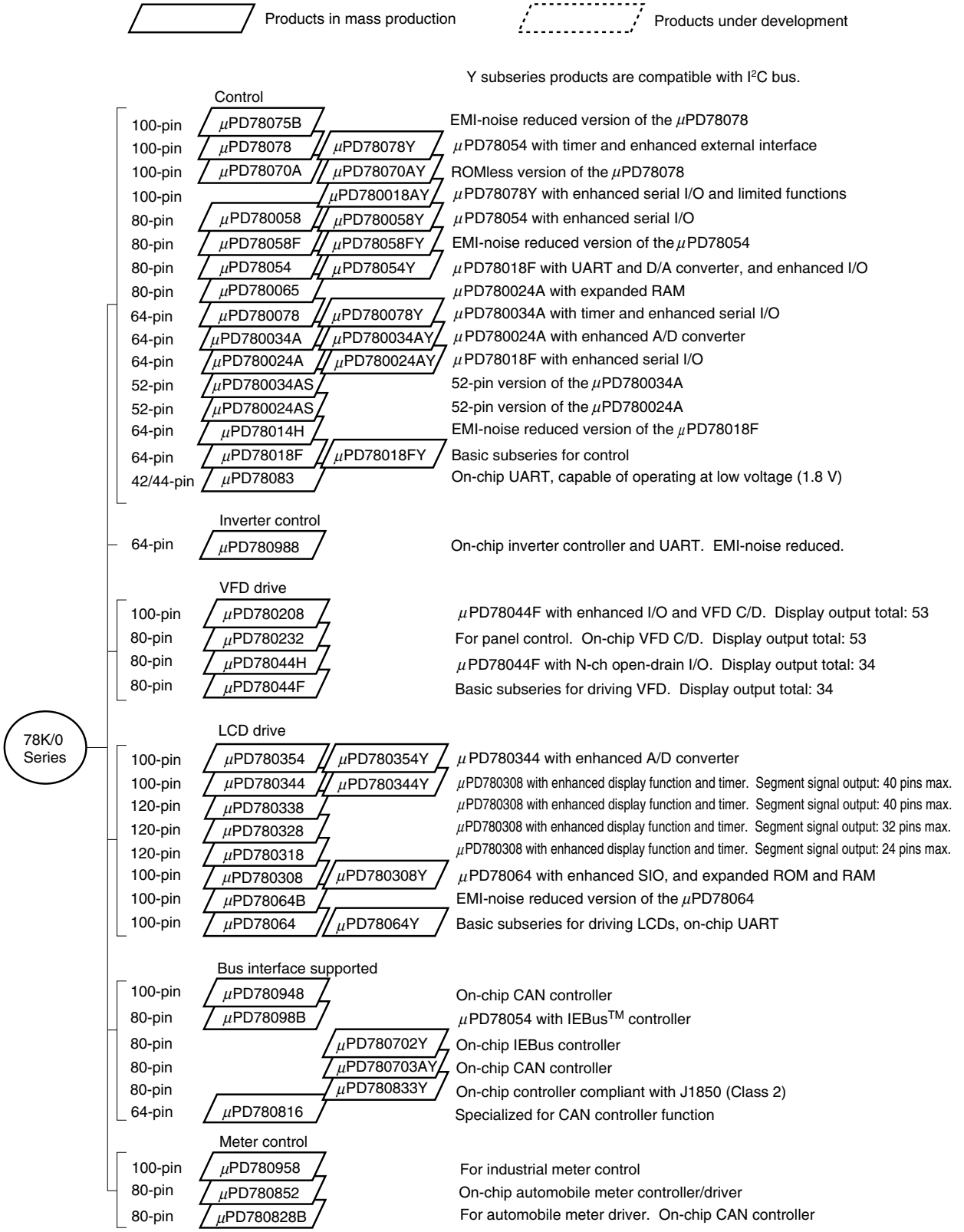


Cautions 1. (L): Connect independently to V_{SS} via a pull-down resistor.
2. (D): Connect via a driver.
3. V_{SS}: Connect to ground.
4. $\overline{RESET}$: Set to low level.
5. Open: Do not connect to anything.

A0 to A16: Address bus
$\overline{CE}$: Chip enable
D0 to D7: Data bus
$\overline{OE}$: Output enable
$\overline{PGM}$: Program
$\overline{RESET}$: Reset
V_{DD}: Power supply
V_{PP}: Programming power supply
V_{SS}: Ground

★ 1.6 78K/0 Series Lineup

The 78K/0 Series product lineup is illustrated below. The part numbers in boxes indicate subseries names.



Remark VFD (Vacuum Fluorescent Display) is referred to as FIP™ (Fluorescent Indicator Panel) in some documents, but the functions of the two are the same.

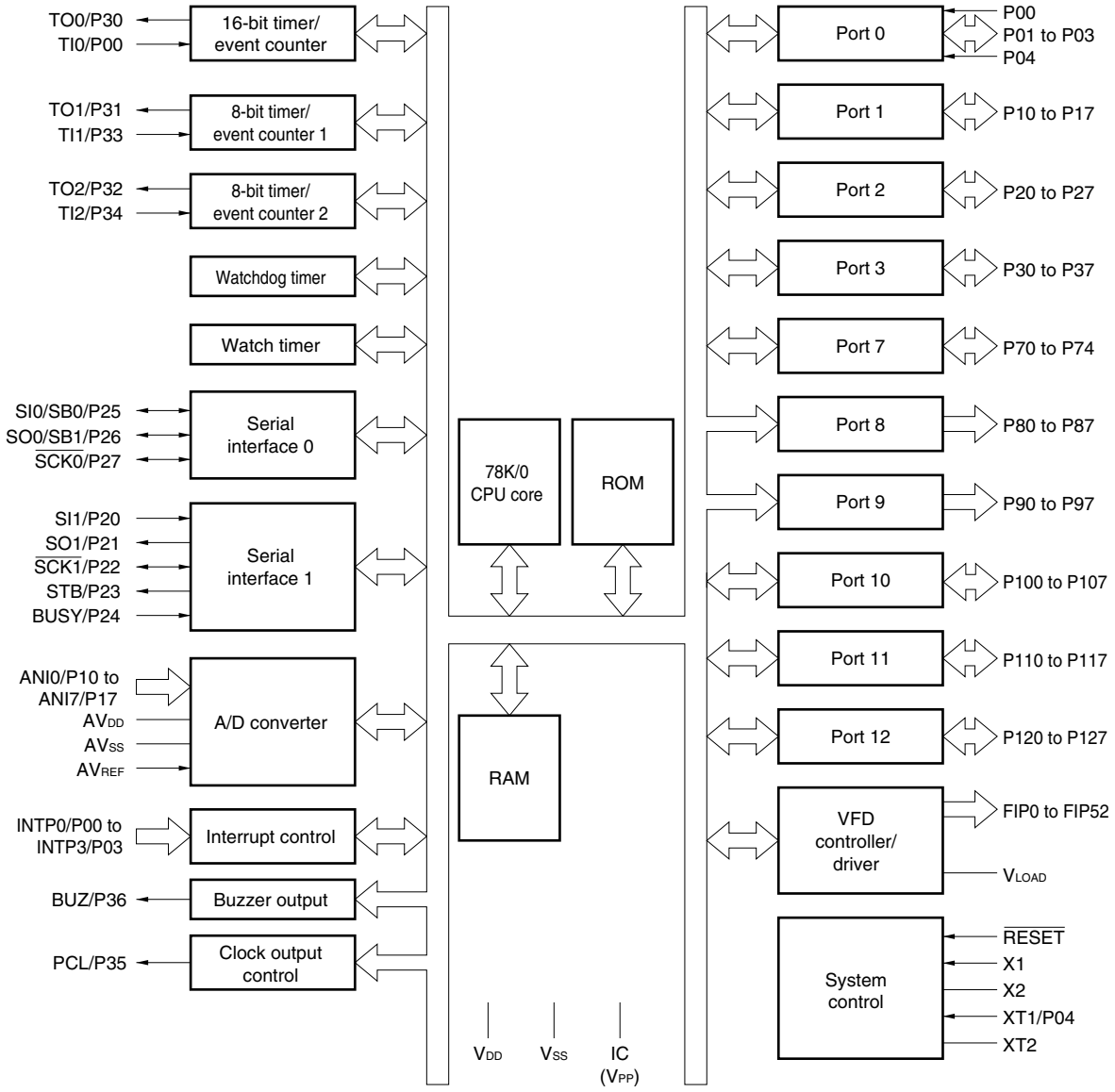
The following lists the main functional differences between subseries products.

- Non-Y subseries

Subseries Name \ Function		ROM Capacity (Bytes)	Timer				8-Bit A/D	10-Bit A/D	8-Bit D/A	Serial Interface	I/O	V_{DD} MIN. Value	External Expansion
			8-Bit	16-Bit	Watch	WDT							
Control	μPD78075B	32 K to 40 K	4 ch	1 ch	1 ch	1 ch	8 ch	–	2 ch	3 ch (UART: 1 ch)	88	1.8 V	√
	μPD78078	48 K to 60 K											
	μPD78070A	–									61	2.7 V	
	μPD780058	24 K to 60 K	2 ch							3 ch (time-division UART: 1 ch)	68	1.8 V	
	μPD78058F	48 K to 60 K								3 ch (UART: 1 ch)	69	2.7 V	
	μPD78054	16 K to 60 K										2.0 V	
	μPD780065	40 K to 48 K							–	4 ch (UART: 1 ch)	60	2.7 V	
	μPD780078	48 K to 60 K		2 ch			–	8 ch		3 ch (UART: 2 ch)	52	1.8 V	
	μPD780034A	8 K to 32 K		1 ch						3 ch (UART: 1 ch)	51		
	μPD780024A						8 ch	–					
	μPD780034AS						–	4 ch			39		–
	μPD780024AS						4 ch	–					
	μPD78014H						8 ch			2 ch	53		√
	μPD78018F	8 K to 60 K											
	μPD78083	8 K to 16 K		–	–					1 ch (UART: 1 ch)	33		–
Inverter control	μPD780988	16 K to 60 K	3 ch	**Note**	–	1 ch	–	8 ch	–	3 ch (UART: 2 ch)	47	4.0 V	√
VFD drive	μPD780208	32 K to 60 K	2 ch	1 ch	1 ch	1 ch	8 ch	–	–	2 ch	74	2.7 V	–
	μPD780232	16 K to 24 K	3 ch	–	–		4 ch				40	4.5 V	
	μPD78044H	32 K to 48 K	2 ch	1 ch	1 ch		8 ch			1 ch	68	2.7 V	
	μPD78044F	16 K to 40 K								2 ch			
LCD drive	μPD780354	24 K to 32 K	4 ch	1 ch	1 ch	1 ch	–	8 ch	–	3 ch (UART: 1 ch)	66	1.8 V	–
	μPD780344						8 ch	–					
	μPD780338	48 K to 60 K	3 ch	2 ch			–	10 ch	1 ch	2 ch (UART: 1 ch)	54		
	μPD780328										62		
	μPD780318										70		
	μPD780308	48 K to 60 K	2 ch	1 ch			8 ch	–	–	3 ch (time-division UART: 1 ch)	57	2.0 V	
	μPD78064B	32 K								2 ch (UART: 1 ch)			
	μPD78064	16 K to 32 K											
Bus interface supported	μPD780948	60 K	2 ch	2 ch	1 ch	1 ch	8 ch	–	–	3 ch (UART: 1 ch)	79	4.0 V	√
	μPD78098B	40 K to 60 K		1 ch					2 ch		69	2.7 V	–
	μPD780816	32 K to 60 K		2 ch			12 ch		–	2 ch (UART: 1 ch)	46	4.0 V	
Meter control	μPD780958	48 K to 60 K	4 ch	2 ch	–	1 ch	–	–	–	2 ch (UART: 1 ch)	69	2.2 V	–
Dash-board control	μPD780852	32 K to 40 K	3 ch	1 ch	1 ch	1 ch	5 ch	–	–	3 ch (UART: 1 ch)	56	4.0 V	–
	μPD780828B	32 K to 60 K									59		

Note 16-bit timer: 2 channels
10-bit timer: 1 channel

1.7 Block Diagram



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

2. Pin names in parentheses only apply to the μPD78P0208.

1.8 Overview of Functions

Item \ Part Number		μPD780204 μPD780204A	μPD780205 μPD780205A	μPD780206	μPD780208	μPD78P0208
Internal memory	ROM	Mask ROM				One-time PROM
		32 KB[Note 1]	40 KB[Note 1]	48 KB	60 KB	60 KB[Note 2]
	High-speed RAM	1024 bytes				
	Expansion RAM	–		1024 bytes		1024 bytes[Note 3]
	Buffer RAM	64 bytes				
	VFD display RAM	80 bytes				
General-purpose registers		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 (when operated at 5.0 MHz)				
	With subsystem clock selected	122 μs (when operated at 32.768 kHz)				
Instruction set		• 16-bit operation • Multiply/divide (8 bits x 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, and Boolean operation) • BCD adjust, and other related operations				
I/O ports (including VFD pins)		Total: 74 pins • CMOS input: 2 pins • CMOS I/O: 27 pins • N-ch open-drain I/O: 5 pins • P-ch open-drain I/O: 24 pins • P-ch open-drain output: 16 pins				
VFD controller/driver		Total of display output: 53 pins • Segments: 9 to 40 pins • Digits: 2 to 16 pins				
A/D converter		• 8-bit resolution x 8 channels • Power supply voltage: AV_{DD} = 4.0 to 5.5 V				
Serial interface		• 3-wire serial I/O/SBI/2-wire serial I/O mode selection possible: 1 channel • 3-wire serial I/O mode (maximum 64-byte on-chip automatic transmit/receive function): 1 channel				

★ **Notes 1.** The initial value of the internal memory size switching register (IMS) in the μPD780204A and 780205A is fixed to CFH (60 KB), regardless of the internal memory capacity. Therefore, set the values shown below for each product before use.
μPD780204A: C8H (32 KB)
μPD780205A: CAH (40 KB)

2. 32, 40, 48, or 60 KB can be selected by the internal memory size switching register (IMS).

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

Item \ Part Number		μPD780204 μPD780204A	μPD780205 μPD780205A	μPD780206	μPD780208	μPD78P0208
Timer		• 16-bit timer/event counter: 1 channel • 8-bit timer/event counter: 2 channels • Watch timer: 1 channel • Watchdog timer: 1 channel				
Timer output		3 outputs (14-bit PWM generation possible from one output)				
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz (@ 5.0 MHz 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 (@ 5.0 MHz operation with main system clock)				
Vectored interrupt sources	Maskable interrupts	Internal: 9, external: 4				
	Non-maskable interrupts	Internal: 1				
	Software interrupts	1				
Test input		Internal: 1				
Power supply voltage		V_{DD} = 2.7 to 5.5 V				
Package		100-pin plastic QFP (14 x 20)				

1.9 Mask Options

The mask ROM versions (μPD780204, μPD780204A, μPD780205, μPD780205A, μPD780206, and μPD780208) have mask options. By specifying the mask options when ordering, the pull-up resistors and pull-down resistors listed in Table 1-1 can be incorporated. When these resistors are necessary, the number of external components and mounting space can be saved by utilizing the mask options.

Table 1-1 shows the mask options provided in the μPD780208 Subseries products.

Table 1-1. Mask Options in Mask ROM Versions

Pin Name	Mask Option
P30/TO0 to P32/TO2, P33/TI1, P34/TI2, P35/PCL, P36/BUZ, P37	On-chip pull-down resistor can be specified in 1-bit units.
P70 to P74	On-chip pull-up resistor can be specified in 1-bit units.
FIP0 to FIP12	On-chip pull-down resistor can be specified in 1-bit units.
P80/FIP13 to P87/FIP20, P90/FIP21 to P97/FIP28, P100/FIP29 to P107/FIP36, P110/FIP37 to P117/FIP44, P120/FIP45 to P127/FIP52	On-chip pull-down resistor can be specified in 1-bit units. The connect destination of a pull-down resistor can be specified for V_{LOAD} or V_{SS} in 4-bit units from P80.

★ **Caution Adjust the number of pull-down resistors so that the total power dissipation (refer to 15.10 Calculating Total Power Dissipation) is not exceeded.**

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

2.1.1 Normal operating mode pins

(1) Port pins (1/2)

Pin Name	I/O	Function		After Reset	Alternate Function
P00	Input	Port 0. 5-bit I/O port.	Input only	Input	INTP0/TI0
P01	I/O		Input/output can be specified in 1-bit units. If used as an input port, use of an on-chip pull-up resistor can be specified by software settings.	Input	INTP1
P02					INTP2
P03					INTP3
P04[Note 1]	Input		Input only	Input	XT1
P10 to P17	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. If used as an input port, use of an on-chip pull-up resistor can be specified by software settings[Note 2].		Input	ANI0 to ANI7
P20	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units. If used as an input port, use of an on-chip pull-up resistor can be specified by software settings.		Input	SI1
P21					SO1
P22					$\overline{\text{SCK1}}$
P23					STB
P24					BUSY
P25					SI0/SB0
P26					SO0/SB1
P27					$\overline{\text{SCK0}}$
P30	I/O	Port 3. 8-bit I/O port. Input/output can be specified in 1-bit units. LEDs can be driven directly. If used as an input port, use of an on-chip pull-up resistor can be specified by software settings. In mask ROM versions, use of an on-chip pull-down resistor can be specified in 1-bit units with the mask option.		Input	TO0
P31					TO1
P32					TO2
P33					TI1
P34					TI2
P35					PCL
P36					BUZ
P37					—

Notes 1. When the P04/XT1 pin is used as an input port, set bit 6 (FRC) of the processor clock control register (PCC) to 1 (do not use the feedback resistor contained in the subsystem clock oscillator).

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

(1) Port pins (2/2)

Pin Name	I/O	Function	After Reset	Alternate Function
P70 to P74	I/O	Port 7. N-ch open-drain 5-bit I/O port. LEDs can be driven directly. Input/output can be specified in 1-bit units. In mask ROM versions, use of an on-chip pull-up resistor can be specified in 1-bit units with the mask option.	Input	—
P80 to P87	Output	Port 8. P-ch open-drain 8-bit high-withstanding-voltage output port. LEDs can be driven directly. In mask ROM versions, use of an on-chip pull-down resistor can be specified in 1-bit units with the mask option (connection to V_{LOAD} or V_{SS} is specifiable in 4-bit units).	Output	FIP13 to FIP20
P90 to P97	Output	Port 9. P-ch open-drain 8-bit high-withstanding-voltage output port. LEDs can be driven directly. In mask ROM versions, use of an on-chip pull-down resistor can be specified in 1-bit units with the mask option (connection to V_{LOAD} or V_{SS} is specifiable in 4-bit units).	Output	FIP21 to FIP28
P100 to P107	I/O	Port 10. P-ch open-drain 8-bit high-withstanding-voltage I/O port. Input/output can be specified in 1-bit units. LEDs can be driven directly. In mask ROM versions, use of an on-chip pull-down resistor can be specified in 1-bit units with the mask option (connection to V_{LOAD} or V_{SS} is specifiable in 4-bit units).	Input	FIP29 to FIP36
P110 to P117	I/O	Port 11. P-ch open-drain 8-bit high-withstanding-voltage I/O port. Input/output can be specified in 1-bit units. LEDs can be driven directly. In mask ROM versions, use of an on-chip pull-down resistor can be specified in 1-bit units with the mask option (connection to V_{LOAD} or V_{SS} is specifiable in 4-bit units).	Input	FIP37 to FIP44
P120 to P127	I/O	Port 12. P-ch open-drain 8-bit high-withstanding-voltage I/O port. Input/output can be specified in 1-bit units. LEDs can be driven directly. In mask ROM versions, use of an on-chip pull-down resistor can be specified in 1-bit units with the mask option (connection to V_{LOAD} or V_{SS} is specifiable in 4-bit units).	Input	FIP45 to FIP52

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

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt request inputs for which the valid edges (rising edge, falling edge, or both rising and falling edges) can be specified.	Input	P00/TI0
INTP1				P01
INTP2				P02
INTP3		External interrupt request input with falling edge detection		P03
SI0	Input	Serial interface serial data input	Input	P25/SB0
SI1				P20
SO0	Output	Serial interface serial data output	Input	P26/SB1
SO1				P21
SB0	I/O	Serial interface serial data input/output	Input	P25/SI0
SB1				P26/SO0
$\overline{SCK0}$	I/O	Serial interface serial clock input/output	Input	P27
$\overline{SCK1}$				P22
STB	Output	Serial interface automatic transmit/receive strobe output	Input	P23
BUSY	Input	Serial interface automatic transmit/receive busy input	Input	P24
TI0	Input	Input of external count clock to 16-bit timer (TM0)	Input	P00/INTP0
TI1		Input of external count clock to 8-bit timer (TM1)		P33
TI2		Input of external count clock to 8-bit timer (TM2)		P34
TO0	Output	16-bit timer (TM0) output (also used for 14-bit PWM output)	Input	P30
TO1		8-bit timer (TM1) output		P31
TO2		8-bit timer (TM2) output		P32
PCL	Output	Clock output (for trimming main system clock and subsystem clock)	Input	P35
BUZ	Output	Buzzer output	Input	P36
FIP0 to FIP12	Output	High withstanding voltage and high current output for VFD controller/driver display output. In mask ROM versions, use of an on-chip pull-down resistor can be specified with the mask option. The μPD78P0208 has on-chip pull-down resistors (connected to V_{LOAD}).	Output	—
FIP13 to FIP20	Output	High withstanding voltage and high current output for VFD controller/driver display output. In mask ROM versions, use of an on-chip pull-down resistor can be specified with the mask option. The μPD78P0208 has no on-chip pull-down resistors.	Output	P80 to P87
FIP21 to FIP28				P90 to P97
FIP29 to FIP36			Input	P100 to P107
FIP37 to FIP44				P110 to P117
FIP45 to FIP52				P120 to P127
V_{LOAD}	—	Pull-down resistor connection for VFD controller/driver	—	—
ANI0 to ANI7	Input	A/D converter analog input	Input	P10 to P17
AV_{REF}	Input	A/D converter reference voltage input	—	—
AV_{DD}	—	A/D converter analog power supply. Connect to V_{DD}.	—	—
AV_{SS}	—	A/D converter ground potential. Connect to V_{SS}.	—	—
$\overline{RESET}$	Input	System reset input	—	—

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

Pin Name	I/O	Function	After Reset	Alternate Function
X1	Input	Crystal resonator connection for main system clock oscillation	—	—
X2	—		—	—
XT1	Input	Crystal resonator connection for subsystem clock oscillation	Input	P04
XT2	—		—	—
V_{DD}	—	Positive power supply	—	—
V_{PP}	—	High-voltage application for program write/verify. Connect directly to V_{SS} in normal operation mode.	—	—
V_{SS}	—	Ground potential	—	—
IC	—	Internally connected. Connect directly to V_{SS}.	—	—

2.1.2 PROM programming mode pins (μPD78P0208 only)

Pin Name	I/O	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	I/O	Data bus
$\overline{CE}$	Input	PROM enable input/program pulse input
$\overline{OE}$	Input	Read strobe input to PROM
$\overline{PGM}$	Input	Program/program inhibit input in PROM programming mode
V_{DD}	—	Positive power supply
V_{SS}	—	Ground potential

2.2 Description of Pin Functions

2.2.1 P00 to P04 (Port 0)

These pins constitute a 5-bit I/O port. Besides serving as I/O port pins, they function as external interrupt request inputs, an external count clock input to the timer, a capture trigger signal input, and crystal resonator connection for subsystem clock oscillation.

The following operating modes can be specified in 1-bit units.

(1) Port mode

P00 and P04 function as input-only port pins and P01 to P03 function as I/O port pins.

P01 to P03 can be specified in input or output mode in 1-bit units using port mode register 0 (PM0). When they are used as input port pins, an on-chip pull-up resistor can be used by setting the pull-up resistor option register (PUO).

(2) Control mode

P00 to P04 function as external interrupt request inputs, an external count clock input to the timer, and crystal connection for subsystem clock oscillation.

(a) INTP0 to INTP3

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

(b) TI0

TI0 is a pin for inputting the external count clock to the 16-bit timer/event counter.

(c) XT1

Crystal connection pin for subsystem clock oscillation

2.2.2 P10 to P17 (Port 1)

These pins constitute an 8-bit I/O port. Besides serving as I/O port pins, they function as A/D converter analog inputs.

The following operating modes can be specified in 1-bit units.

(1) Port mode

P10 to P17 function as an 8-bit I/O port. They can be specified in input or output mode in 1-bit units using port mode register 1 (PM1). When they are used as input port pins, an on-chip pull-up resistor can be used by setting the pull-up resistor option register (PUO).

(2) Control mode

P10 to P17 function as A/D converter analog input pins (ANI0 to ANI7). The on-chip pull-up resistors are automatically disabled when the pins are specified for analog input.

2.2.3 P20 to P27 (Port 2)

These pins constitute an 8-bit I/O port. Besides serving as I/O port pins, they function as serial interface data I/O, clock I/O, automatic transmit/receive busy input, and strobe output pins.

The following operating modes can be specified in 1-bit units.

(1) Port mode

P20 to P27 function as an 8-bit I/O port. They can be specified in input or output mode in 1-bit units using port mode register 2 (PM2). When they are used as input port pins, an on-chip pull-up resistor can be used by setting the pull-up resistor option register (PUO).

(2) Control mode

P20 to P27 function as serial interface data I/O, clock I/O, automatic transmit/receive busy input, and strobe output pins.

(a) SI0, SI1, SO0, SO1

Serial interface serial data I/O pins

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

Serial interface serial clock I/O pins

(c) SB0 and SB1

NEC Electronics standard serial bus interface I/O pins

(d) BUSY

Serial interface automatic transmit/receive busy input pin

(e) STB

Serial interface automatic transmit/receive strobe output pin

Caution If port 2 is used as serial interface pins, the I/O and output latches must be set according to the function. For the setting method, refer to Figure 13-3 Format of Serial Operating Mode Register 0 and Figure 14-3 Format of Serial Operating Mode Register 1.

2.2.4 P30 to P37 (Port 3)

These pins constitute an 8-bit I/O port. Besides serving as I/O port pins, they function as timer I/O, clock output, and buzzer output pins.

In mask ROM versions, use of pull-down resistors can be specified with the mask option.

Port 3 can drive LEDs directly.

The following operating modes can be specified in 1-bit units.

(1) Port mode

P30 to P37 function as an 8-bit I/O port. They can be specified in input or output mode in 1-bit units using port mode register 3 (PM3). When they are used as input port pins, an on-chip pull-up resistor can be used by setting the pull-up resistor option register (PUO).

(2) Control mode

P30 to P37 function as timer I/O, clock output, and buzzer output pins.

(a) TI1 and TI2

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

(b) TO0 to TO2

Timer output pins

(c) PCL

Clock output pin

(d) BUZ

Buzzer output pin

2.2.5 P70 to P74 (Port 7)

These pins constitute a 5-bit I/O port. They can be specified in input or output mode in 1-bit units using port mode register 7 (PM7).

Port 7 can drive LEDs directly.

P70 to P74 are N-ch open-drain outputs. In mask ROM versions, use of pull-up resistors can be specified with the mask option.

2.2.6 P80 to P87 (Port 8)

These pins constitute an 8-bit output-only port. Besides serving as output port pins, they function as display outputs for the VFD controller/driver.

Port 8 can drive LEDs directly.

The following operating modes can be specified in 1-bit units.

(1) Port mode

P80 to P87 function as an 8-bit output-only port.

P80 to P87 are P-ch open-drain outputs. In mask ROM versions, use of pull-down resistors can be specified with the mask option.

(2) Control mode

P80 to P87 function as the display output pins of the VFD controller/driver (FIP13 to FIP20).

2.2.7 P90 to P97 (Port 9)

These pins constitute an 8-bit output-only port. Besides serving as output port pins, they function as display outputs for the VFD controller/driver.

Port 9 can drive LEDs directly.

The following operating modes can be specified in 1-bit units.

(1) Port mode

P90 to P97 function as an 8-bit output-only port.

P90 to P97 are P-ch open-drain outputs. In mask ROM versions, use of pull-down resistors can be specified with the mask option.

(2) Control mode

P90 to P97 function as the display output pins of the VFD controller/driver (FIP21 to FIP28).

2.2.8 P100 to P107 (Port 10)

These pins constitute an 8-bit I/O port. Besides serving as I/O port pins, they function as display outputs for the VFD controller/driver.

Port 10 can drive LEDs directly.

The following operating modes can be specified in 1-bit units.

(1) Port mode

P100 to P107 function as an 8-bit I/O port. They can be specified in input or output mode in 1-bit units using port mode register 10 (PM10).

P100 to P107 are P-ch open-drain outputs. In mask ROM versions, use of pull-down resistors can be specified with the mask option.

(2) Control mode

P100 to P107 function as display output pins for the VFD controller/driver (FIP29 to FIP36).

2.2.9 P110 to P117 (Port 11)

These pins constitute an 8-bit I/O port. Besides serving as I/O port pins, they function as display outputs for the VFD controller/driver.

Port 11 can drive LEDs directly.

The following operating modes can be specified in 1-bit units.

(1) Port mode

P110 to P117 function as an 8-bit I/O port. They can be specified in input or output mode in 1-bit units using port mode register 11 (PM11).

P110 to P117 are P-ch open-drain outputs. In mask ROM versions, use of pull-down resistors can be specified with the mask option.

(2) Control mode

P110 to P117 function as display output pins for the VFD controller/driver (FIP37 to FIP44).

2.2.10 P120 to P127 (Port 12)

These pins constitute an 8-bit I/O port. Besides serving as I/O port pins, they function as display outputs for the VFD controller/driver.

Port 12 can drive LEDs directly.

The following operating modes can be specified in 1-bit units.

(1) Port mode

P120 to P127 function as an 8-bit I/O port. They can be specified in input or output mode in 1-bit units using port mode register 12 (PM12).

P120 to P127 are P-ch open-drain outputs. In mask ROM versions, use of pull-down resistors can be specified with the mask option.

(2) Control mode

P120 to P127 function as display output pins for the VFD controller/driver (FIP45 to FIP52).

2.2.11 FIP0 to FIP12

These are display output pins for the VFD controller/driver.

FIP0 to FIP12 are P-ch open-drain outputs. In mask ROM versions, use of pull-down resistors can be specified with the mask option. The μPD78P0208 contains pull-down resistors at FIP0 to FIP12 (connected to V_{LOAD}).

★ **2.2.12 V_{LOAD}**

This is the pull-down resistor connection pin of the VFD controller/driver.

2.2.13 AV_{REF}

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

2.2.14 AV_{DD}

This pin supplies power for A/D converter operations.

Always make this pin the same potential as the V_{DD} pin even if the A/D converter is not used.

2.2.15 AV_{SS}

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

Always make this pin the same potential as the V_{SS} pin even if the A/D converter is not used.

2.2.16 $\overline{\text{RESET}}$

This is an active-low system reset input pin.

2.2.17 X1 and X2

These are crystal resonator connection pins for main system clock oscillation.

For external clock supply, input the clock to X1 and its inverted signal to X2.

2.2.18 XT1 and XT2

These are crystal resonator connection pins for subsystem clock oscillation.

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

2.2.19 V_{DD}

This is the positive power supply pin.

2.2.20 V_{SS}

This is the ground potential pin.

2.2.21 V_{PP} (μPD78P0208 only)

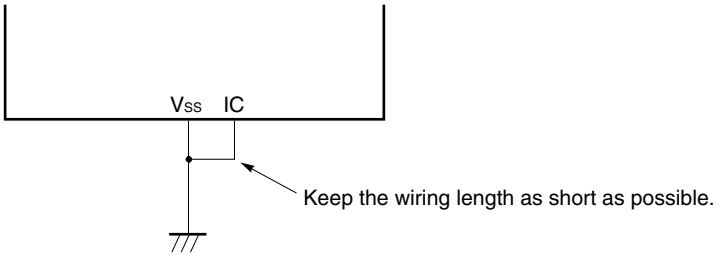
A high-voltage should be applied to this pin during PROM programming mode setting and in program write/verify mode. Connect directly to V_{SS} in normal operation mode.

2.2.22 IC (mask ROM version only)

The IC (Internally Connected) pin sets a test mode in which the μPD780204, 780204A, 780205, 780205A, 780206, and 780208 are tested before shipment. In normal operation mode, connect the IC pin directly to the V_{SS} pin with as short a wiring length as possible.

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

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



2.3 Pin I/O Circuits and Recommended Connection of Unused Pins

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

★ **Table 2-1. Types of Pin I/O Circuits (1/2)**

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

Note Leave open when an on-chip pull-down resistor is specified by the mask option.

★ **Table 2-1. Types of Pin I/O Circuits (2/2)**

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
μPD78P0208			
P30/TO0	5-A	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P31/TO1			
P32/TO2			
P33/TI1	8-A		
P34/TI2			
P35/PCL	5-A		
P36/BUZ			
P37			
Mask ROM version			
P70 to P74	13-B	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor**Note**. Output: Leave open.
FIP0 to FIP12	14-A	Output	Leave open.
P80/FIP13 to P87/FIP20			
P90/FIP21 to P97/FIP28			
P100/FIP29 to P107/FIP36	15-C	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor**Note**. Output: Leave open.
P110/FIP37 to P117/FIP44			
P120/FIP45 to P127/FIP52			
IC	—	—	Connect directly to V_{SS}.
μPD78P0208			
P70 to P74	13-D	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
FIP0 to FIP12	14	Output	Leave open.
P80/FIP13 to P87/FIP20	14-B	Output	Leave open.
P90/FIP21 to P97/FIP28			
P100/FIP29 to P107/FIP36	15-B	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P110/FIP37 to P117/FIP44			
P120/FIP45 to P127/FIP52			
V_{PP}	—	—	Connect directly to V_{SS}.
$\overline{RESET}$	2	Input	—
XT2	16	—	Leave open.
AV_{REF}	—		Connect directly to V_{SS}.
AV_{DD}			Connect directly to V_{DD}.
AV_{SS}			Connect directly to V_{SS}.
V_{LOAD}			

Note Leave open when an on-chip pull-up or pull-down resistor is specified by the mask option.

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

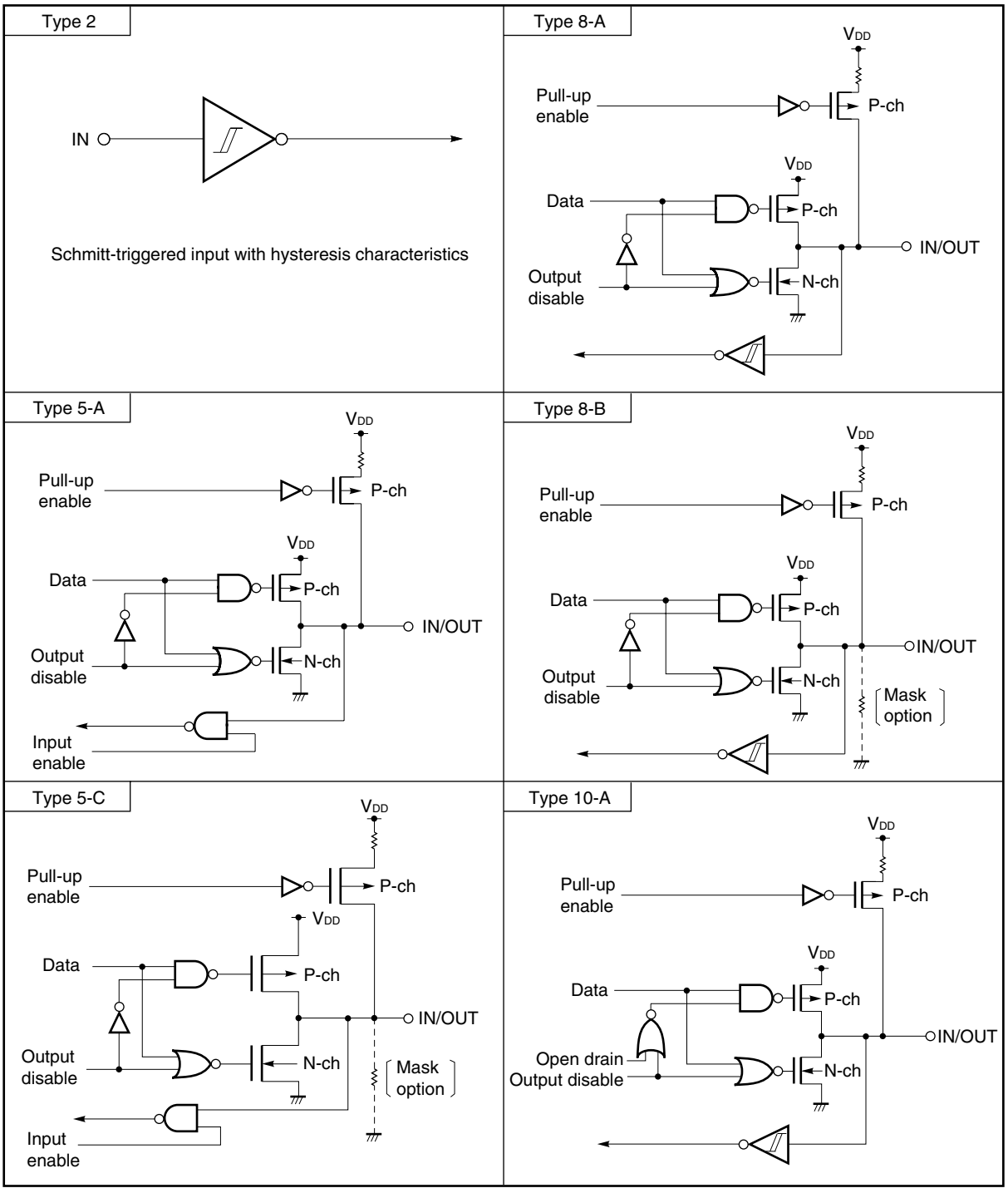


Figure 2-1. Pin I/O Circuits (2/3)

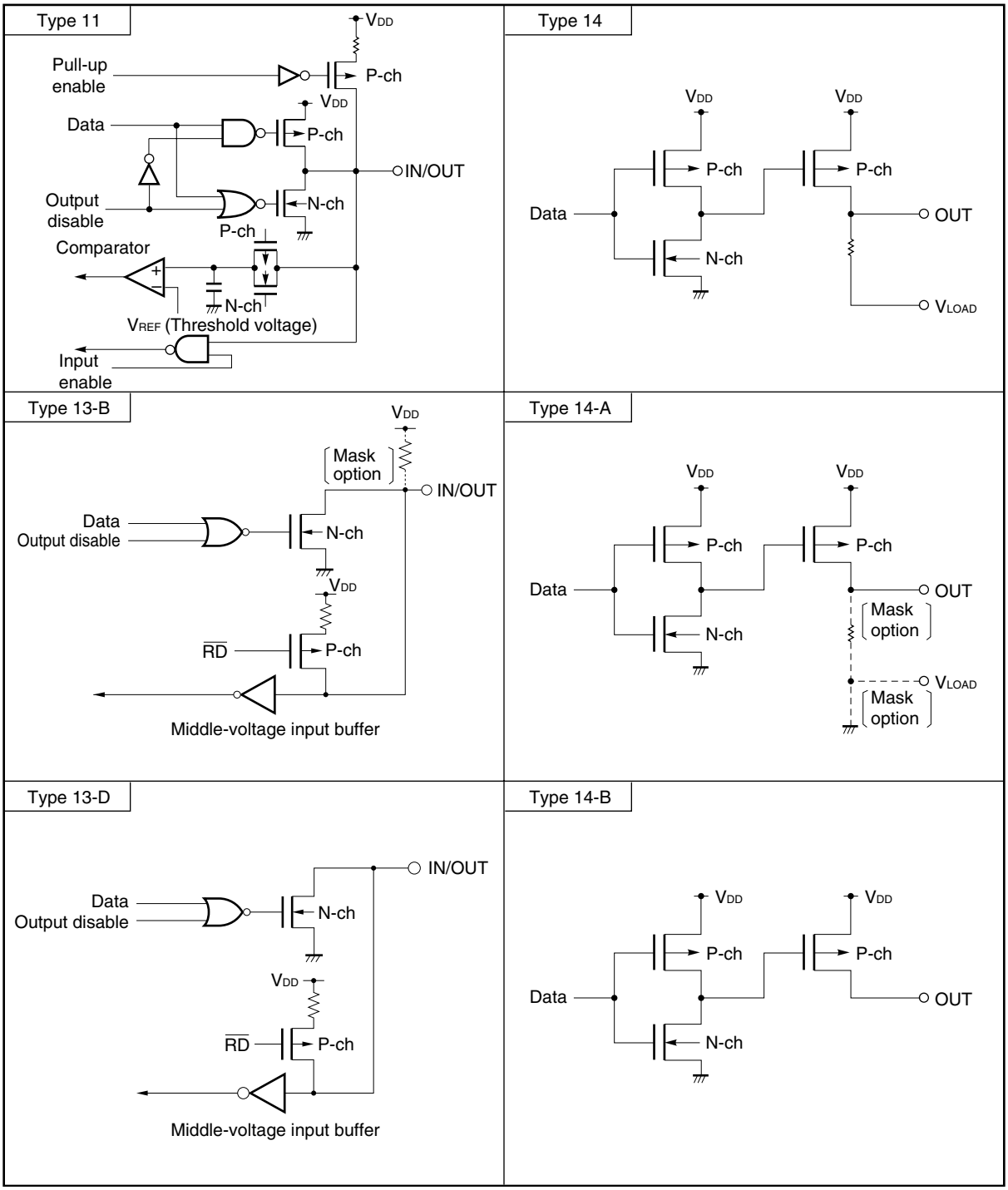
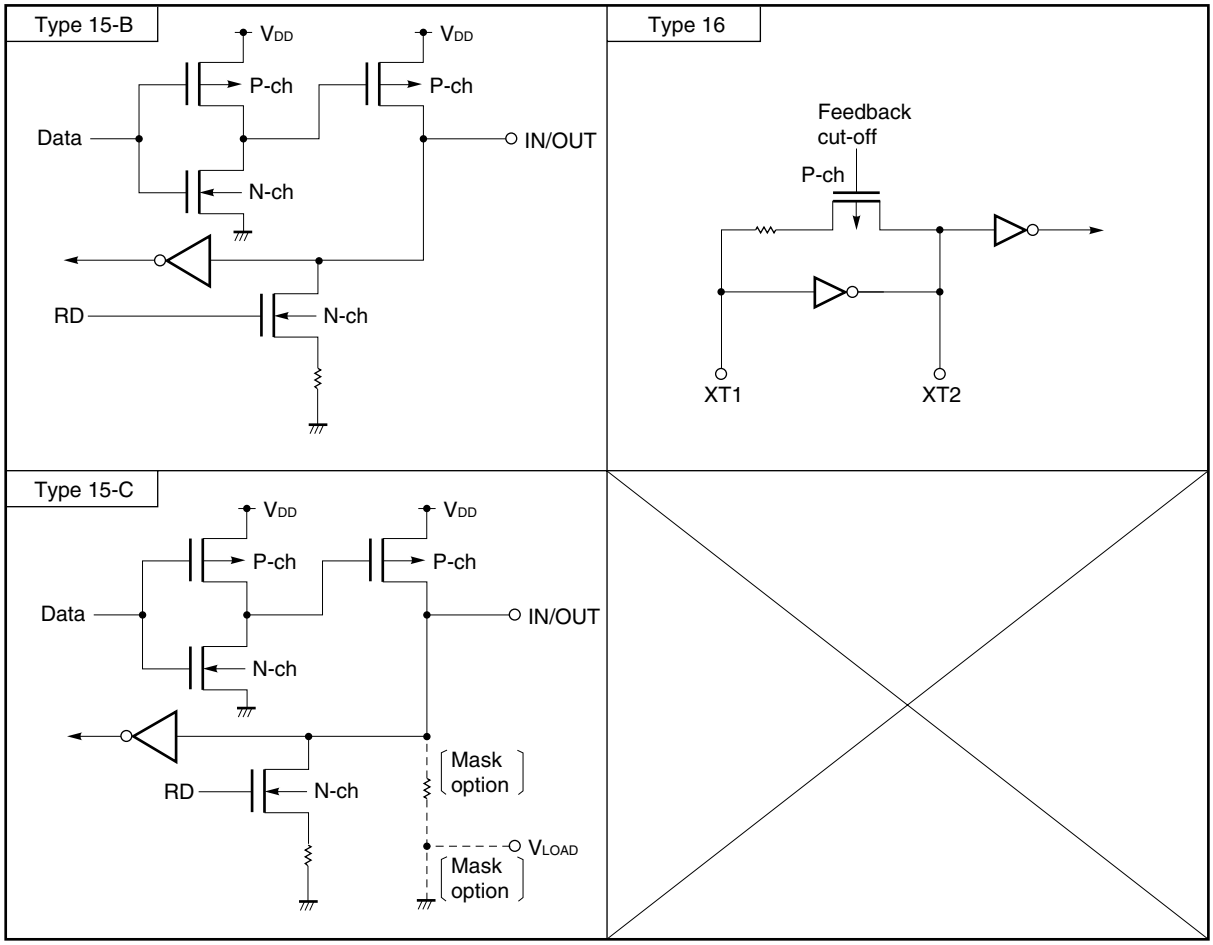


Figure 2-1. Pin I/O Circuits (3/3)



CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Each product of the μPD780208 Subseries accesses a memory space of 64 KB. Figures 3-1 to 3-5 show memory maps.

★ **Caution The initial values of the internal memory size switching register (IMS) in the μPD780204A, 780205A, and 78P0208 are fixed to CFH, regardless of the internal memory capacity. Therefore, set the values shown below for each product before use.**
μPD780204A: C8H
μPD780205A: CAH
μPD78P0208: Value corresponding to mask ROM version

Figure 3-1. Memory Map (μPD780204 and μPD780204A)

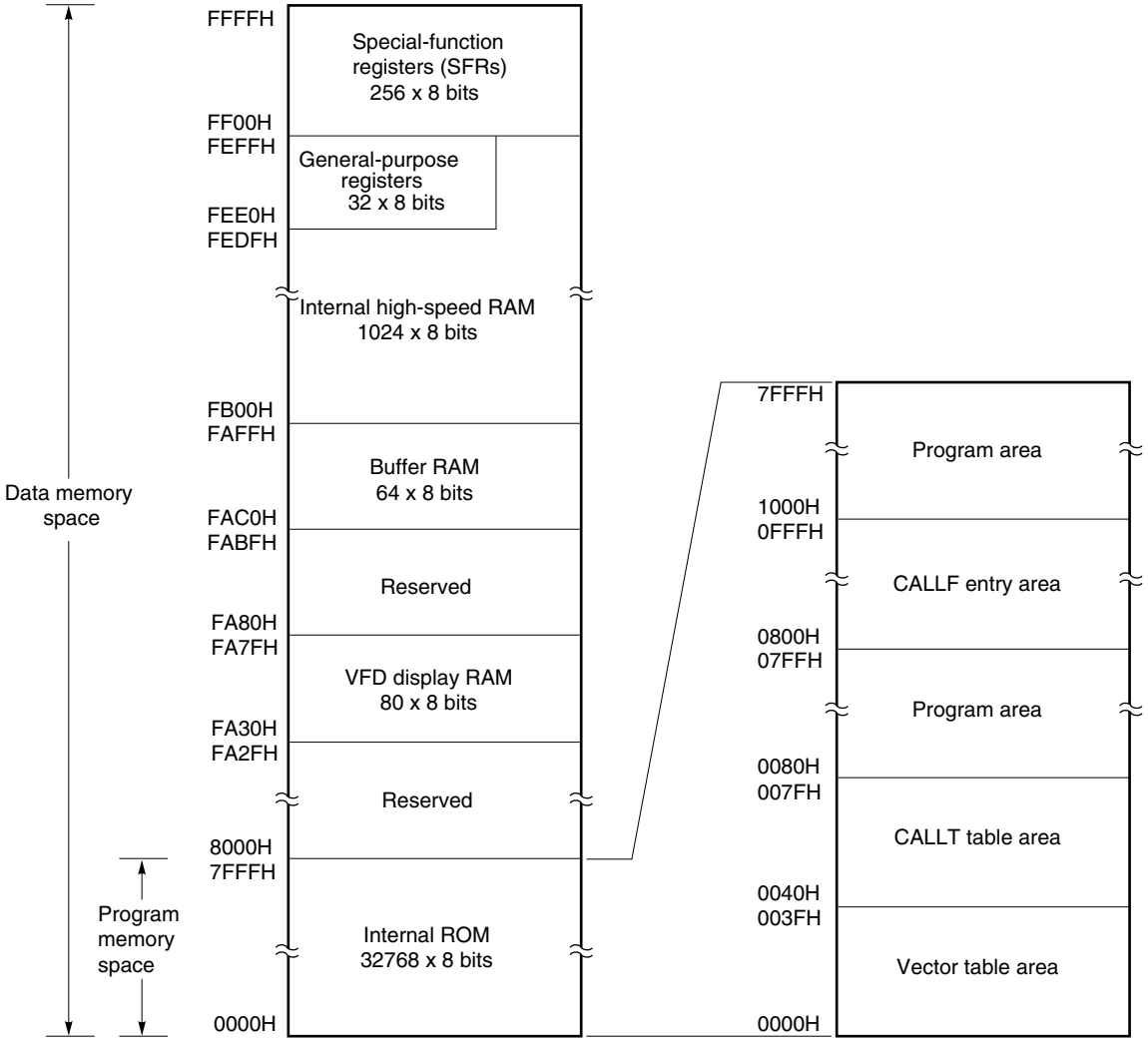


Figure 3-2. Memory Map (μPD780205 and μPD780205A)

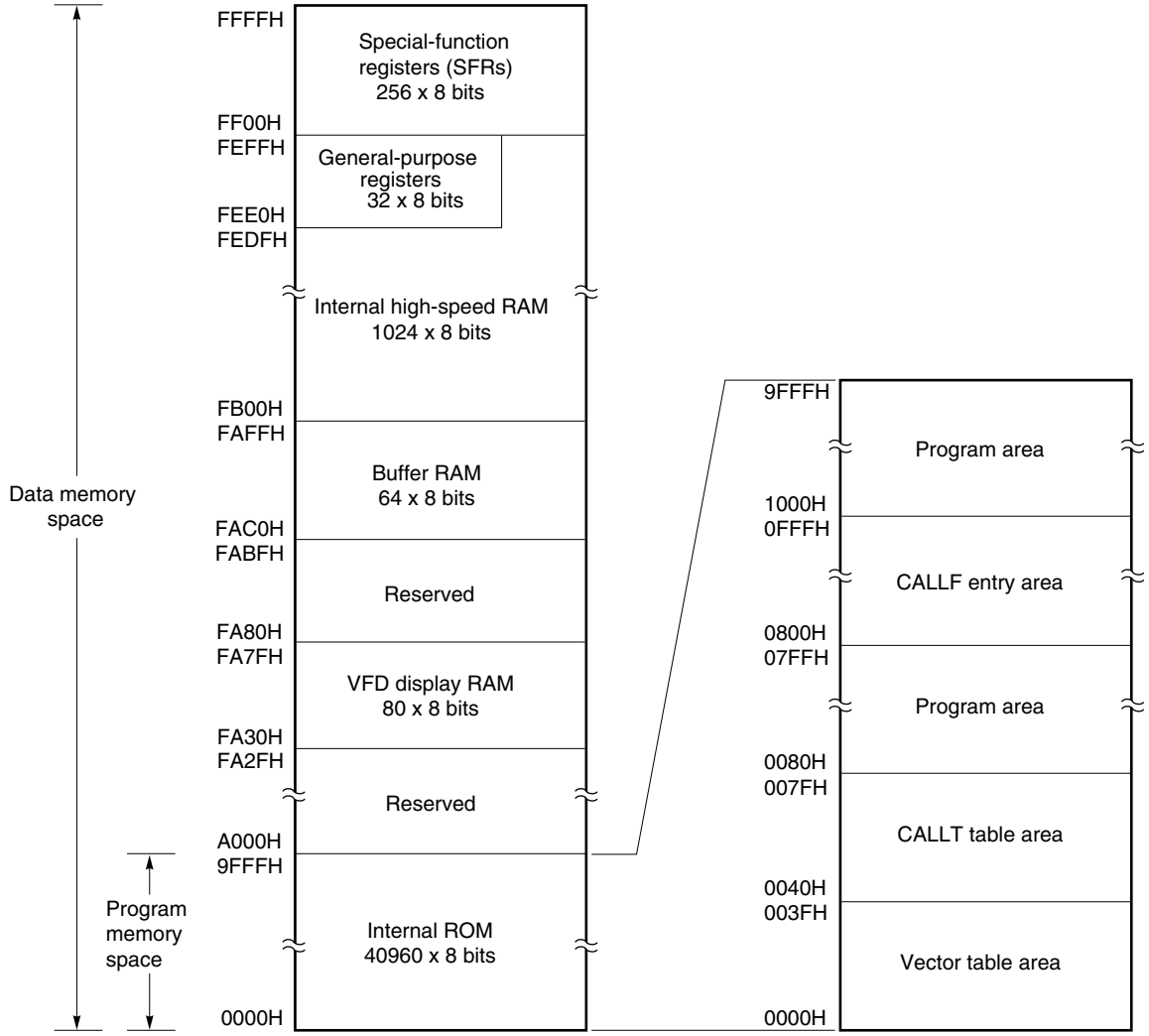


Figure 3-3. Memory Map (μPD780206)

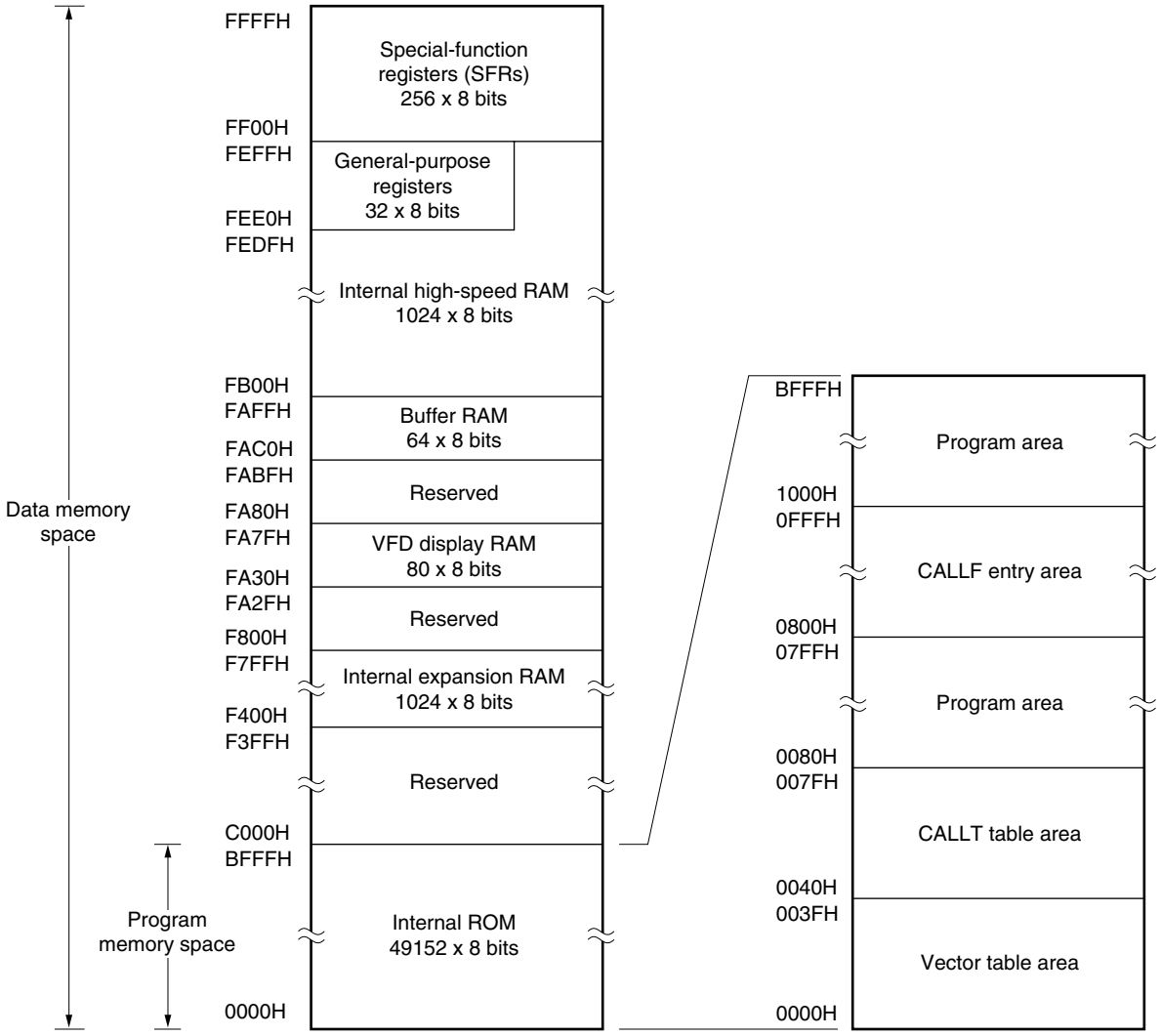


Figure 3-4. Memory Map (μPD780208)

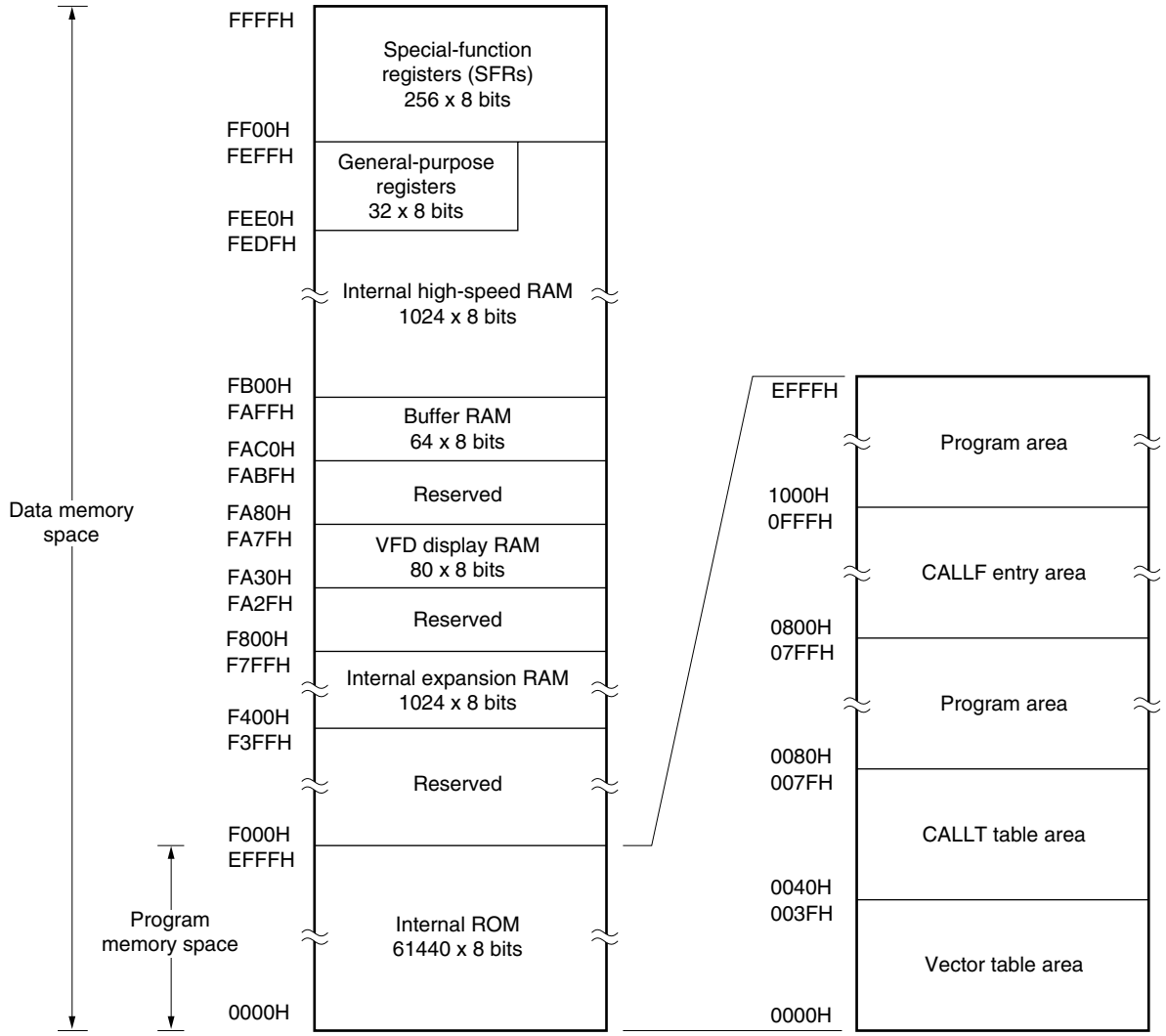
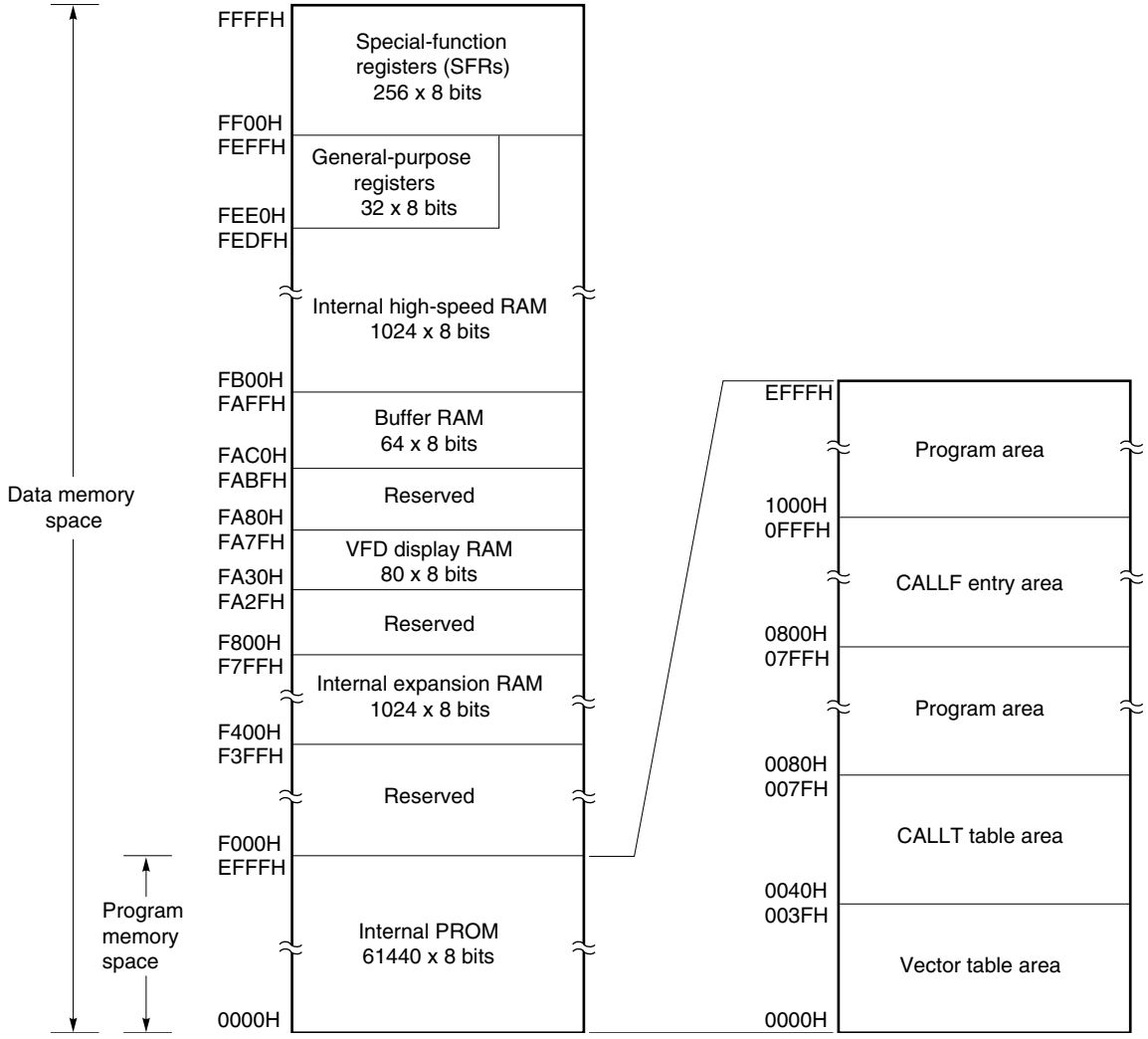


Figure 3-5. Memory Map (μPD78P0208)



3.1.1 Internal program memory space

The internal program memory space stores programs and table data. Normally, this space is addressed using the program counter (PC).

Each product in the μPD780208 Subseries contains internal ROM (or PROM) with the capacity shown below.

Table 3-1. Internal ROM Capacity

Part Number	Internal ROM	
	Configuration	Capacity
μPD780204 μPD780204A	Mask ROM	32768 x 8 bits
μPD780205 μPD780205A	Mask ROM	40960 x 8 bits
μPD780206	Mask ROM	49152 x 8 bits
μPD780208	Mask ROM	61440 x 8 bits
μPD78P0208	PROM	61440 x 8 bits

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

(1) Vector table area

The 64-byte area 0000H to 003FH is reserved as vector table area. Program start addresses for branch upon $\overline{\text{RESET}}$ input or interrupt request generation are stored in the vector table area. Of the 16-bit address, the lower 8 bits are stored at even addresses and the higher 8 bits are stored at odd addresses.

Table 3-2. Vector Table

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

(2) CALLT instruction table area

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

(3) CALLF instruction entry area

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

3.1.2 Internal data memory space

The μPD780208 Subseries units incorporate the following RAMs.

(1) Internal high-speed RAM

Internal high-speed RAM is allocated to the 1024-byte area from FB00H to FEFFH of the μPD780208 Subseries.

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

This area cannot be used as a program area in which instructions are written and executed.

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

(2) Internal expansion RAM

Internal expansion RAM is allocated to the 1024-byte area from F400H to F7FFH of the μPD780206, 780208, and 78P0208.

This area can also be used as a normal data area similar to the internal high-speed RAM, as well as a program area in which instructions can be written and executed.

The internal expansion RAM cannot be used as a stack memory.

(3) Buffer RAM

Buffer RAM is allocated to the 64-byte area from FAC0H to FAFFH. 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, buffer RAM can be used as normal RAM.

(4) VFD display RAM

VFD display RAM is allocated to the 80-byte area from FA30H to FA7FH. VFD display RAM can also be used as normal RAM.

3.1.3 Special-function register (SFR) area

On-chip peripheral hardware special-function registers (SFRs) are allocated to the area FF00H to FFFFH (see **Table 3-3 Special-Function Register List** under **3.2.3 Special-function registers (SFRs)**).

Caution Do not access addresses where SFRs are not assigned.

3.1.4 Data memory addressing

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

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

To address the memory area to be manipulated when an instruction is executed, the μPD780208 Subseries has many addressing modes to improve the operability. Especially, in the areas to which the data memory is assigned (addresses FB00H to FFFFH), the special-function registers (SFRs) and general-purpose registers can be addressed in accordance with thier function.

Data memory addressing is shown in Figures 3-6 to 3-10. For details of each addressing, refer to **3.4 Operand Address Addressing**.

Figure 3-6. Data Memory Addressing (μPD780204 and μPD780204A)

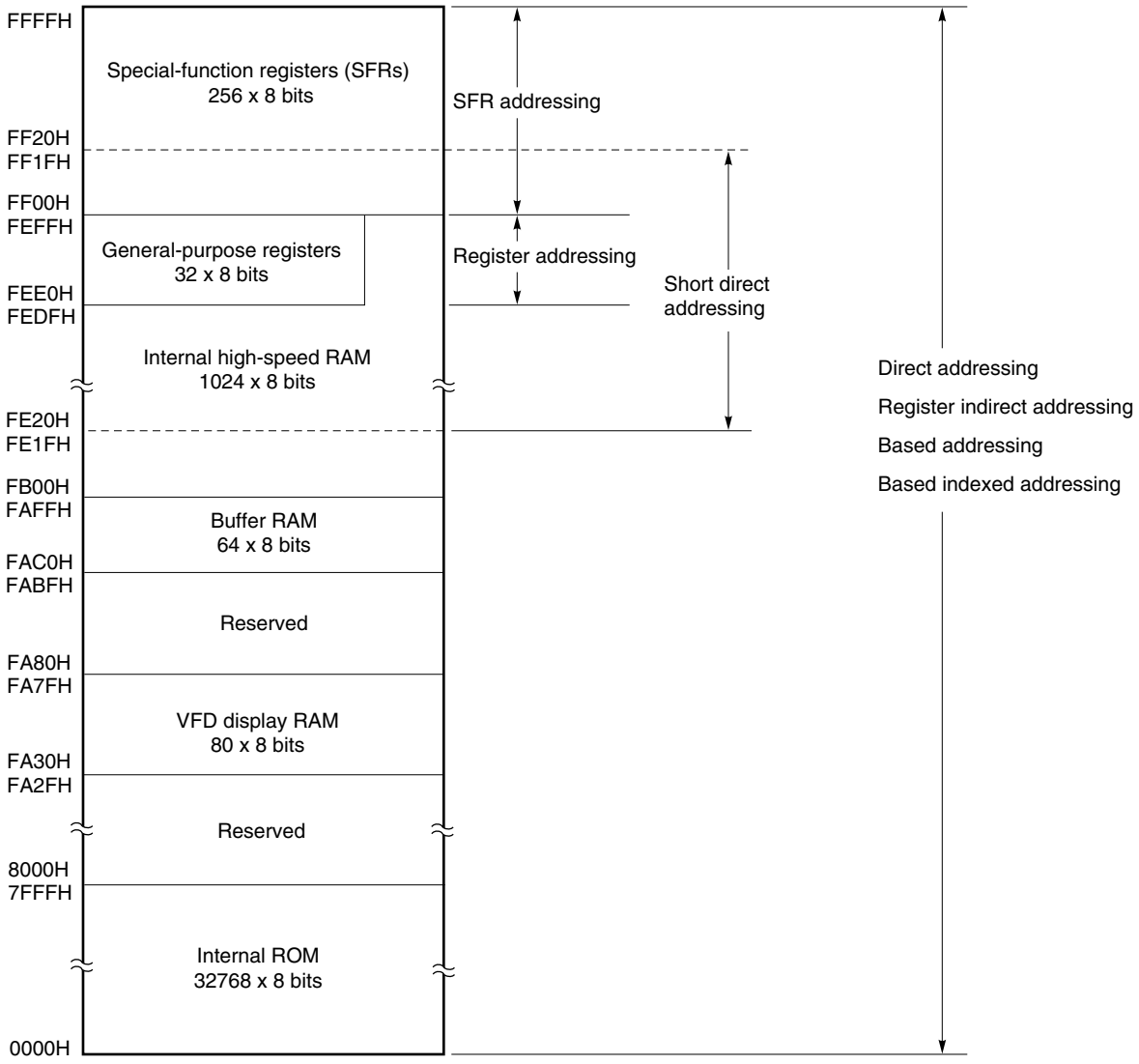


Figure 3-7. Data Memory Addressing (μPD780205 and μPD780205A)

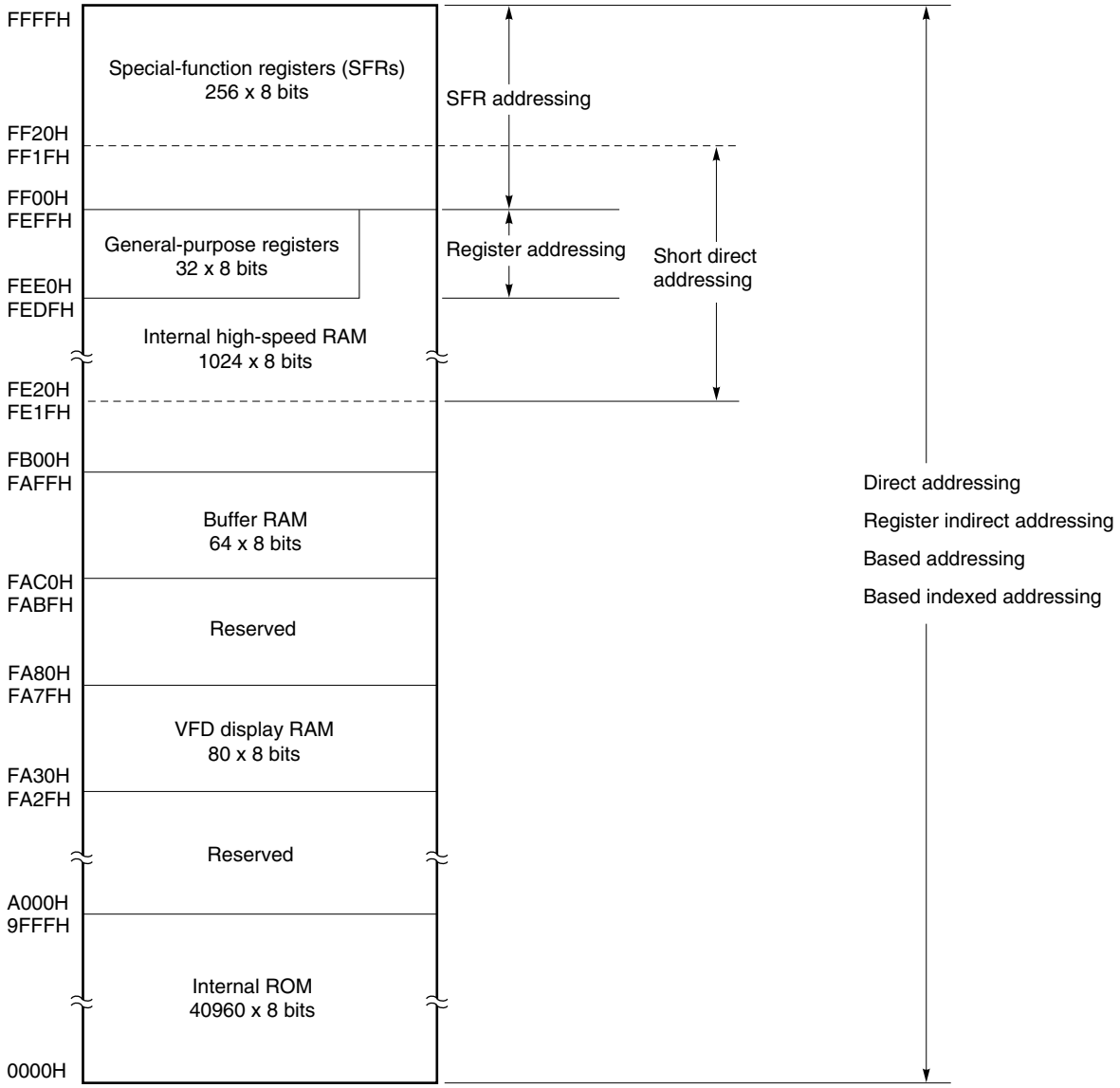


Figure 3-8. Data Memory Addressing (μPD780206)

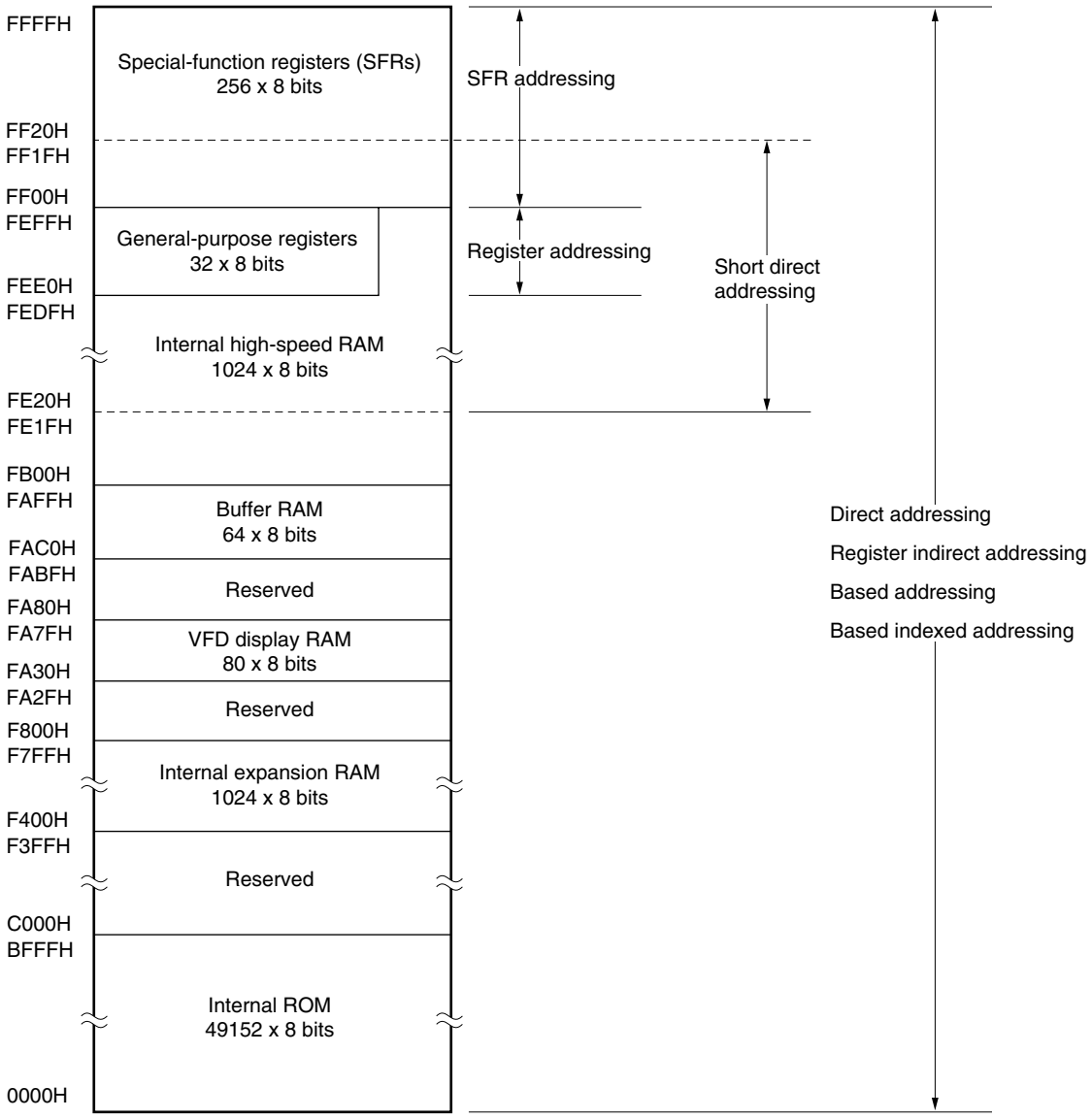


Figure 3-9. Data Memory Addressing (μPD780208)

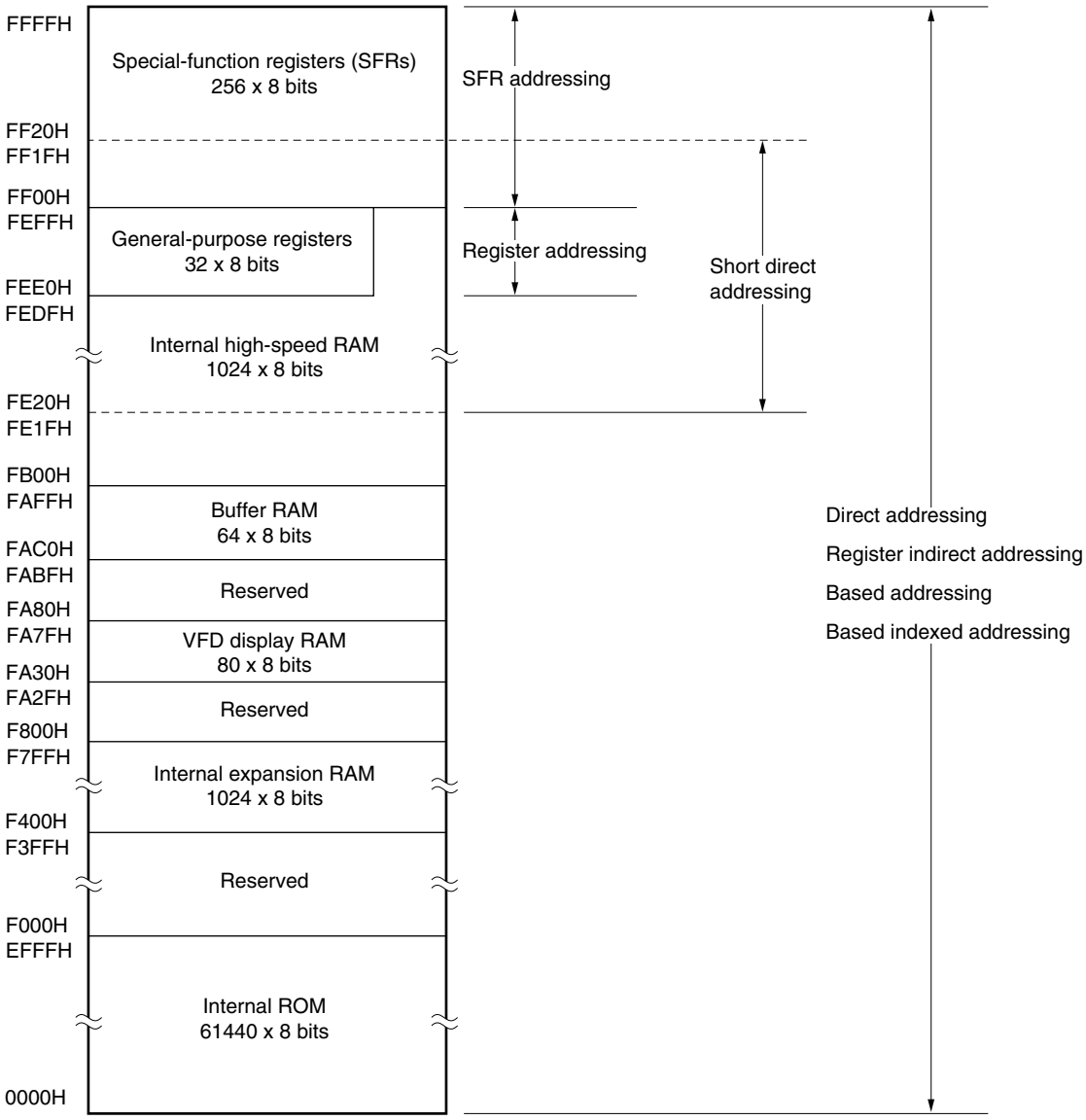
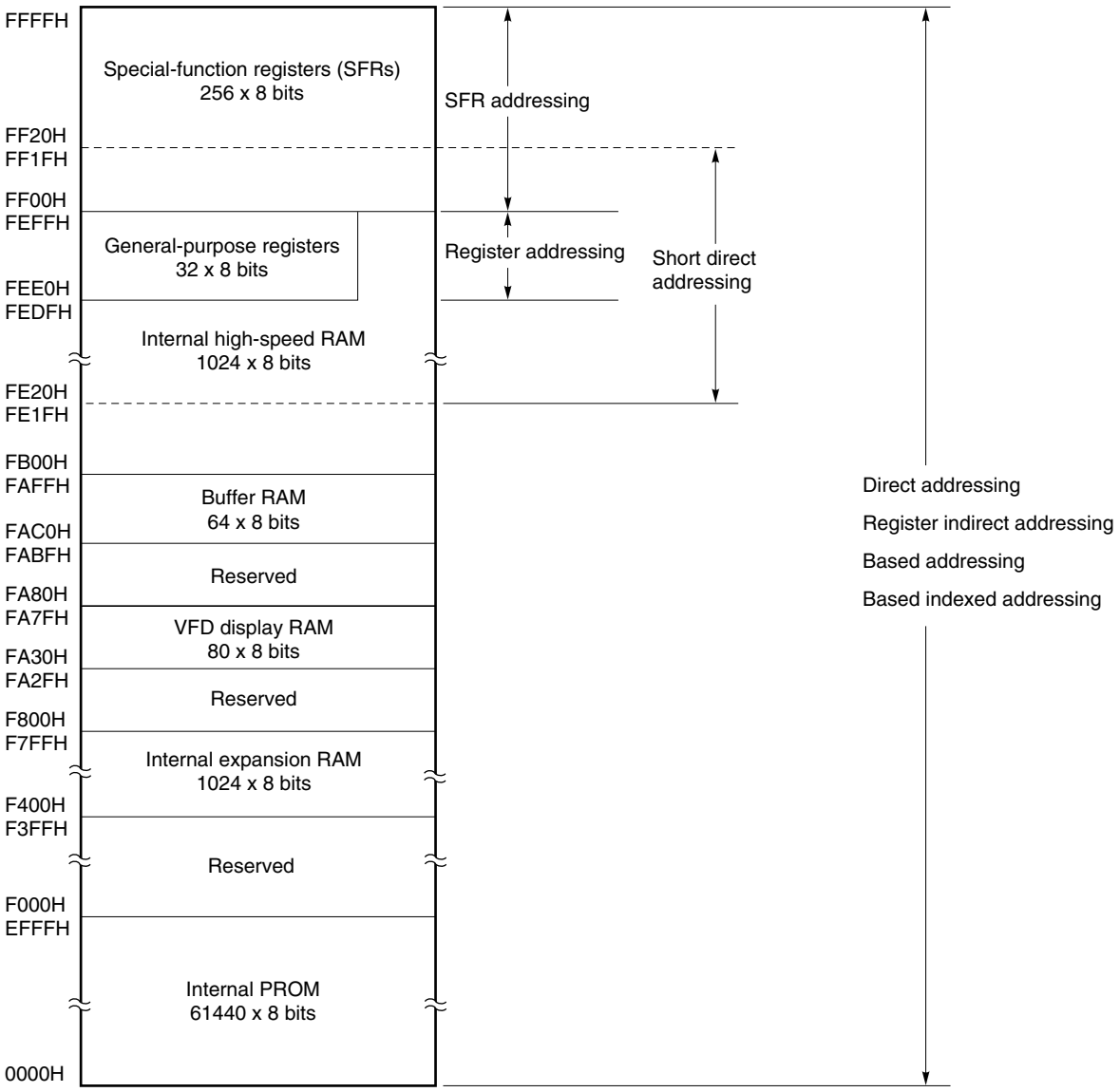


Figure 3-10. Data Memory Addressing (μPD78P0208)



3.2 Processor Registers

The μPD780208 Subseries units incorporate the following processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

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

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

Figure 3-11. Program Counter Format

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

(2) Program status word (PSW)

The program status word is an 8-bit register consisting of various flags to be set/reset by instruction execution. Program status word contents are automatically stacked upon interrupt request generation or PUSH PSW instruction execution and are automatically reset upon execution of the RETB, RETI, and POP PSW instructions.

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

Figure 3-12. Program Status Word Format

	7							0
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY

(a) Interrupt enable flag (IE)

This flag controls interrupt request acknowledgment operations of the CPU.

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

When IE = 1, the IE flag is set to the interrupt enabled (EI) status and interrupt request acknowledgment is controlled by an in-service priority flag (ISP), an interrupt mask flag for each interrupt source, and a priority specification flag.

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

(b) Zero flag (Z)

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

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

These are 2-bit flags used 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 a vectored interrupt request specified as lower priority by the priority specification flag registers (PR0L and PR0H) (refer to **16.3 (3) Priority specification flag registers (PR0L, PR0H)**) is disabled. Whether the interrupt request is actually acknowledged or not is controlled by 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 used to hold the start address of the memory stack area. Only the internal high-speed RAM area (FB00H to FEFFH) can be set as the stack area.

Figure 3-13. Stack Pointer Format

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

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

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

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

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

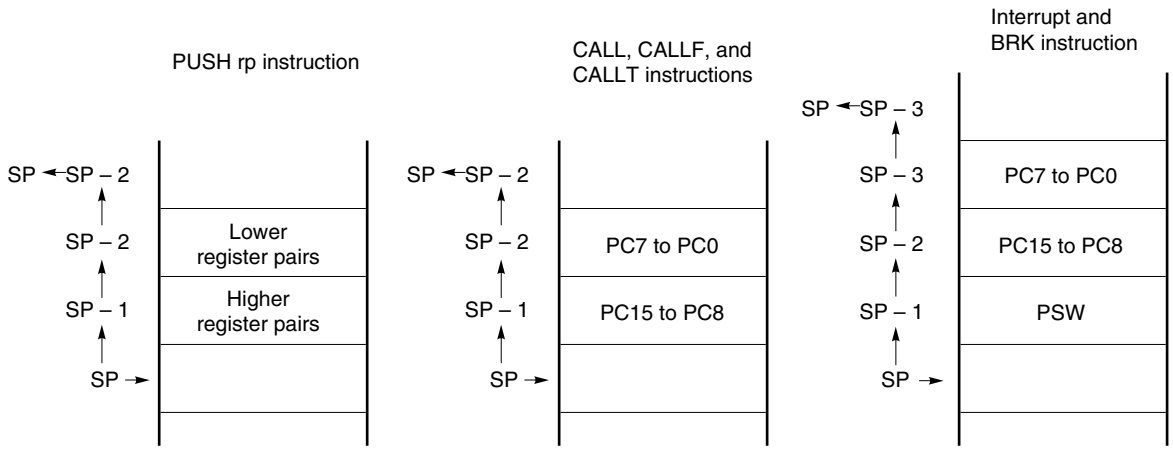
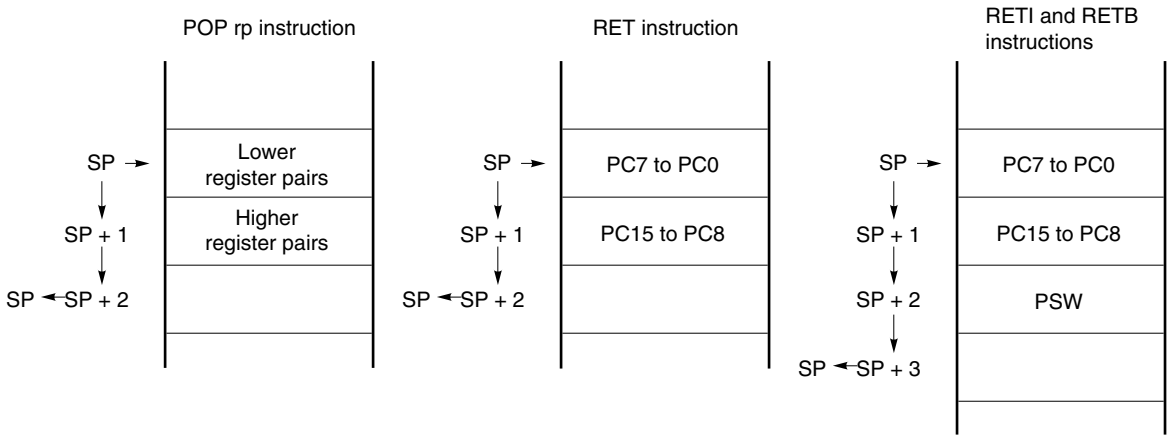


Figure 3-15. Data to Be Reset from Stack Memory



3.2.2 General-purpose registers

General-purpose registers are mapped at particular addresses (FEE0H to FEFFH) of the data memory. They consist of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

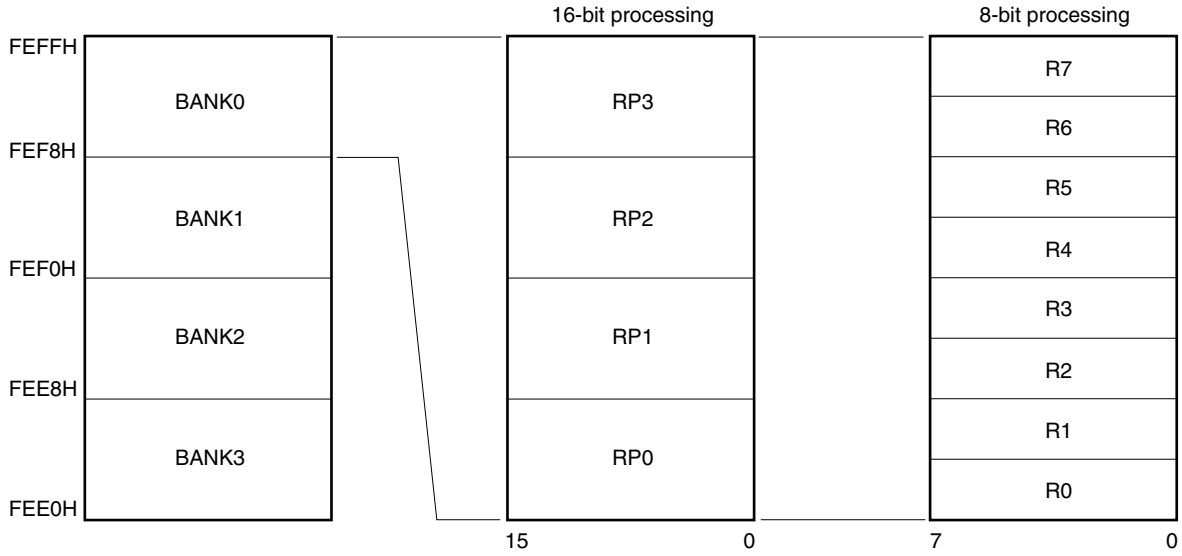
Each register can be used as an 8-bit register, and two 8-bit registers can be used in pairs as a 16-bit register (AX, BC, DE, and HL).

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).

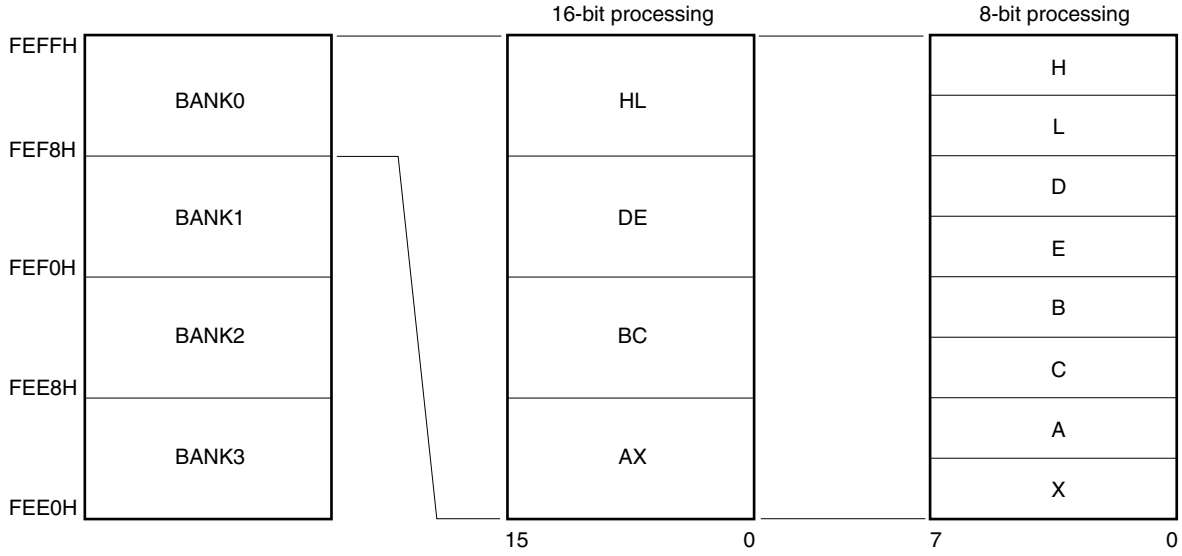
Register banks to be used for instruction execution are set using the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

Figure 3-16. General-Purpose Register Configuration

(a) Absolute name



(b) Function name



3.2.3 Special-function registers (SFRs)

Unlike a general-purpose register, each special-function register has a special function. The special-function registers are allocated in the FF00H to FFFFH area.

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

Each manipulation bit unit can be specified as follows.

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

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

- **Symbol**
  Indicates symbols that specify the addresses of the special-function registers. The RA78K0 uses these symbols as reserved words, and the CC78K0 defines them in the header file “sfrbit.h”. Symbols can be used as instruction operands if the RA78K0, ID78K0, or SD78K0 is used.

- **R/W**
  Indicates whether the corresponding special-function register can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only

- **Manipulatable bit units**
  √ indicates manipulatable bit units (1, 8, and 16). – indicates unmanipulatable bit units.

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

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

Address	Special-Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0	R/W	√	√	–	00H
FF01H	Port 1	P1		√	√	–	
FF02H	Port 2	P2		√	√	–	
FF03H	Port 3	P3		√	√	–	
FF07H	Port 7	P7		√	√	–	
FF08H	Port 8	P8	W	√	√	–	
FF09H	Port 9	P9		√	√	–	
FF0AH	Port 10	P10	R/W	√	√	–	
FF0BH	Port 11	P11		√	√	–	
FF0CH	Port 12	P12		√	√	–	
FF10H FF11H	16-bit compare register	CR00		–	–	√	Undefined
FF12H FF13H	16-bit capture register	CR01	R	–	–	√	
FF14H FF15H	16-bit timer register	TM0		–	–	√	0000H
FF16H	8-bit compare register 10	CR10	R/W	–	√	–	Undefined
FF17H	8-bit compare register 20	CR20		–	√	–	
FF18H	8-bit timer register 1	TMS TM1	R	–	√	√	00H
FF19H	8-bit timer register 2	TM2		–	√		
FF1AH	Serial I/O shift register 0	SIO0	R/W	–	√	–	Undefined
FF1BH	Serial I/O shift register 1	SIO1		–	√	–	
FF1FH	A/D conversion result register	ADCR	R	–	√	–	
FF20H	Port mode register 0	PM0	R/W	√	√	–	1FH
FF21H	Port mode register 1	PM1		√	√	–	FFH
FF22H	Port mode register 2	PM2		√	√	–	
FF23H	Port mode register 3	PM3		√	√	–	
FF27H	Port mode register 7	PM7		√	√	–	1FH
FF2AH	Port mode register 10	PM10		√	√	–	FFH
FF2BH	Port mode register 11	PM11		√	√	–	
FF2CH	Port mode register 12	PM12		√	√	–	
FF40H	Timer clock select register 0	TCL0	R/W	√	√	–	00H
FF41H	Timer clock select register 1	TCL1		–	√	–	
FF42H	Timer clock select register 2	TCL2		–	√	–	
FF43H	Timer clock select register 3	TCL3		–	√	–	88H
FF47H	Sampling clock select register	SCS		–	√	–	00H
FF48H	16-bit timer mode control register	TMC0		√	√	–	
FF49H	8-bit timer mode control register	TMC1		√	√	–	
FF4AH	Watch timer mode control register	TMC2		√	√	–	
FF4EH	16-bit timer output control register	TOC0		√	√	–	
FF4FH	8-bit timer output control register	TOC1		√	√	–	

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

Address	Special-Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FF60H	Serial operating mode register 0	CSIM0		R/W	√	√	–	00H
FF61H	Serial bus interface control register	SBIC			√	√	–	
FF62H	Slave address register	SVA			–	√	–	Undefined
FF63H	Interrupt timing specification register	SINT			√	√	–	00H
FF68H	Serial operating mode register 1	CSIM1			√	√	–	
FF69H	Automatic data transmit/receive control register	ADTC			√	√	–	
FF6AH	Automatic data transmit/receive address pointer	ADTP			–	√	–	
FF6BH	Automatic data transmit/receive interval specification register	ADTI			√	√	–	
FF80H	A/D converter mode register	ADM			√	√	–	01H
FF84H	A/D converter input select register	ADIS			–	√	–	00H
FFA0H	Display mode register 0	DSPM0			△Note	√	–	
FFA1H	Display mode register 1	DSPM1			–	√	–	
FFA2H	Display mode register 2	DSPM2			–	√	–	
FFE0H	Interrupt request flag register 0L	IF0	IF0L		√	√	√	
FFE1H	Interrupt request flag register 0H		IF0H		√	√		
FFE4H	Interrupt mask flag register 0L	MK0	MK0L		√	√	√	FFH
FFE5H	Interrupt mask flag register 0H		MK0H		√	√		
FFE8H	Priority order specification flag register 0L	PR0	PR0L		√	√	√	
FFE9H	Priority order specification flag register 0H		PR0H		√	√		
FFECH	External interrupt mode register	INTM0			–	√	–	00H

Note Only bit 7 can be manipulated, and only as a read operation.

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

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

★ **Note** The value after resetting the internal memory size switching register (IMS) and internal expansion RAM size switching register (IXS) depends on the product.

	μPD780204	μPD780204A	μPD780205	μPD780205A	μPD780206	μPD780208	μPD78P0208
IMS	C8H	CFH	CAH	CFH	CCH	CFH	CFH
IXS	None				0AH		

When using the μPD780204, 780205, 780206, and 780208, do not set any value other than that of IMS and IXS after reset.

When using the μPD780204A, 780205A, and 78P0208, the initial values of IMS are fixed to CFH, regardless of the internal memory capacity. Therefore, set the values shown below for each product before use.

μPD780204A: C8H

μPD780205A: CAH

μPD78P0208: Value corresponding to mask ROM version

3.3 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 the instruction to be fetched each time another instruction is executed. When a branch instruction is executed, the branch destination information is set to the PC and the program is branched by the following addressing (for details of instructions, refer to the **78K/0 Series Instructions User's Manual (U12326E)**).

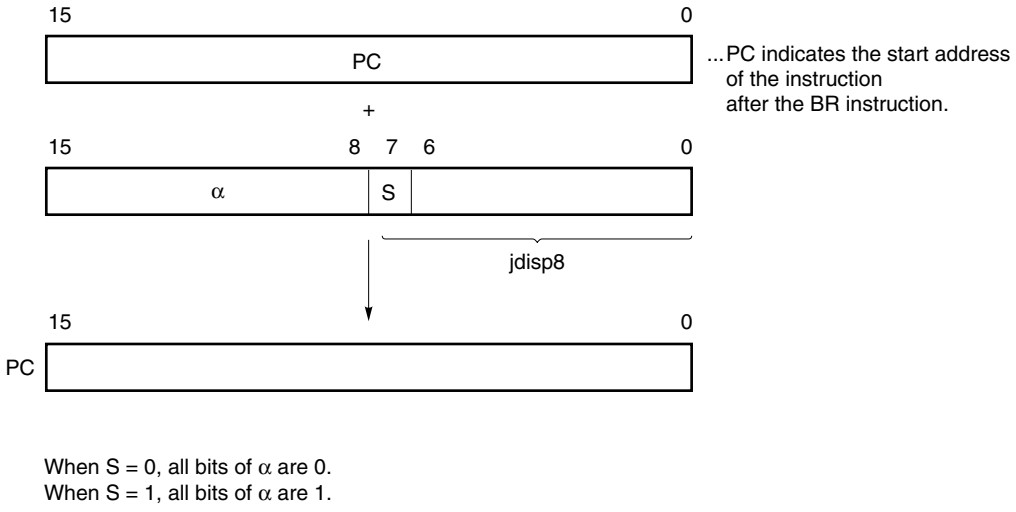
3.3.1 Relative addressing

[Function]

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

3.3.2 Immediate addressing

[Function]

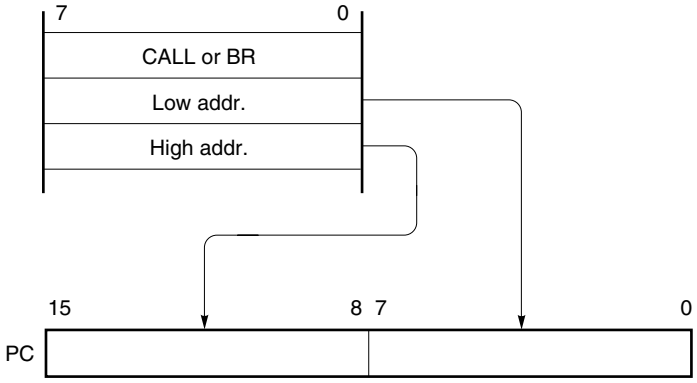
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

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

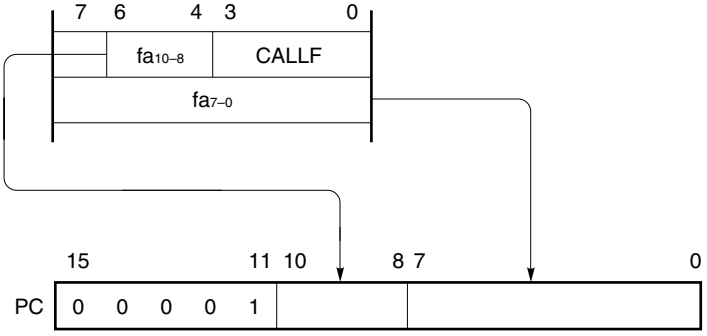
CALL !addr16 and BR !addr16 instructions can branch to all the memory spaces. CALLF !addr11 instruction branches to the area from 0800H to 0FFFH.

[Illustration]

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



In the case of CALLF !addr11 instruction



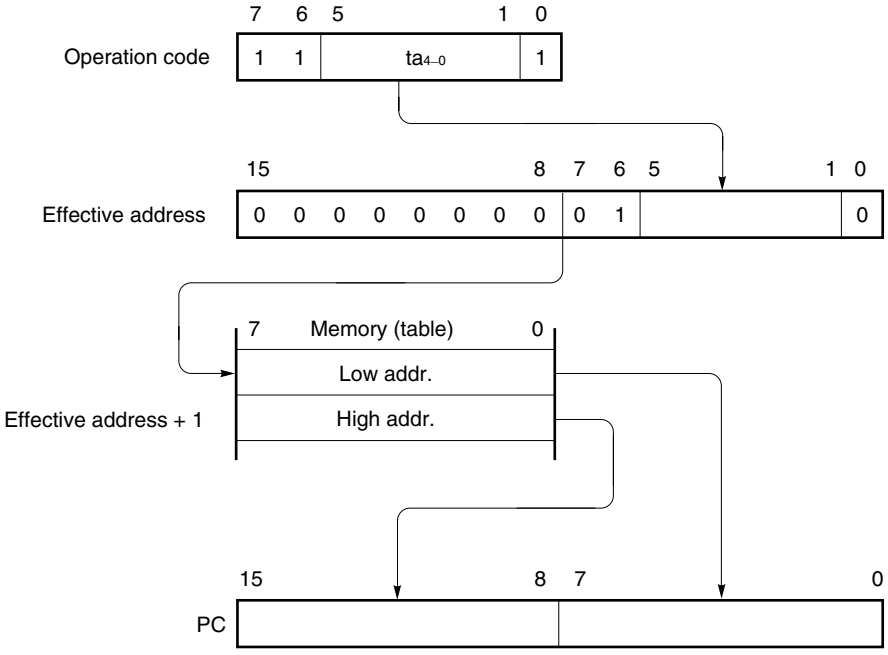
3.3.3 Table indirect addressing

[Function]

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

Table indirect addressing is 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 spaces.

[Illustration]



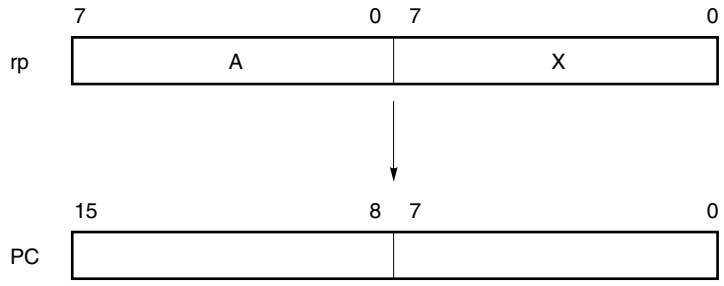
3.3.4 Register addressing

[Function]

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

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

[Illustration]



3.4 Operand Address Addressing

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

3.4.1 Implied addressing

[Function]

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

Of the μPD780208 Subseries instruction words, the following instructions employ implied addressing.

Instruction	Register to Be Specified by Implied Addressing
MULU	Register A for multiplicand and register AX for product storage
DIVUW	Register AX for dividend and quotient storage
ADJBA/ADJBS	Register A for storage of numeric values subject to decimal adjustment
ROR4/ROL4	Register A 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 register A and register X is stored in AX. In this example, the A and AX registers are specified by implied addressing.

3.4.2 Register addressing

[Function]

A general-purpose register is accessed as an operand. The general-purpose register to be accessed is specified by register bank select flags (RBS0 and RBS1) and the register specification code (Rn, RPn) in the operation code.

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

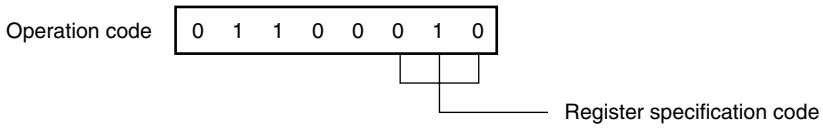
[Operand format]

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

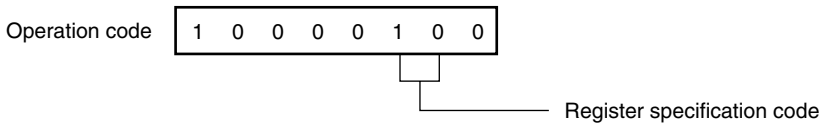
'r' and 'rp' can be described using function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) as well as absolute names (R0 to R7 and RP0 to RP3).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



3.4.3 Direct addressing

[Function]

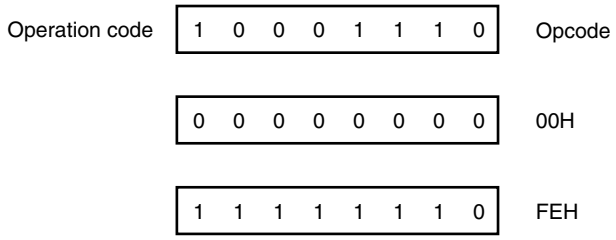
The memory indicated by immediate data in an instruction word is directly addressed.

[Operand format]

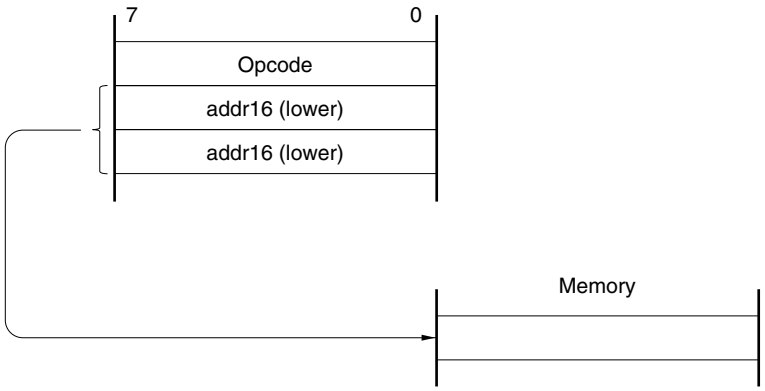
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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



[Illustration]



3.4.4 Short direct addressing

[Function]

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

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 special-function registers (SFRs) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.

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

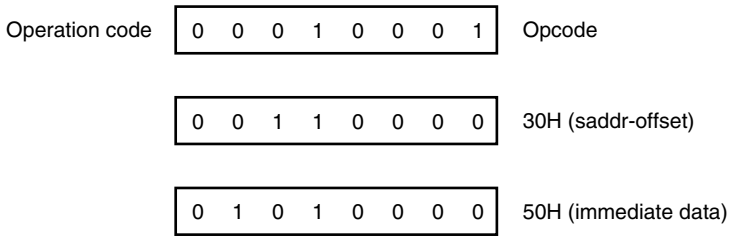
When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is set to 0. When it is at 00H to 1FH, bit 8 is set to 1. Refer to **[Illustration]** below.

[Operand format]

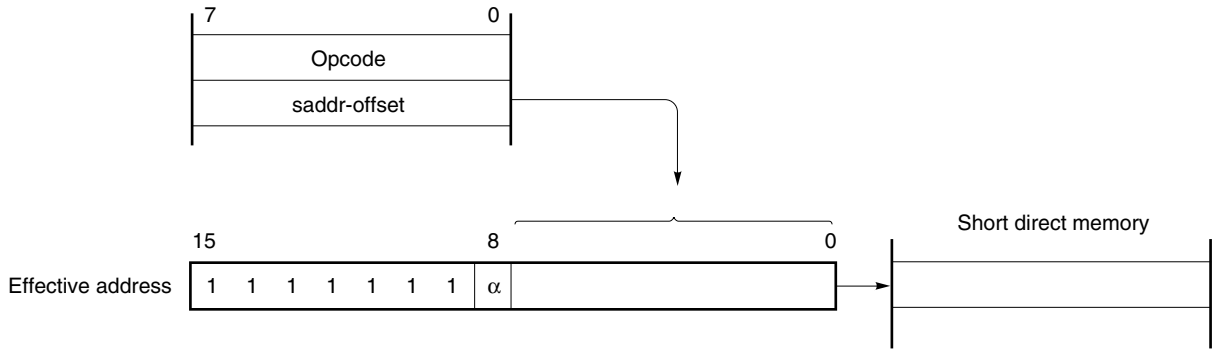
Identifier	Description
saddr	Label or immediate data indicating FE20H to FF1FH
saddrp	Label or immediate data indicating FE20H to FF1FH (even address only)

[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, $\alpha = 1$

3.4.5 Special-function register (SFR) addressing

[Function]

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

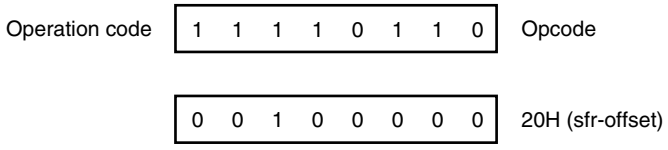
This addressing is applied to the 240-byte spaces FF00H to FFCFH and FFE0H to FFFFH. However, the SFRs mapped at FF00H to FF1FH can be accessed with short direct addressing.

[Operand format]

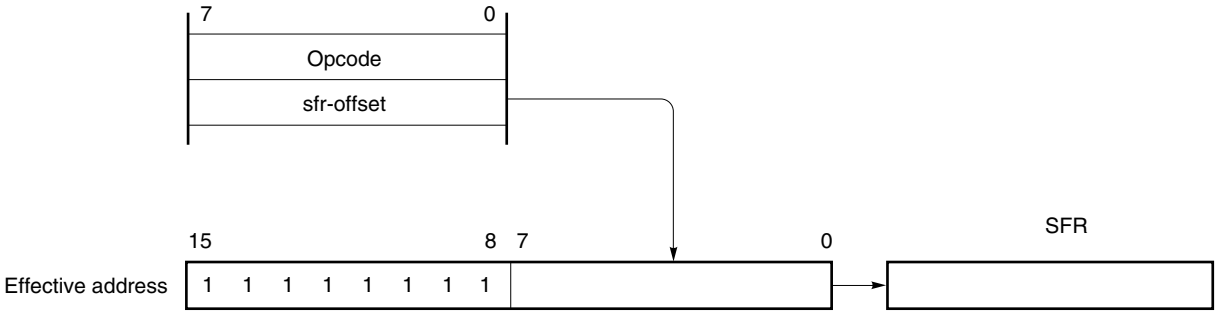
Identifier	Description
sfr	Special-function register name
sfrp	16-bit manipulatable special-function register name (even address only)

[Description example]

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



[Illustration]



3.4.6 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 specification code in the instruction code. This addressing can be carried out for all the memory spaces.

[Operand format]

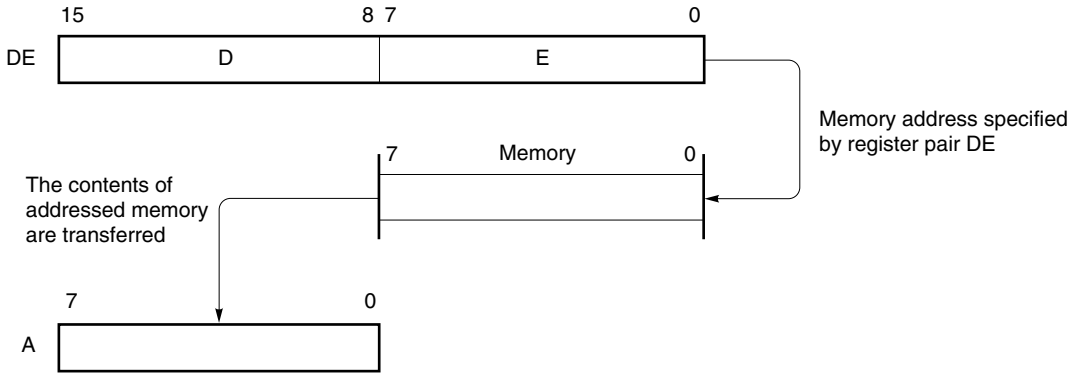
Identifier	Description
—	[DE], [HL]

[Description example]

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

Operation code	1	0	0	0	0	1	0	1

[Illustration]



3.4.7 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

3.4.8 Based indexed addressing

[Function]

The B or C register contents specified in an instruction word 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] (select B register)

Operation code	1	0	1	0	1	0	1	1

3.4.9 Stack addressing

[Function]

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

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

Stack addressing can be used to address the internal high-speed RAM area only.

[Description example]

In the case of PUSH DE (save DE register)

Operation code	1	0	1	1	0	1	0	1

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The μPD780208 Subseries units incorporate two input ports, 16 output ports, and 56 I/O ports. Figure 4-1 shows the port configuration. Every port is capable of 1-bit and 8-bit manipulations and can carry out various control operations. Besides port functions, the ports can also serve as on-chip hardware I/O pins.

Figure 4-1. Port Types

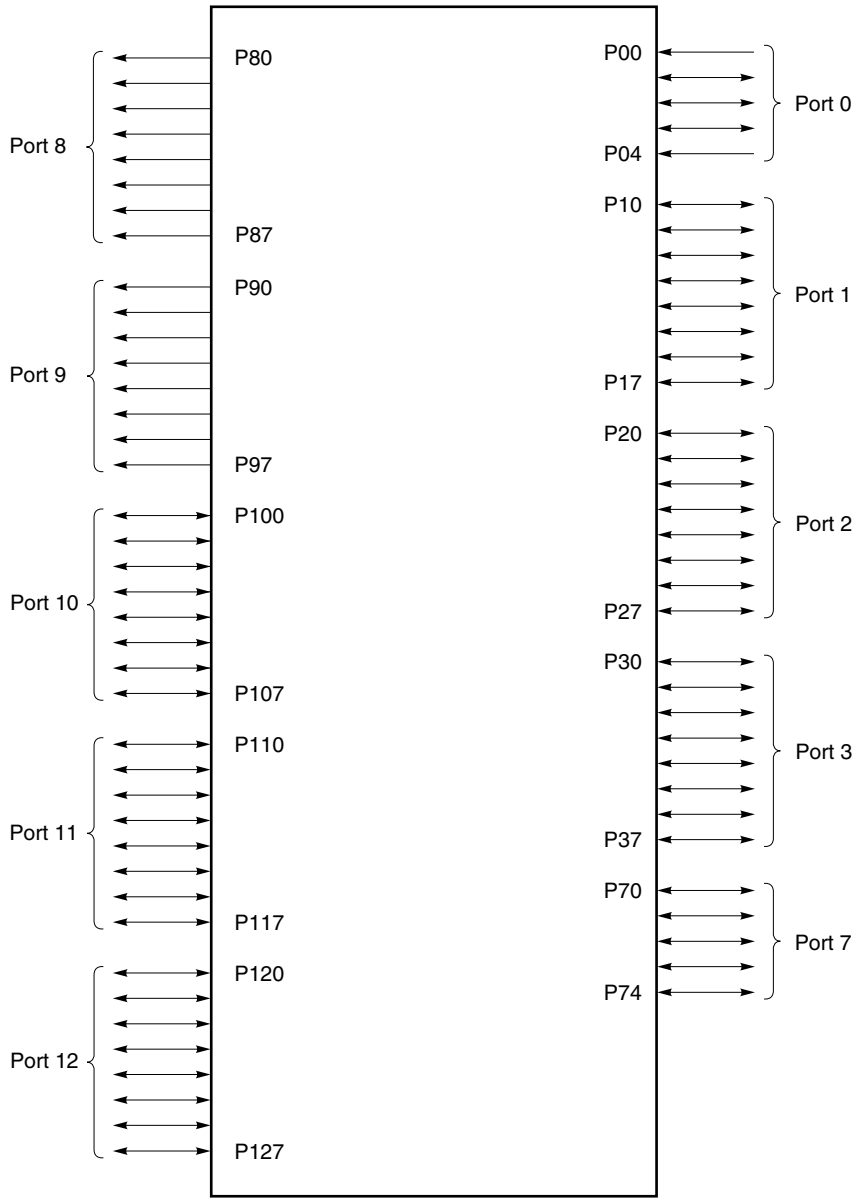


Table 4-1. Port Functions (1/2)

Pin Name	Function		Alternate Function
P00	Port 0. 5-bit I/O port.	Input only.	INTP0/TI0
P01		Input/output can be specified in 1-bit units. If used as an input port, on-chip pull-up resistors can be used by software settings.	INTP1
P02			INTP2
P03			INTP3
P04		Input only.	XT1
P10 to P17	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. If used as an input port, on-chip pull-up resistors can be used by software settings.		ANI0 to ANI7
P20	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units. If used as an input port, on-chip pull-up resistors can be used by software settings.		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 I/O port. Input/output can be specified in 1-bit units. LEDs can be driven directly. If used as an input port, on-chip pull-up resistors can be used by software settings. In mask ROM versions, use of pull-down resistors can be specified in 1-bit units with the mask option.		TO0
P31			TO1
P32			TO2
P33			TI1
P34			TI2
P35			PCL
P36			BUZ
P37			–
P70 to P74	Port 7. N-ch open-drain 5-bit I/O port. Input/output can be specified in 1-bit units. LEDs can be driven directly. In mask ROM versions, use of pull-up resistors can be specified in 1-bit units with the mask option.		–
P80 to P87	Port 8. P-ch open-drain 8-bit high withstanding voltage output port. LEDs can be driven directly. In mask ROM versions, use of pull-down resistors can be specified in 1-bit units with the mask option (can be specified as connected to V_{LOAD} or V_{SS} in 4-bit units).		FIP13 to FIP20
P90 to P97	Port 9. P-ch open-drain 8-bit high withstanding voltage output port. LEDs can be driven directly. In mask ROM versions, use of pull-down resistors can be specified in 1-bit units with the mask option (can be specified as connected to V_{LOAD} or V_{SS} in 4-bit units).		FIP21 to FIP28

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

Pin Name	Function	Alternate Function
P100 to P107	Port 10. P-ch open-drain 8-bit high withstanding voltage I/O port. Input/output can be specified in 1-bit units. LEDs can be driven directly. In mask ROM versions, use of pull-down resistors can be specified in 1-bit units with the mask option (can be specified as connected to V_{LOAD} or V_{SS} in 4-bit units).	FIP29 to FIP36
P110 to P117	Port 11. P-ch open-drain 8-bit high withstanding voltage I/O port. Input/output can be specified in 1-bit units. LEDs can be driven directly. In mask ROM versions, use of pull-down resistors can be specified in 1-bit units with the mask option (can be specified as connected to V_{LOAD} or V_{SS} in 4-bit units).	FIP37 to FIP44
P120 to P127	Port 12. P-ch open-drain 8-bit high withstanding voltage I/O port. Input/output can be specified in 1-bit units. LEDs can be driven directly. In mask ROM versions, use of pull-down resistors can be specified in 1-bit units with the mask option (can be specified as connected to V_{LOAD} or V_{SS} in 4-bit units).	FIP45 to FIP52

4.2 Port Configuration

A port consists of the following hardware.

Table 4-2. Port Configuration

Item	Configuration
Control registers	Port mode register (PMm: m = 0, 1, 2, 3, 7, 10, 11, 12) Pull-up resistor option register (PUO)
Ports	Total: 74 (2 input, 16 output, 56 I/O)
Pull-up resistors	• Mask ROM versions Total: 32 (software control: 27, mask option control: 5) • μPD78P0208 ... Total: 27
Pull-down resistors	• Mask ROM versions ... Total: 48 (mask option control: 48)

4.2.1 Port 0

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

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

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

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

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

Figure 4-2. Block Diagram of P00 and P04

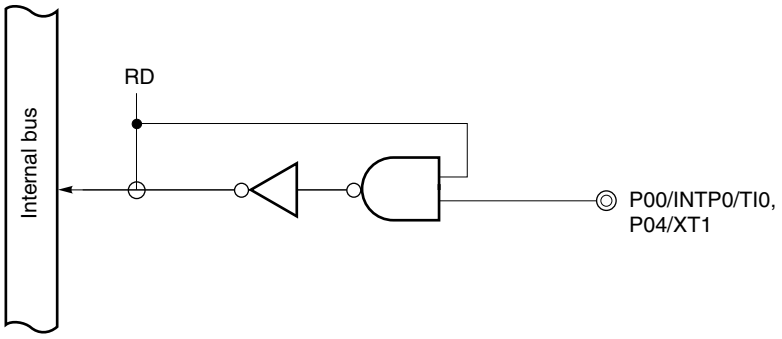
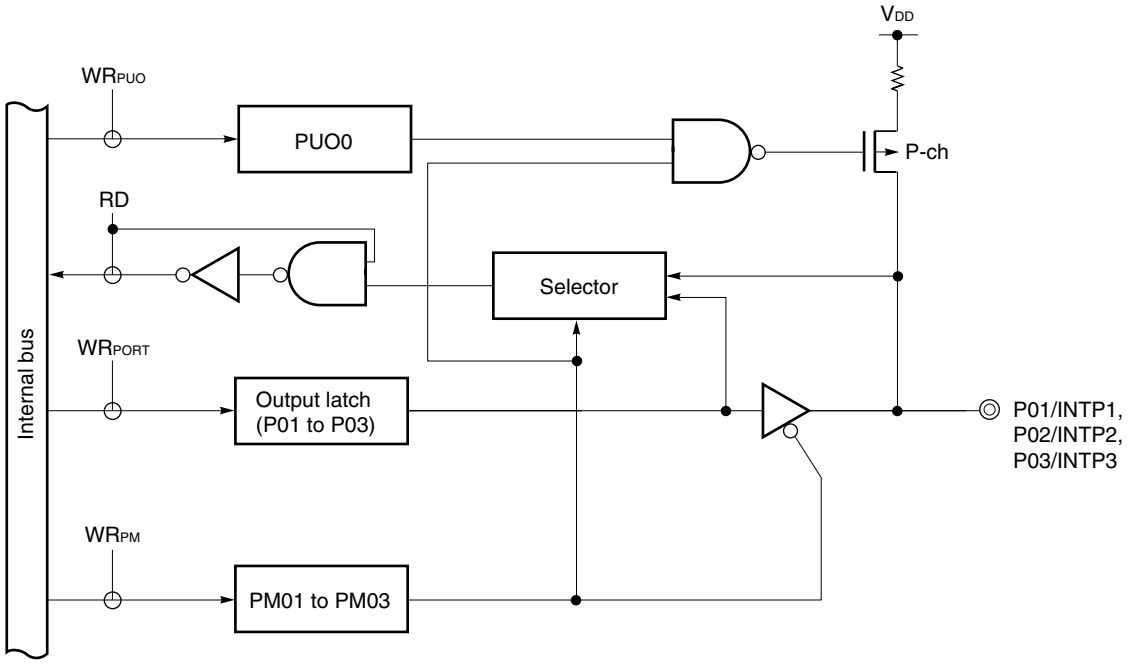


Figure 4-3. Block Diagram of P01 to P03



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

4.2.2 Port 1

Port 1 is an 8-bit I/O port with an output latch. The P10 to P17 pins can be set to input mode/output mode in 1-bit units using port mode register 1 (PM1). When the P10 to P17 pins are used as input port pins, on-chip pull-up resistors can be connected to them in 8-bit units using the pull-up resistor option register (PUO).

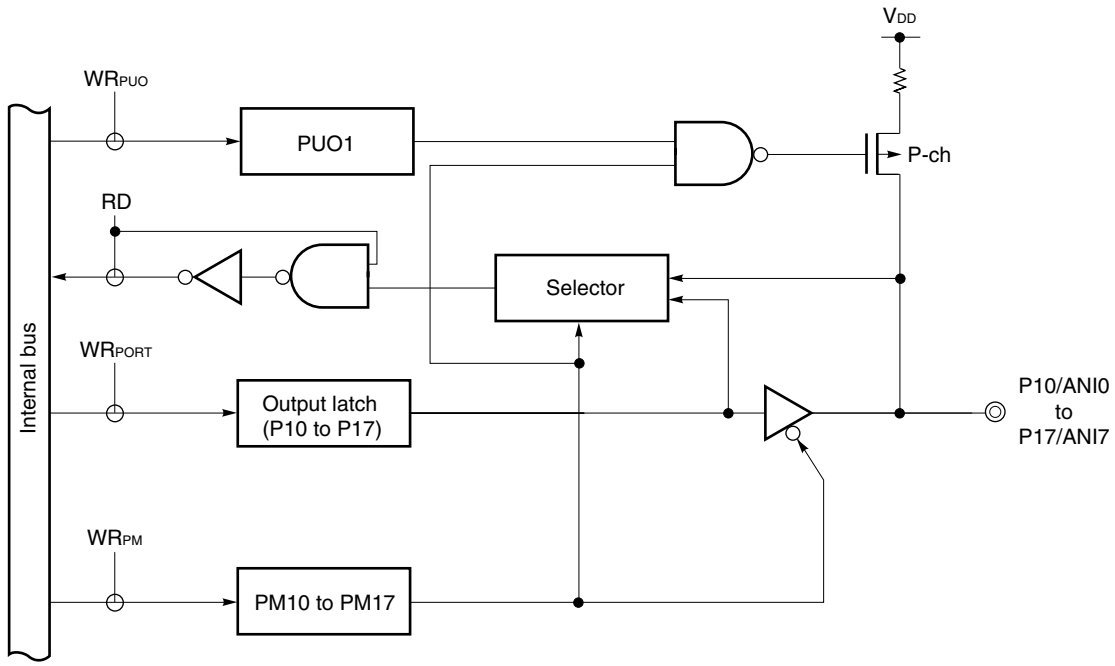
Alternate functions include A/D converter analog input.

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

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

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

Figure 4-4. 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

4.2.3 Port 2

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

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

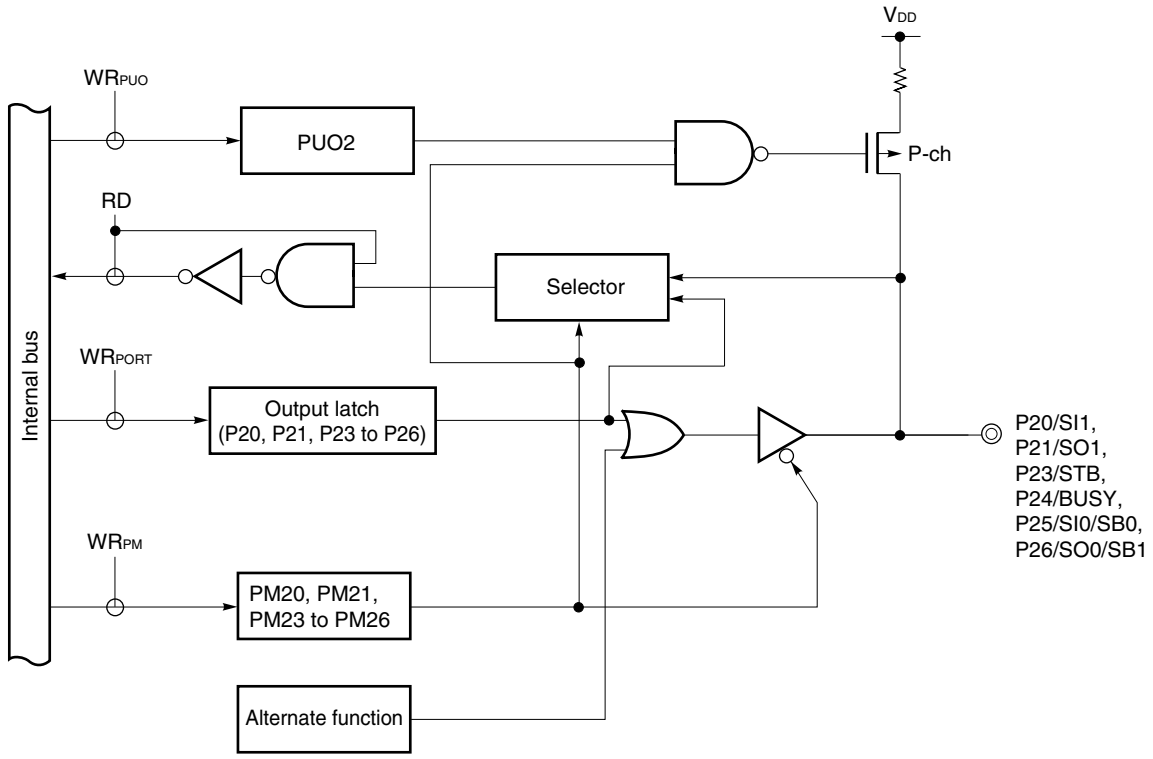
$\overline{\text{RESET}}$ input sets port 2 to input mode.

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

Cautions 1. If used as serial interface pins, set the I/O and output latch according to each function. Refer to Figure 13-3 Format of Serial Operating Mode Register 0 and Figure 14-3 Format of Serial Operating Mode Register 1 for the settings.

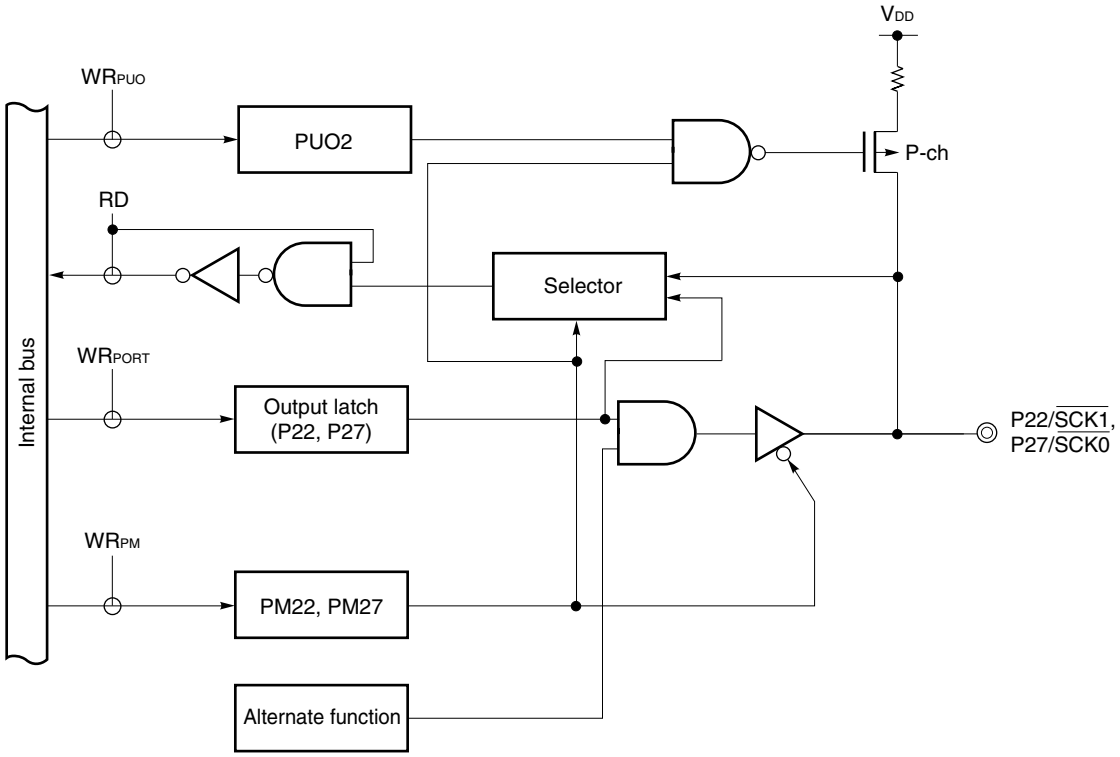
2. When reading the pin state in SBI mode, set the PM2n bit of PM2 to 1 (n = 5, 6) (refer to the description of (10) Judging busy status of slave in 13.4.3 SBI mode operation).

Figure 4-5. 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 4-6. 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

4.2.4 Port 3

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

In mask ROM versions, use of pull-down resistors can be specified in 1-bit units with the mask option. The μPD78P0208 does not contain pull-down resistors.

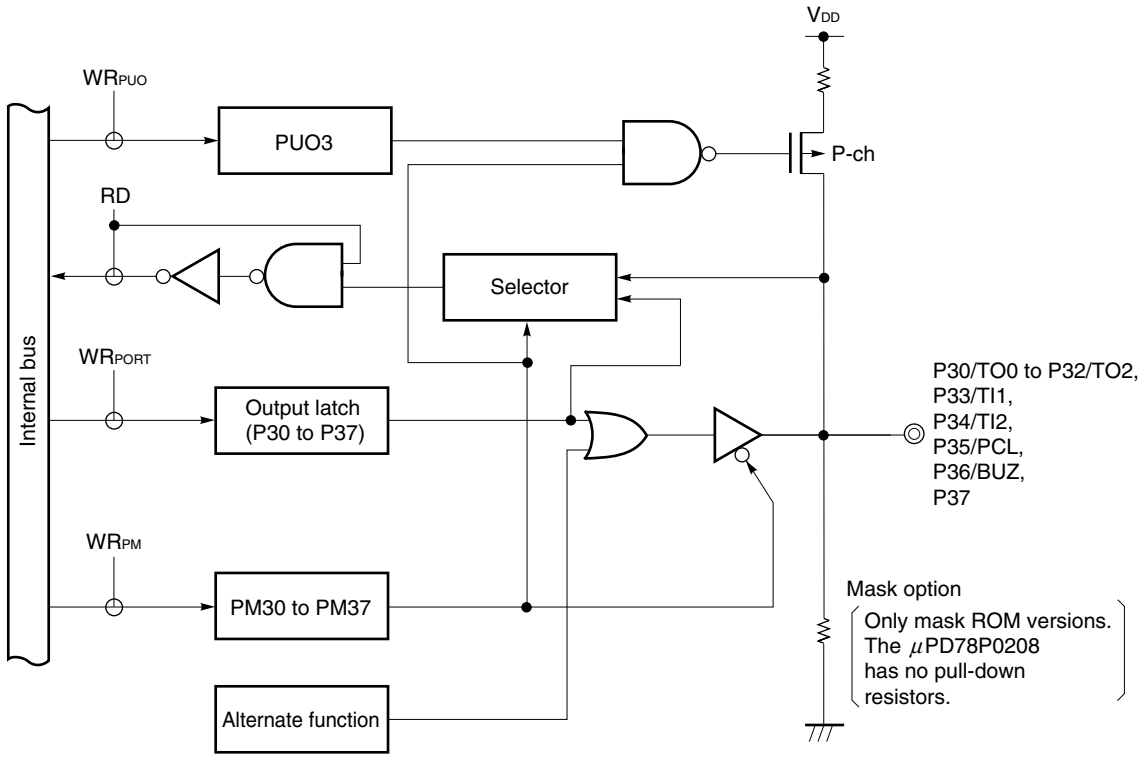
Port 3 can drive LEDs directly.

Alternate functions include timer I/O, clock output, and buzzer output.

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

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

Figure 4-7. 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

4.2.5 Port 7

Port 7 is a 5-bit I/O port with an output latch. The P70 to P74 pins can be set to input mode/output mode in 1-bit units using port mode register 7 (PM7). In mask ROM versions, use of pull-up resistors can be specified in 1-bit units with the mask option. The μPD78P0208 does not contain pull-up resistors.

Port 7 can drive LEDs directly.

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

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

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

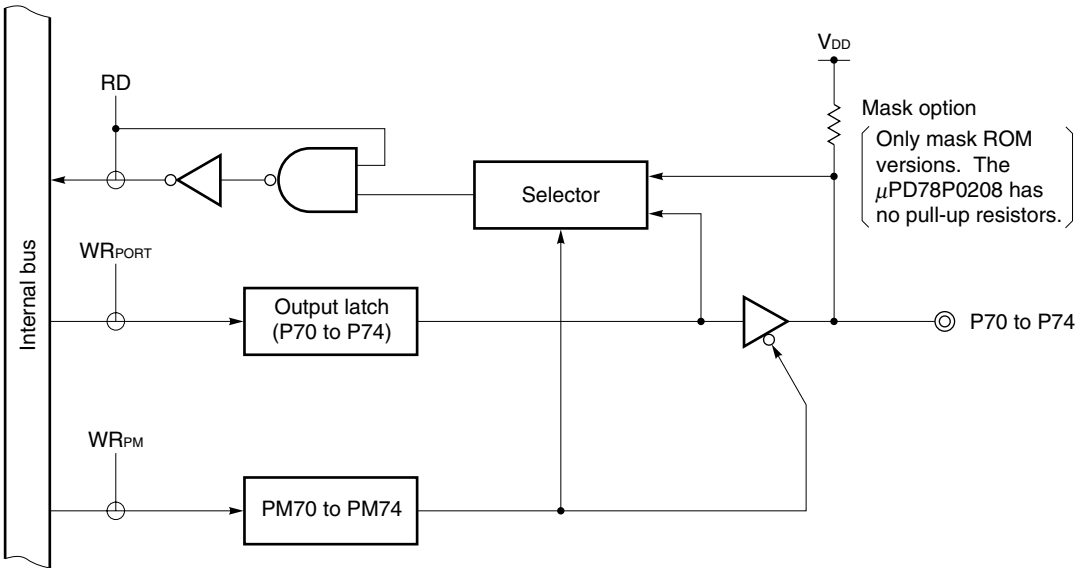
[For mask ROM version]

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

[For PROM version]

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

Figure 4-8. Block Diagram of P70 to P74



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

4.2.6 Port 8

Port 8 is an 8-bit output-only port. In mask ROM versions, use of pull-down resistors can be specified in 1-bit units with the mask option. Pull-down resistor connection to V_{LOAD} or V_{SS} can be specified in 4-bit units. The μPD78P0208 does not contain pull-down resistors.

Port 8 can drive LEDs directly.

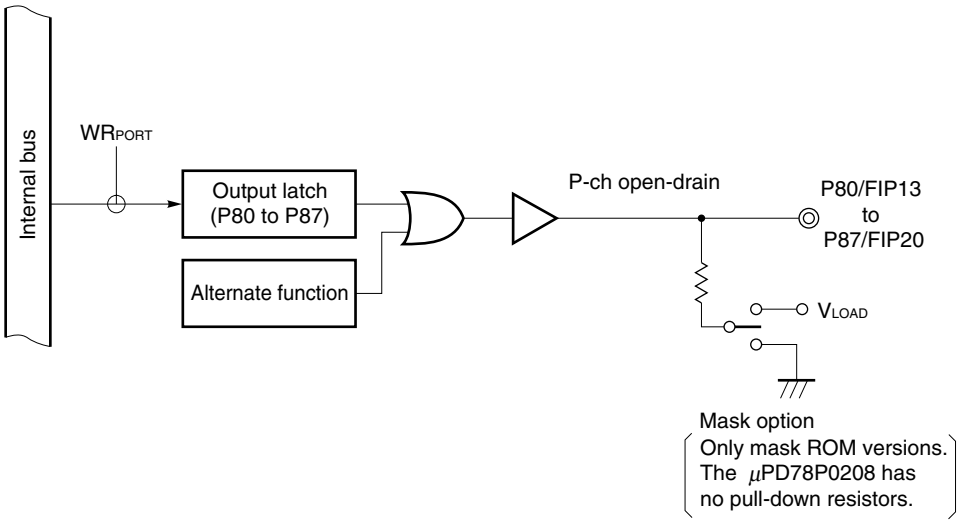
Alternate functions include VFD controller/driver display output.

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

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

★ **Caution Adjust the number of pull-down resistors so that the total power dissipation (refer to 15.10 Calculating Total Power Dissipation) is not exceeded.**

Figure 4-9. Block Diagram of P80 to P87



WR: Port 8 write signal

4.2.7 Port 9

Port 9 is an 8-bit output-only port. In mask ROM versions, use of pull-down resistors can be specified in 1-bit units with the mask option. Pull-down resistor connection to V_{LOAD} or V_{SS} can be specified in 4-bit units. The μPD78P0208 does not contain pull-down resistors.

Port 9 can drive LEDs directly.

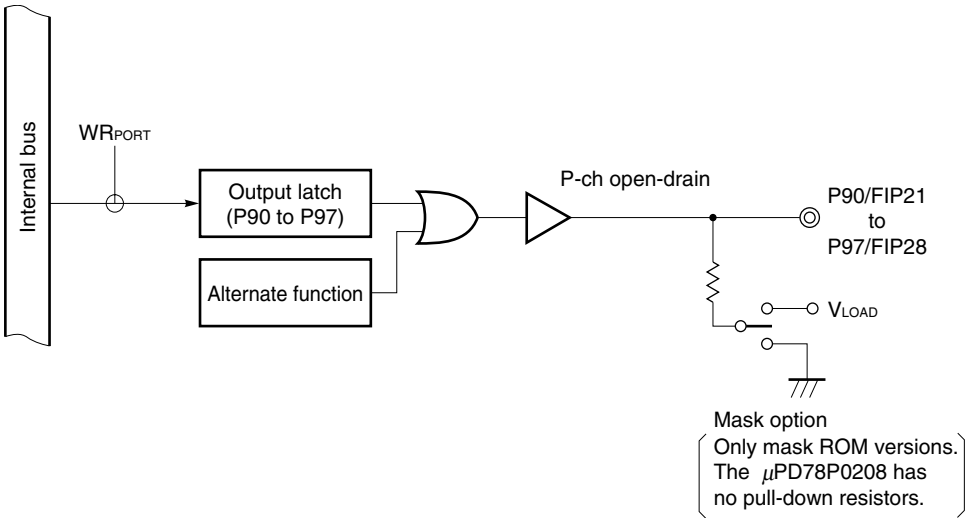
Alternate functions include VFD controller/driver display output.

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

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

★ **Caution Adjust the number of pull-down resistors so that the total power dissipation (refer to 15.10 Calculating Total Power Dissipation) is not exceeded.**

Figure 4-10. Block Diagram of P90 to P97



WR: Port 9 write signal

4.2.8 Port 10

Port 10 is an 8-bit I/O port with an output latch. The P100 to P107 pins can be set to input mode/output mode in 1-bit units using port mode register 10 (PM10). In mask ROM versions, use of pull-down resistors can be specified in 1-bit units with the mask option. Pull-down resistor connection to V_{LOAD} or V_{SS} can be specified in 4-bit units. The μPD78P0208 does not contain pull-down resistors.

Port 10 can drive LEDs directly.

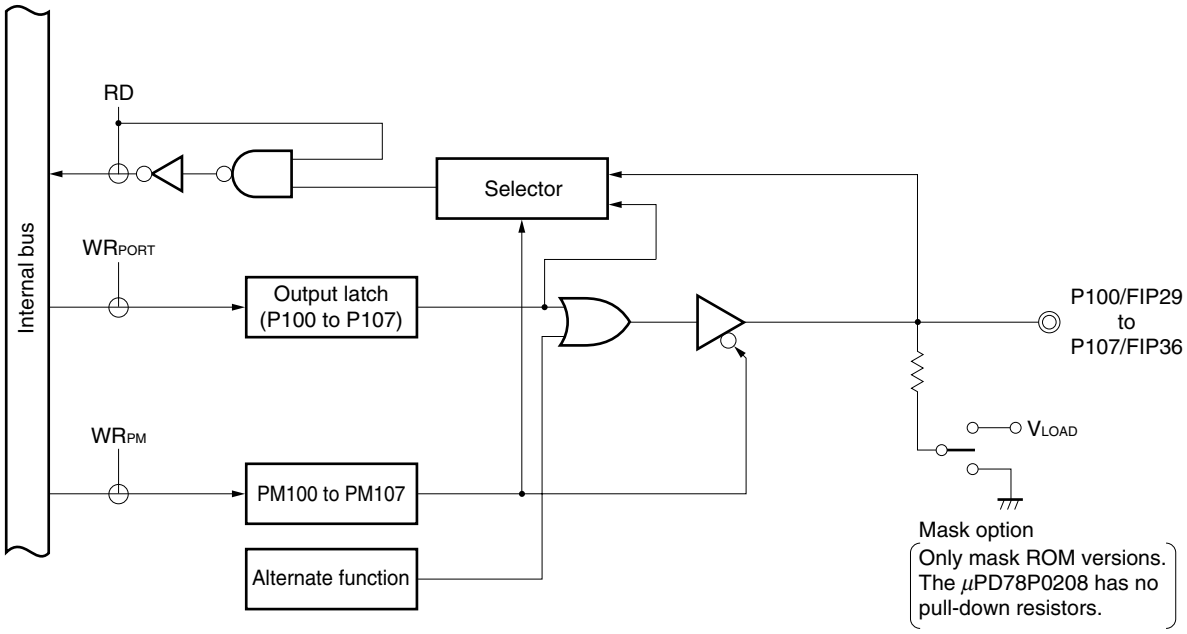
Alternate functions include VFD controller/driver display output.

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

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

★ **Caution Adjust the number of pull-down resistors so that the total power dissipation (refer to 15.10 Calculating Total Power Dissipation) is not exceeded.**

Figure 4-11. Block Diagram of P100 to P107



PM: Port mode register

RD: Port 10 read signal

WR: Port 10 write signal

4.2.9 Port 11

Port 11 is an 8-bit I/O port with an output latch. The P110 to P117 pins can be set to input mode/output mode in 1-bit units using port mode register 11 (PM11). In mask ROM versions, use of pull-down resistors can be specified in 1-bit units with the mask option. Pull-down resistor connection to V_{LOAD} or V_{SS} can be specified in 4-bit units. The μPD78P0208 does not contain pull-down resistors.

Port 11 can drive LEDs directly.

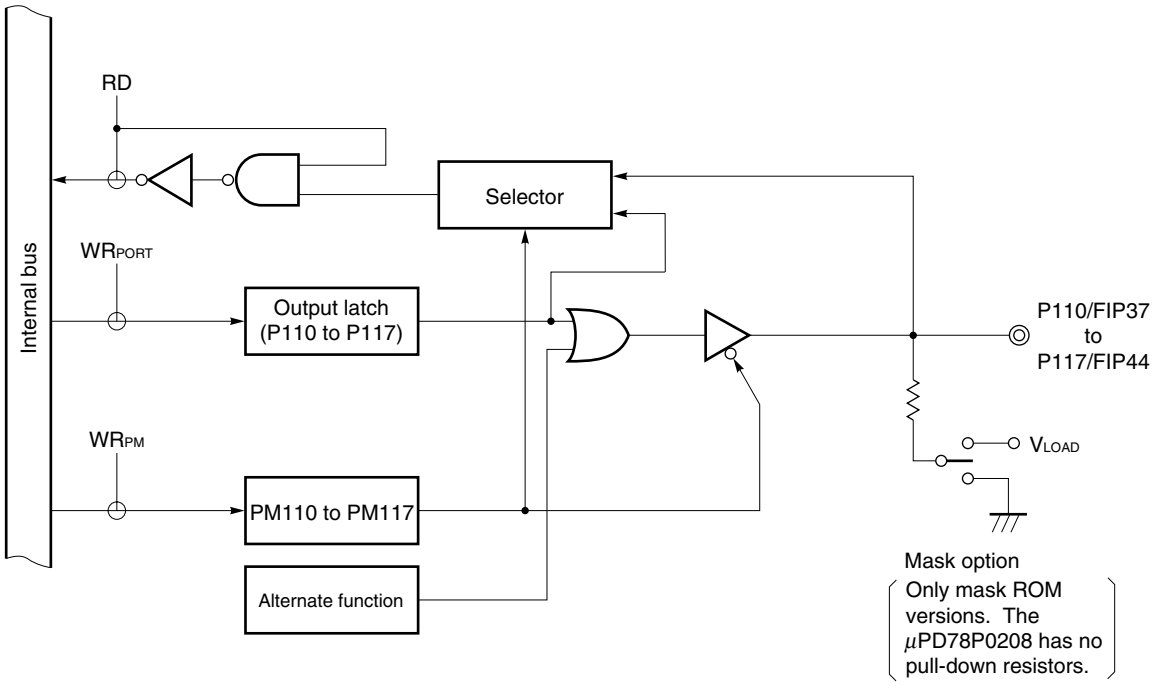
Alternate functions include VFD controller/driver display output.

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

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

★ **Caution Adjust the number of pull-down resistors so that the total power dissipation (refer to 15.10 Calculating Total Power Dissipation) is not exceeded.**

Figure 4-12. Block Diagram of P110 to P117



PM: Port mode register

RD: Port 11 read signal

WR: Port 11 write signal

4.2.10 Port 12

Port 12 is an 8-bit I/O port with an output latch. The P120 to P127 pins can be set to input mode/output mode in 1-bit units using port mode register 12 (PM12). In mask ROM versions, use of pull-down resistors can be specified in 1-bit units with the mask option. Pull-down resistor connection to V_{LOAD} or V_{SS} can be specified in 4-bit units. The μPD78P0208 does not contain pull-down resistors.

Port 12 can drive LEDs directly.

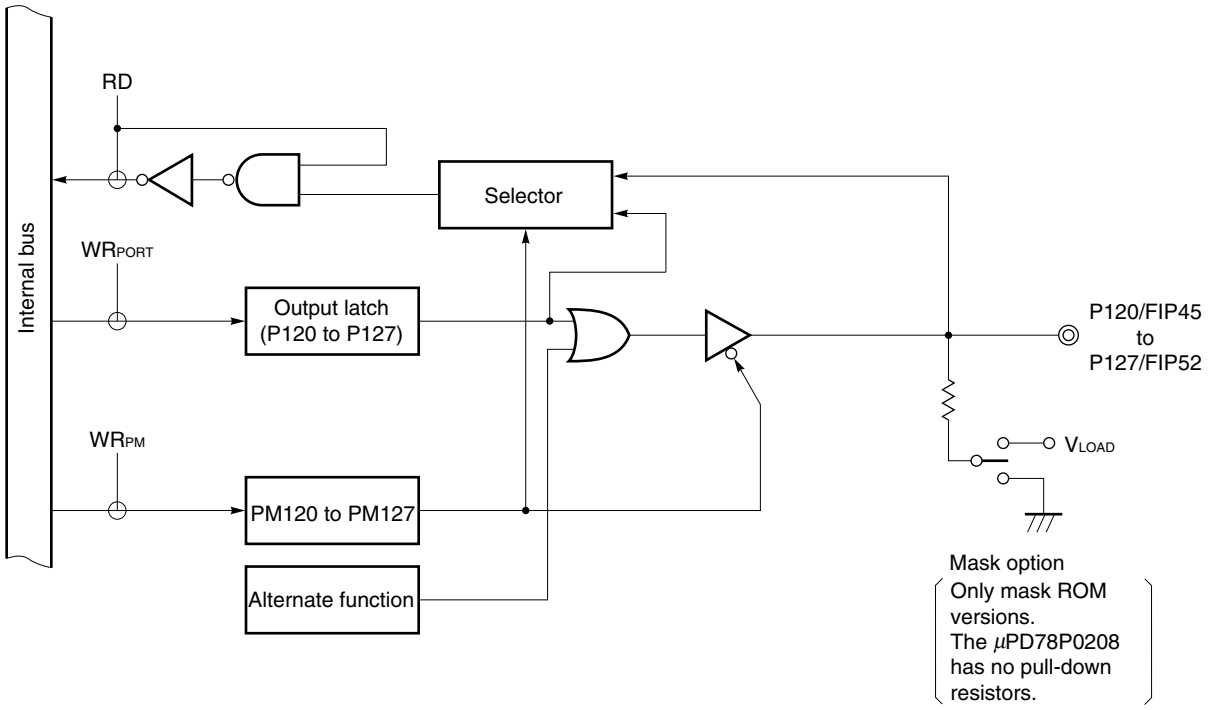
Alternate functions include VFD controller/driver display output.

$\overline{\text{RESET}}$ input sets port 12 to input mode.

Figure 4-13 shows a block diagram of port 12.

★ **Caution Adjust the number of pull-down resistors so that the total power dissipation (refer to 15.10 Calculating Total Power Dissipation) is not exceeded.**

Figure 4-13. Block Diagram of P120 to P127



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

4.3 Port Function Control Registers

The following two types of registers control the ports.

- Port mode registers (PM0, PM1, PM2, PM3, PM7, PM10, PM11, PM12)
- Pull-up resistor option register (PUO)

(1) Port mode registers (PM0, PM1, PM2, PM3, PM7, PM10, PM11, PM12)

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

PM0, PM1, PM2, PM3, PM7, PM10, PM11, and PM12 are independently set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM0 and PM7 to 1FH, and the other registers to FFH.

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

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

2. Pins P80 to P87 and P90 to P97 are output-only pins.

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

Table 4-3. Port Mode Register and Output Latch Setting When Alternate Function Is Used

Pin Name	Alternate Function		PMxx	Pxx	Pin Name	Alternate Function		PMxx	Pxx
	Function Name	I/O				Function Name	I/O		
P00	INTP0	Input	1 (fixed)	None	P33, P34	TI1, TI2	Input	1	X
	TI0	Input	1 (fixed)	None	P35	PCL	Output	0	0
P01, P02	INTP1, INTP2	Input	1	X	P36	BUZ	Output	0	0
P03	INTP3	Input	1	X	P100 to P107	FIP29 to FIP36	Output	0	0Note 2
P04Note 1	XT1	Input	1 (fixed)	None	P110 to P117	FIP37 to FIP44	Output	0	0Note 2
P10 to P17Note 1	ANI0 to ANI7	Input	1	X	P120 to P127	FIP45 to FIP52	Output	0	0Note 2
P30 to P32	TO0 to TO2	Output	0	0					

Notes 1. If a read instruction is executed to these ports in the alternate-function mode, the read data will be undefined.

2. Key scan data can be set while the VFD controller/driver is operating.

Caution When port 2 is used as serial interface pins, I/O and the output latch should be set according to the function. For the settings, refer to Figure 13-3 Format of Serial Operating Mode Register 0 and Figure 14-3 Format of Serial Operating Mode Register 1.

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

Figure 4-14. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	0	0	0	1	PM03	PM02	PM01	1	FF20H	1FH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM7	0	0	0	PM74	PM73	PM72	PM71	PM70	FF27H	1FH	R/W
PM10	PM107	PM106	PM105	PM104	PM103	PM102	PM101	PM100	FF2AH	FFH	R/W
PM11	PM117	PM116	PM115	PM114	PM113	PM112	PM111	PM110	FF2BH	FFH	R/W
PM12	PM127	PM126	PM125	PM124	PM123	PM122	PM121	PM120	FF2CH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0, 1, 2, 3, 7, 10, 11, 12 : n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

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

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

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

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

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

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

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

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

PUOm	Pm on-chip pull-up resistor selection (m = 0, 1, 2, 3)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

4.4 Port Function Operations

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

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

4.4.2 Reading from I/O port

(1) Output mode

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

(2) Input mode

The pin status is read by a transfer instruction. The output latch contents do not change.

4.4.3 Operations on I/O 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 become undefined. However, the pin status does not change because the output buffer is turned off.

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

4.5 Selection of Mask Option

The following mask option is provided in mask ROM versions. The μPD78P0208 has no mask option.

Table 4-4. Comparison Between Mask Option of Mask ROM Version and μPD78P0208

Pin Name	Mask Option of Mask ROM Version	μPD78P0208
P30/TO0 to P32/TO2, P33/TI1, P34/TI2, P35/PCL, P36/BUZ, P37	Can incorporate pull-down resistors in 1-bit units.	Does not incorporate pull-down resistors.
P70 to P74	Can incorporate pull-up resistors in 1-bit units.	Does not incorporate pull-up resistors.
FIP0 to FIP12	Can incorporate pull-down resistors in 1-bit units.	Incorporates pull-down resistors (connected to V_{LOAD}).
P80/FIP13 to P87/FIP20, P90/FIP21 to P97/FIP28, P100/FIP29 to P107/FIP36, P110/FIP37 to P117/FIP44, P120/FIP45 to P127/FIP52	Can incorporate pull-down resistors in 1-bit units. The pull-down resistors can be specified to be connected to V_{LOAD} or V_{SS} in 4-bit units from P80.	Does not incorporate pull-down resistors.

CHAPTER 5 CLOCK GENERATOR

5.1 Clock Generator Functions

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

(1) Main system clock oscillator

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

(2) Subsystem clock oscillator

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

The noise eliminator operates automatically to reduce the effect of switching noise during VFD display.

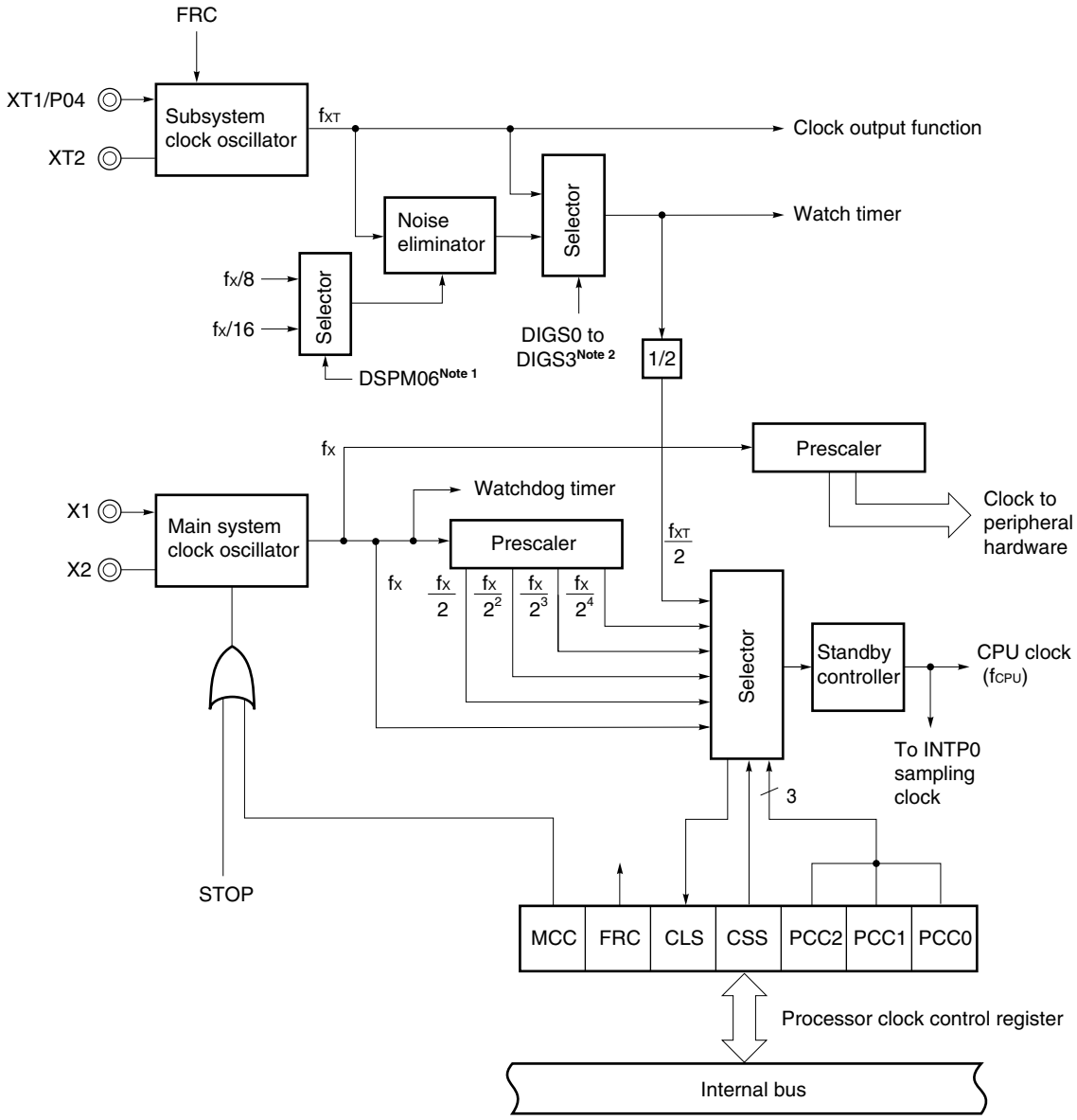
5.2 Clock Generator Configuration

The clock generator consists of the following hardware.

Table 5-1. Clock Generator Configuration

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

Figure 5-1. Clock Generator Block Diagram



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

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

5.3 Clock Generator Control Registers

The clock generator is controlled by the following three registers.

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

(1) Processor clock control register (PCC)

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

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

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

Figure 5-2. Feedback Resistor of Subsystem Clock

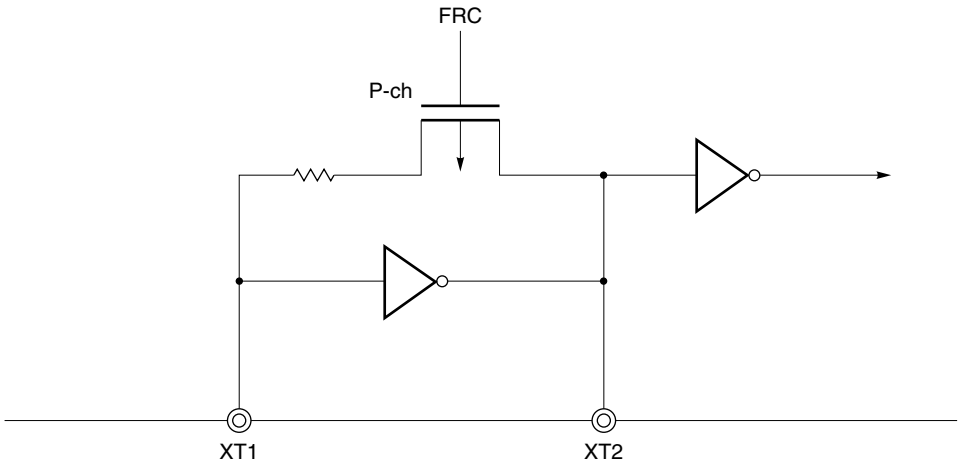


Figure 5-3. Format of Processor Clock Control Register

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

R/W

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

R

CLS	CPU clock status
0	Main system clock
1	Subsystem clock

R/W

FRC	Subsystem clock feedback resistor selection
0	Internal feedback resistor used
1	Internal feedback resistor not used[Note 2]

R/W

MCC	Main system clock oscillation control[Note 3]
0	Oscillation possible
1	Oscillation stopped

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

★ **2.** This bit can be set to 1 only when the subsystem clock is not used.

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

Cautions 1. Bit 3 must be set to 0.

2. Do not set MCC while an external clock is being input. This is because the X2 pin is pulled up to V_{DD}.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

The fastest instruction of the μPD780208 Subseries is executed in two CPU clocks. Therefore, the relationship between the CPU clock (f_{CPU}) and minimum instruction execution time is as shown in Table 5-2.

Table 5-2. Relationship Between CPU Clock and Minimum Instruction Execution Time

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

f_X = 5.0 MHz, f_{XT} = 32.768 kHz

f_X: Main system clock oscillation frequency

f_{XT}: Subsystem clock oscillation frequency

(2) Display mode register 0 (DSPM0)

This register sets the mode for the noise eliminator of the subsystem clock.

DSPM0 is set with an 8-bit memory manipulation instruction.

Only bit 7 (KSF) can be read with a 1-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets DSPM0 to 00H.

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

Figure 5-4. Format of Display Mode Register 0 (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
DSPM0	KSF	DSPM06	DSPM05	SEGS4	SEGS3	SEGS2	SEGS1	SEGS0	F F A 0 H	0 0 H	R/W

R/W	SEGS4	SEGS3	SEGS2	SEGS1	SEGS0	Display segment (display mode 1)	Display output total (display mode 2)
	0	0	0	0	0	9	9
	0	0	0	0	1	10	10
	0	0	0	1	0	11	11
	0	0	0	1	1	12	12
	0	0	1	0	0	13	13
	0	0	1	0	1	14	14
	0	0	1	1	0	15	15
	0	0	1	1	1	16	16
	0	1	0	0	0	17	17
	0	1	0	0	1	18	18
	0	1	0	1	0	19	19
	0	1	0	1	1	20	20
	0	1	1	0	0	21	21
	0	1	1	0	1	22	22
	0	1	1	1	0	23	23
	0	1	1	1	1	24	24
	1	0	0	0	0	25	25
	1	0	0	0	1	26	26
	1	0	0	1	0	27	27
	1	0	0	1	1	28	28
	1	0	1	0	0	29	29
	1	0	1	0	1	30	30
	1	0	1	1	0	31	31
	1	0	1	1	1	32	32
	1	1	0	0	0	33	33
	1	1	0	0	1	34	34
	1	1	0	1	0	35	35
	1	1	0	1	1	36	36
	1	1	1	0	0	37	37
	1	1	1	0	1	38**Note**	38
	1	1	1	1	0	39**Note**	39
	1	1	1	1	1	40**Note**	40

Note When the sum of digits and segments is over 53, specification of the number of digits has priority.

Figure 5-4. Format of Display Mode Register 0 (2/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
DSPM0	KSF	DSPM06	DSPM05	SEGS4	SEGS3	SEGS2	SEGS1	SEGS0	F F A 0 H	0 0 H	R/W[Note 1]

R/W	DSPM05	Display mode setting
	0	Display mode 1 (segment/character type)
	1	Display mode 2 (type in which a segment spans two or more grids)

R/W	DSPM06	Mode of noise eliminator for subsystem clock[Note 2]
	0	2.5 MHz < f_X ≤ 5.0 MHz
	1	1.25 MHz ≤ f_X ≤ 2.5 MHz[Note 3]

R	KSF	Timing status
	0	Display timing
	1	Key scan timing

Notes 1. Bit 7 (KSF) is a read-only bit.

2. Set this bit according to the main system clock oscillation frequency (f_X) selected. The noise eliminator operates during VFD display.

3. When f_X is used between 1.25 MHz and 2.5 MHz, set bit 6 (DSPM06) to 1 prior to VFD display.

Caution When the main system clock frequency selected is below 1.25 MHz and the VFD controller/driver is enabled, make sure to use the main system clock for watch timer counting by setting TCL24 to 0.

(3) Display mode register 1 (DSPM1)

Register to set display operation/stop.

DSPM1 is set with an 8-bit memory manipulation instruction.

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

Remark In addition to setting display operation/stop, DSPM1 can also set the display digits/number of display patterns, cut width of the VFD output, and display cycle.

Figure 5-5. Format of Display Mode Register 1

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

DIMS0	Display mode cycle setting
0	1024/fx is 1 display cycle (1 display cycle = 204.8 μs: @ 5.0 MHz operation)
1	2048/fx is 1 display cycle (1 display cycle = 409.6 μs: @ 5.0 MHz operation)

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

DIGS3	DIGS2	DIGS1	DIGS0	Display digits (display mode 1) DSPM05 = 0	Display patterns (display mode 2) DSPM05 = 1
0	0	0	0	Display stopped (static display)**Note**	Display stopped (static display)**Note**
0	0	0	1	2 digits	2 patterns
0	0	1	0	3 digits	3 patterns
0	0	1	1	4 digits	4 patterns
0	1	0	0	5 digits	5 patterns
0	1	0	1	6 digits	6 patterns
0	1	1	0	7 digits	7 patterns
0	1	1	1	8 digits	8 patterns
1	0	0	0	9 digits	9 patterns
1	0	0	1	10 digits	10 patterns
1	0	1	0	11 digits	11 patterns
1	0	1	1	12 digits	12 patterns
1	1	0	0	13 digits	13 patterns
1	1	0	1	14 digits	14 patterns
1	1	1	0	15 digits	15 patterns
1	1	1	1	16 digits	16 patterns

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

Remarks 1. fx: Main system clock oscillation frequency

2. DSPM05: Bit 5 of display mode register 0 (DSPM0)

5.4 System Clock Oscillator

5.4.1 Main system clock oscillator

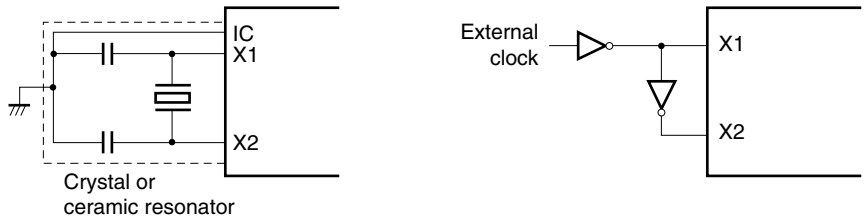
The main system clock oscillator oscillates with a crystal resonator or a ceramic resonator (standard: 5.0 MHz) connected to the X1 and X2 pins.

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

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

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

(a) Crystal or ceramic oscillation **(b) External clock**



Caution Do not execute the STOP instruction or set bit 7 (MCC) of the processor clock control register (PCC) to 1 while an external clock is being input. This is because the operation of the main system clock is stopped and the X2 pin is pulled up to V_{DD} if the STOP instruction is executed or MCC is set to 1.

5.4.2 Subsystem clock oscillator

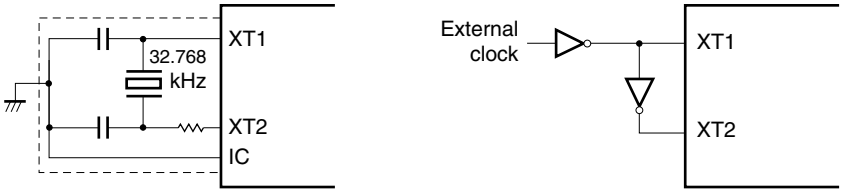
The subsystem clock oscillator oscillates with a crystal resonator (standard: 32.768 kHz) connected to the XT1 and XT2 pins.

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

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

Figure 5-7. External Circuit of Subsystem Clock Oscillator

(a) Crystal oscillation **(b) External clock**



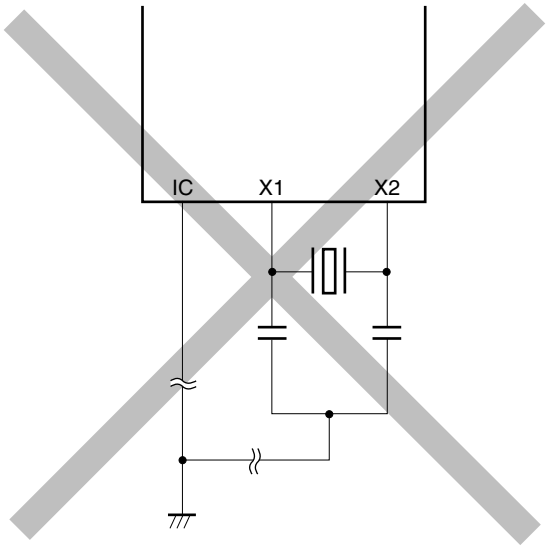
Cautions 1. When using the main system or subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in Figures 5-6 and 5-7 to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

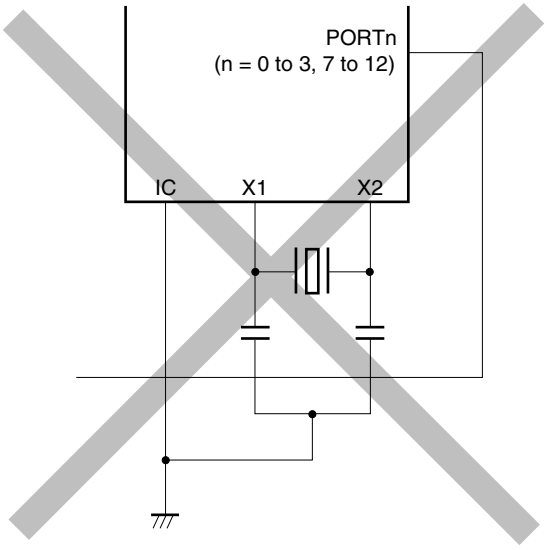
When using the subsystem clock oscillator, pay special attention because the subsystem clock oscillator has low amplification to minimize power consumption.
Figure 5-8 shows examples of incorrect resonator connection.

Figure 5-8. Examples of Incorrect Resonator Connection (1/2)

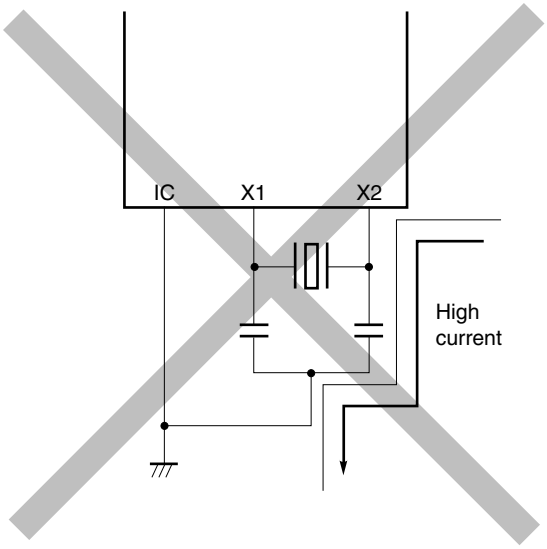
(a) Too long wiring of connected circuit



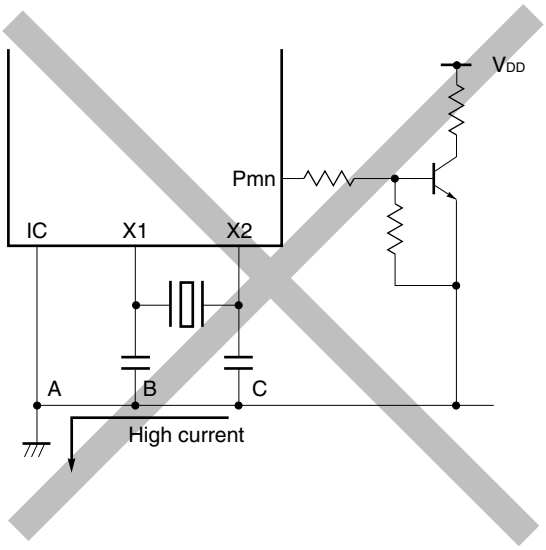
(b) Crossed signal lines



(c) High alternating current close to signal lines



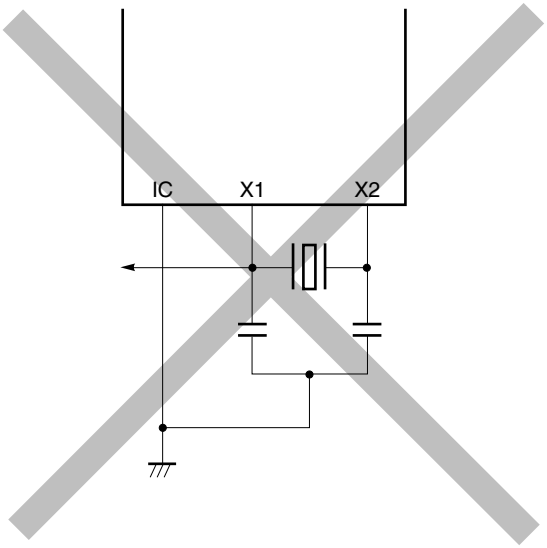
(d) Current flowing through ground line of oscillator (potentials at points A, B, and C change)



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

Figure 5-8. Examples of Incorrect Resonator Connection (2/2)

(e) Signal fetched



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.

Cautions 2. If XT2 and X1 are wired in parallel, malfunction may occur due to the crosstalk noise between XT2 and X1.
To prevent this, connect the IC pin directly to the V_{SS} pin located between the XT2 and X1 pins, and do not wire XT2 and X1 in parallel.

5.4.3 Divider

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

5.4.4 When subsystem clock is not used

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

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

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

5.5 Clock Generator Operations

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

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

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

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

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

(c) With the main system clock selected, two standby modes, the STOP and HALT modes, are available. When the system is not using the subsystem clock, the power consumption in the STOP mode can be decreased if the internal feedback resistor is not used by setting bit 6 (FRC) of PCC.

(d) PCC can be used to select the subsystem clock and to operate the system with low power consumption (122 μs when operated at 32.768 kHz).

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

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

5.5.1 Main system clock operations

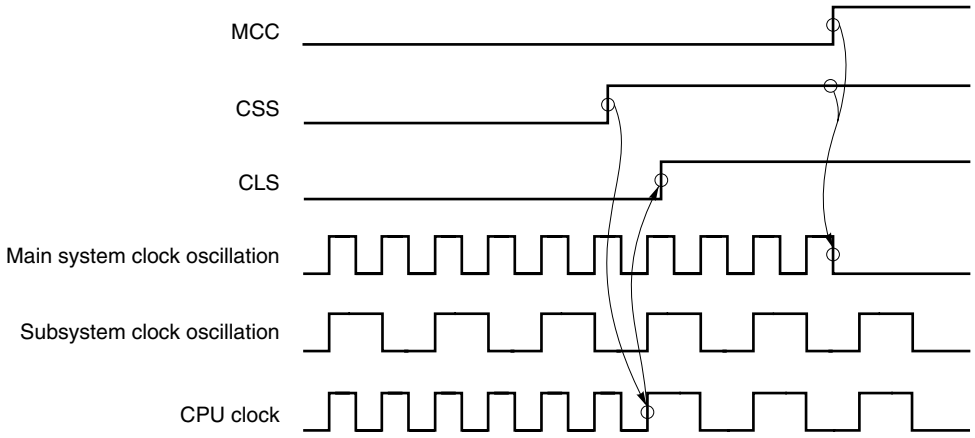
During operation with the main system clock (when bit 5 (CLS) of the processor clock control register (PCC) is set to 0), the following operations are carried out via PCC settings.

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

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

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

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



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

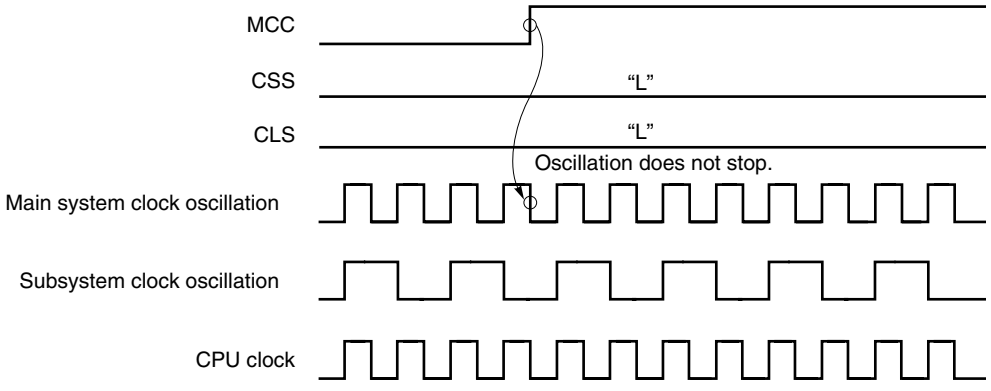
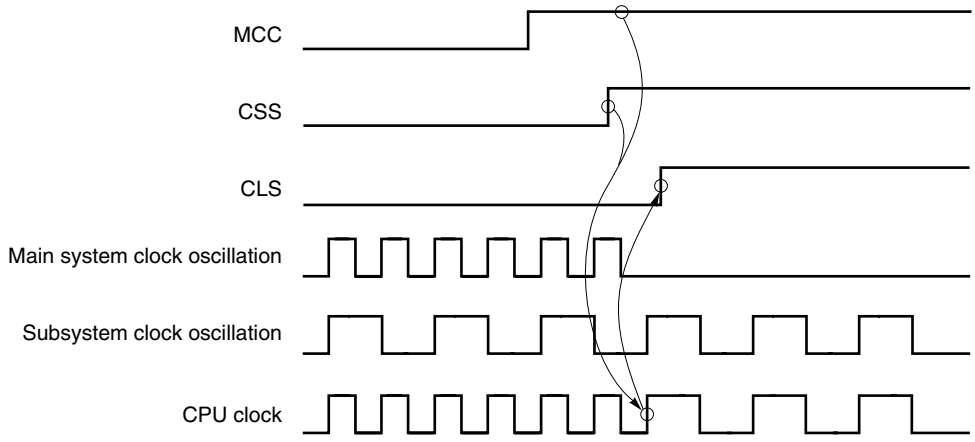


Figure 5-9. Main System Clock Stop Function (2/2)

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



5.5.2 Subsystem clock operations

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

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

(b) Watchdog timer counting stops.

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

5.6 Changing System Clock and CPU Clock Settings

5.6.1 Time required for switchover between system clock and CPU clock

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

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

It can be judged by bit 5 (CLS) of PCC whether the system is operating on the main system clock or the subsystem clock.

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

Set Values Before Switchover				Set Values After Switchover																							
CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0
				0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	x	x	x
0	0	0	0					16 instructions				16 instructions				16 instructions				16 instructions				$f_X/2f_{XT}$ instructions (64 instructions)			
	0	0	1	8 instructions								8 instructions				8 instructions				8 instructions				$f_X/4f_{XT}$ instructions (32 instructions)			
	0	1	0	4 instructions				4 instructions								4 instructions				4 instructions				$f_X/8f_{XT}$ instructions (16 instructions)			
	0	1	1	2 instructions				2 instructions				2 instructions								2 instructions				$f_X/16f_{XT}$ instructions (8 instructions)			
	1	0	0	1 instruction				1 instruction				1 instruction				1 instruction								$f_X/32f_{XT}$ instructions (4 instructions)			
1	x	x	x	1 instruction				1 instruction				1 instruction				1 instruction				1 instruction							

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

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

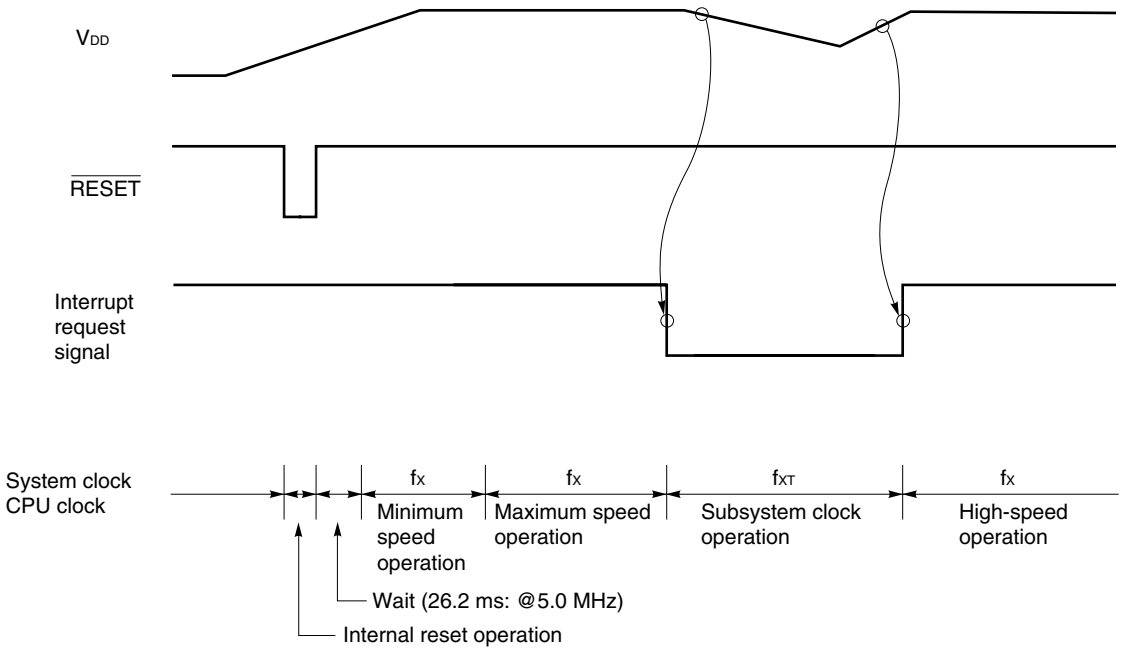
Remarks 1. One instruction is the minimum instruction execution time with the pre-switchover CPU clock.

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

5.6.2 System clock and CPU clock switching procedure

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

Figure 5-10. System Clock and CPU Clock Switching



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

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

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

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

Caution When the main system clock is stopped and the subsystem clock is operating, switch to the main system clock after securing the oscillation stabilization time by program.

CHAPTER 6 16-BIT TIMER/EVENT COUNTER

6.1 Outline of Timers Incorporated in μPD780208 Subseries

This chapter explains the 16-bit timer/event counter. First of all, the timers incorporated in the μPD780208 Subseries and other related parts are outlined below.

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

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

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

TM1 and TM2 can be used for 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 (refer to **CHAPTER 7 8-BIT TIMER/EVENT COUNTER**).

(3) Watch timer (TM3)

This timer can set a flag every 0.5 seconds and simultaneously generate interrupt requests at preset time intervals (refer to **CHAPTER 8 WATCH TIMER**).

(4) Watchdog timer (WDTM)

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

(5) Clock output controller

This circuit supplies other devices with the divided main system clock and the subsystem clock (refer to **CHAPTER 10 CLOCK OUTPUT CONTROLLER**).

(6) Buzzer output controller

This circuit outputs the buzzer frequency obtained by dividing the main system clock (refer to **CHAPTER 11 BUZZER OUTPUT CONTROLLER**).

Table 6-1. Timer/Event Counter Operations

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

Notes 1. 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.

6.2 16-Bit Timer/Event Counter Functions

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

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

(1) Interval timer

TM0 generates interrupt requests at the preset time interval.

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

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

Remarks 1. fx: Main system clock oscillation frequency

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

(2) PWM output

TM0 can generate 14-bit resolution PWM output.

(3) Pulse width measurement

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

(4) External event counter

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

(5) Square-wave output

TM0 can output a square wave with any selected frequency.

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

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

Remarks 1. fx: Main system clock oscillation frequency

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

6.3 16-Bit Timer/Event Counter Configuration

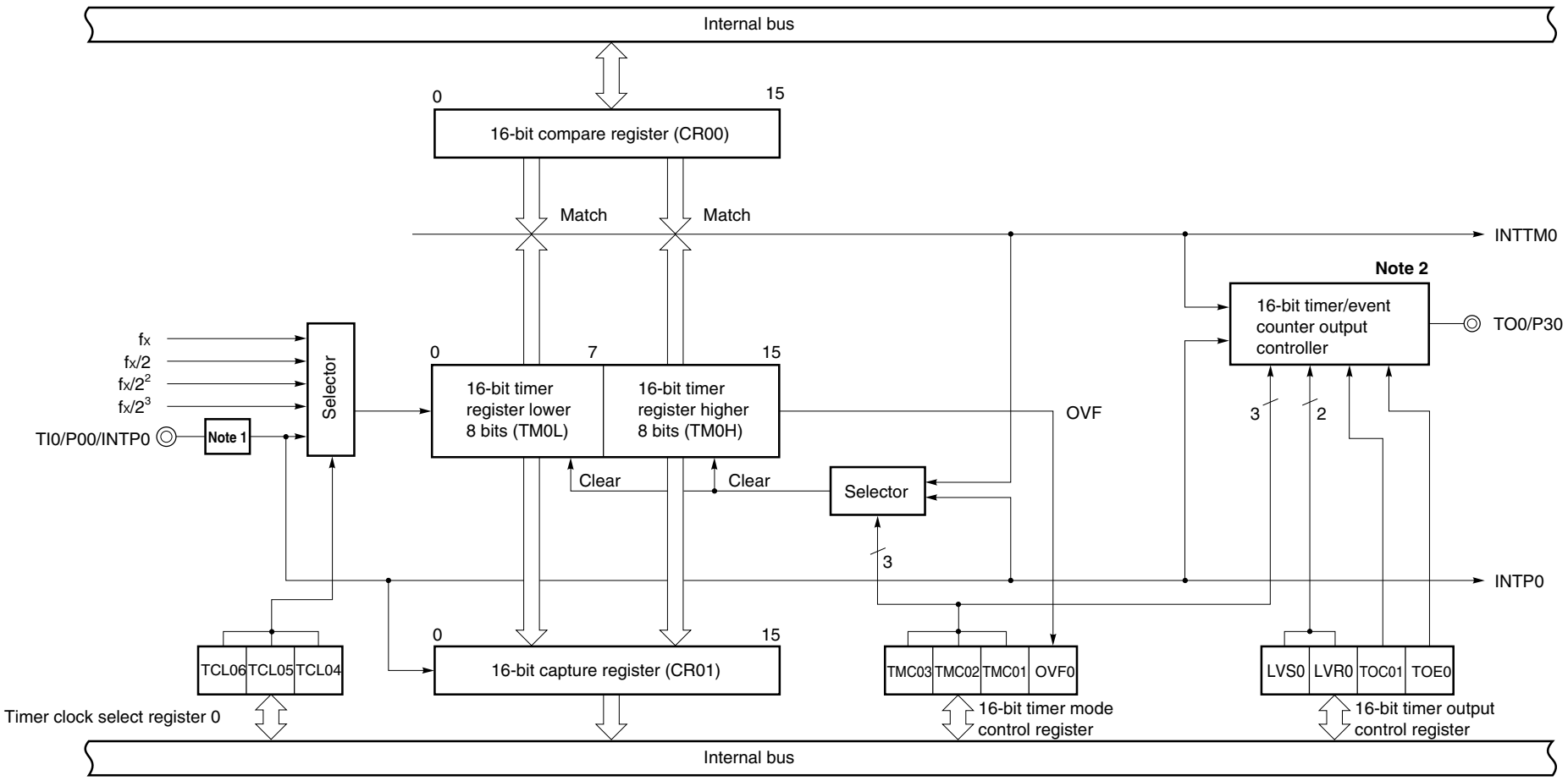
The 16-bit timer/event counter consists of the following hardware.

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

Item	Configuration
Timer register	16 bits x 1 (TM0)
Registers	16-bit compare register: 1 (CR00) 16-bit capture register: 1 (CR01)
Timer outputs	1 (TO0)
Control registers	Timer clock select register 0 (TCL0) 16-bit timer mode control register (TMC0) 16-bit timer output control register (TOC0) Port mode register 3 (PM3) External interrupt mode register (INTM0) Sampling clock select register (SCS)**Note**

Note Refer to **Figure 16-1 Basic Configuration of Interrupt Function**.

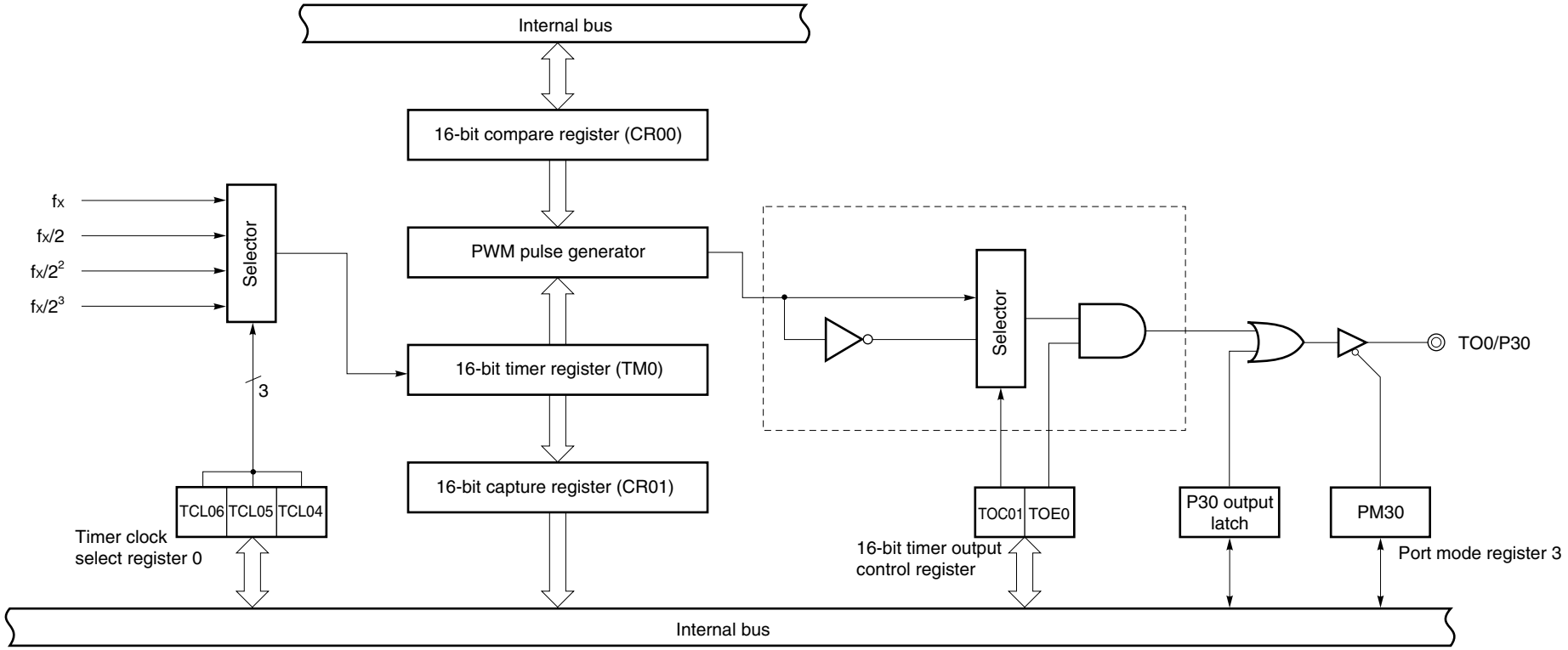
Figure 6-1. Block Diagram of 16-Bit Timer/Event Counter (Timer Mode)



Notes 1. Edge detector

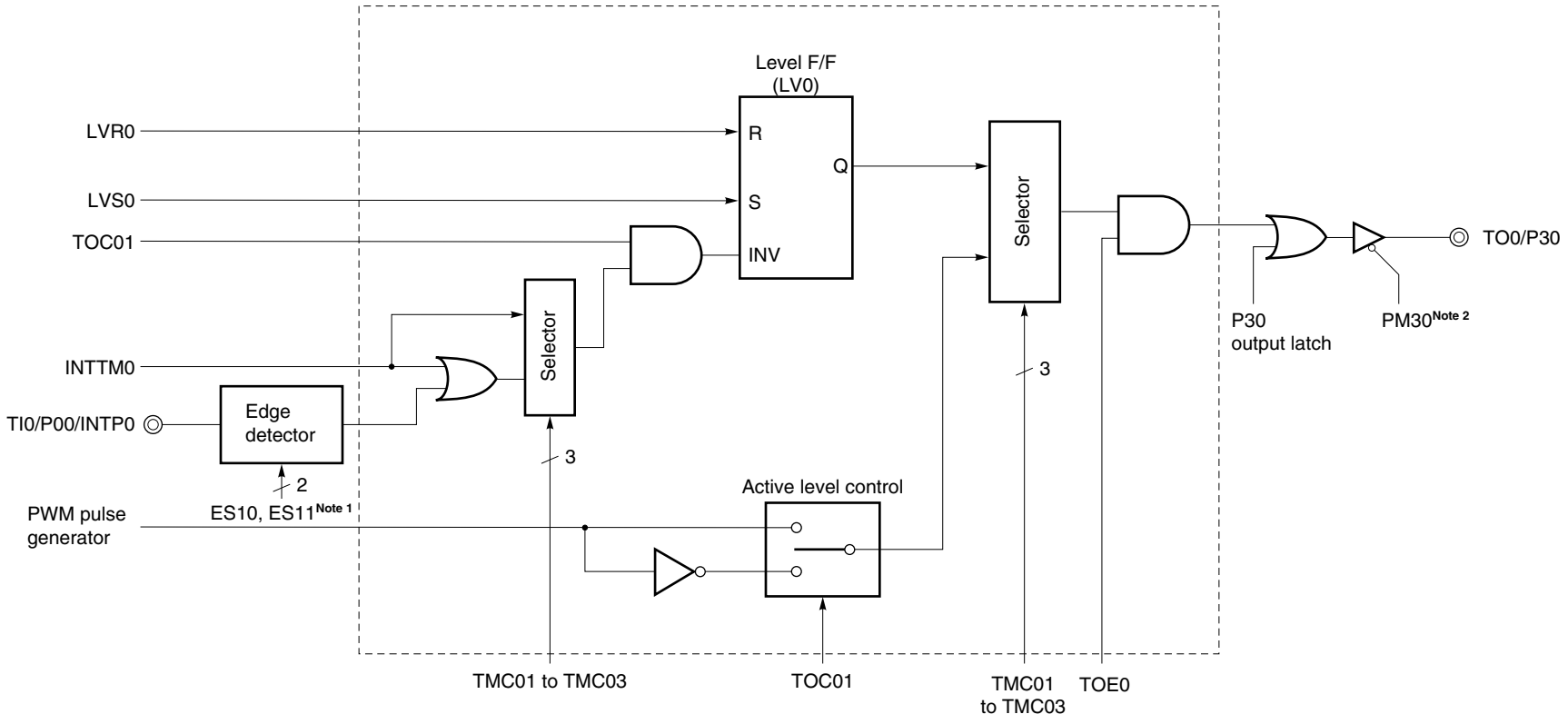
2. For the configuration of the 16-bit timer/event counter output controller, refer to **Figure 6-3**.

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



Remark The circuitry enclosed by the dotted line is the output controller.

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



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

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

Remark The circuitry enclosed by the dotted line is the output controller.

(1) 16-bit compare register (CR00)

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

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

CR00 is set with a 16-bit memory manipulation instruction. Values from 0001H to FFFFH can be set.

$\overline{\text{RESET}}$ input makes CR00 undefined.

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

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

3. When the value after CR00 is changed is smaller than that of the 16-bit timer register (TM0), TM0 continues to count and overflows, then resumes counting from 0. Therefore, if the value after CR00 is changed is smaller than the value before CR00 is changed, the timer needs to be restarted after CR00 is changed.

(2) 16-bit capture register (CR01)

CR01 is a 16-bit register used to capture the contents of the 16-bit timer (TM0).

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

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

$\overline{\text{RESET}}$ input makes CR01 undefined.

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

(3) 16-bit timer register (TM0)

TM0 is a 16-bit register that counts the count pulse.

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

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

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

6.4 16-Bit Timer/Event Counter Control Registers

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

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

(1) Timer clock select register 0 (TCL0)

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

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

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

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

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

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

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

TCL06	TCL05	TCL04	16-bit timer register count clock selection
0	0	0	TI0 (Valid edge specifiable)
0	0	1	f_X (5.0 MHz)
0	1	0	$f_X/2$ (2.5 MHz)
0	1	1	$f_X/2^2$ (1.25 MHz)
1	0	0	$f_X/2^3$ (625 kHz)
Other than above			Setting prohibited

CLOE	PCL output control
0	Output disabled
1	Output enabled

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

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

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

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

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

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

4. TM0: 16-bit timer register

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

6. Refer to **CHAPTER 10 CLOCK OUTPUT CONTROLLER** for PCL.

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

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

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

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

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

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

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

OVF0	16-bit timer register overflow detection
0	Overflow not detected
1	Overflow detected

TMC03	TMC02	TMC01	Operating mode & clear mode selection	TO0 output timing selection	Interrupt request generation
0	0	0	Operation stop (TM0 cleared to 0)	No change	Not generated
0	0	1	PWM mode (free-running)	PWM pulse output	Generated on match between TM0 and CR00
0	1	0	Free-running mode	Match between TM0 and CR00	
0	1	1		Match between TM0 and CR00 or TI0 valid edge	
1	0	0	Clear & start on TI0 valid edge	Match between TM0 and CR00	
1	0	1		Match between TM0 and CR00 or TI0 valid edge	
1	1	0	Clear & start on match between TM0 and CR00	Match between TM0 and CR00	
1	1	1		Match between TM0 and CR00 or TI0 valid edge	

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

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

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

Remark TO0: 16-bit timer/event counter output pin
TI0: 16-bit timer/event counter input pin
TM0: 16-bit timer register
CR00: Compare register 00

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

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

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

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

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

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

TOE0	16-bit timer/event counter output control
0	Output disabled (port mode)
1	Output enabled

TOC01	In PWM mode	In other modes
	Active level selection	Timer output F/F control
0	Active high	Inversion operation disabled
1	Active low	Inversion operation enabled

LVS0	LVR0	16-bit timer/event counter timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

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

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

(4) Port mode register 3 (PM3)

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

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

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

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

Figure 6-7. Format of Port Mode Register 3

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

PM3n	P3n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

(5) External interrupt mode register (INTM0)

This register is used to set the INTP0 to INTP2 and TI0 valid edges.

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

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

Remarks 1. The INTP0 pin is also used as TI0/P00.

2. The valid edge of INTP3 is fixed to the falling edge.

Figure 6-8. Format of External Interrupt Mode Register

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

ES11	ES10	INTP0/TI0 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

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

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

★ **Caution When using the INTP0/TI0/P00 pin as a timer input pin (TI0), stop the operation of the 16-bit timer by clearing bits 1 to 3 (TMC01 to TMC03) of the 16-bit timer mode control register (TMC0) to 0, 0, 0, before setting the valid edge of TI0. When using the INTP0/TI0/P00 pin as an external interrupt input pin (INTP0), the valid edge of INTP0 may be set while the 16-bit timer is operating.**

(6) Sampling clock select register (SCS)

This register sets the clock to be used for sampling the valid edge input to INTP0. When remote controlled reception is carried out using INTP0, digital noise is eliminated using the sampling clock.

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

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

Figure 6-9. Format of Sampling Clock Select Register

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

SCS1	SCS0	INTP0 sampling clock selection
0	0	$f_X/2^{N+1}$
0	1	Setting prohibited
1	0	$f_X/2^6$ (78.1 kHz)
1	1	$f_X/2^7$ (39.1 kHz)

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

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

2. f_X: Main system clock oscillation frequency

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

6.5 16-Bit Timer/Event Counter Operations

6.5.1 Interval timer operations

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

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

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

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

Figure 6-10. Interval Timer Configuration Diagram

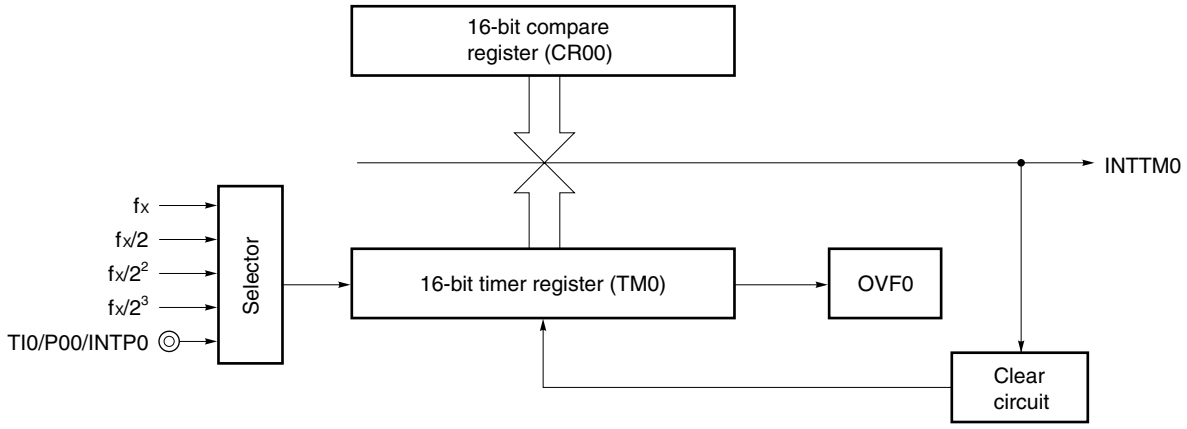
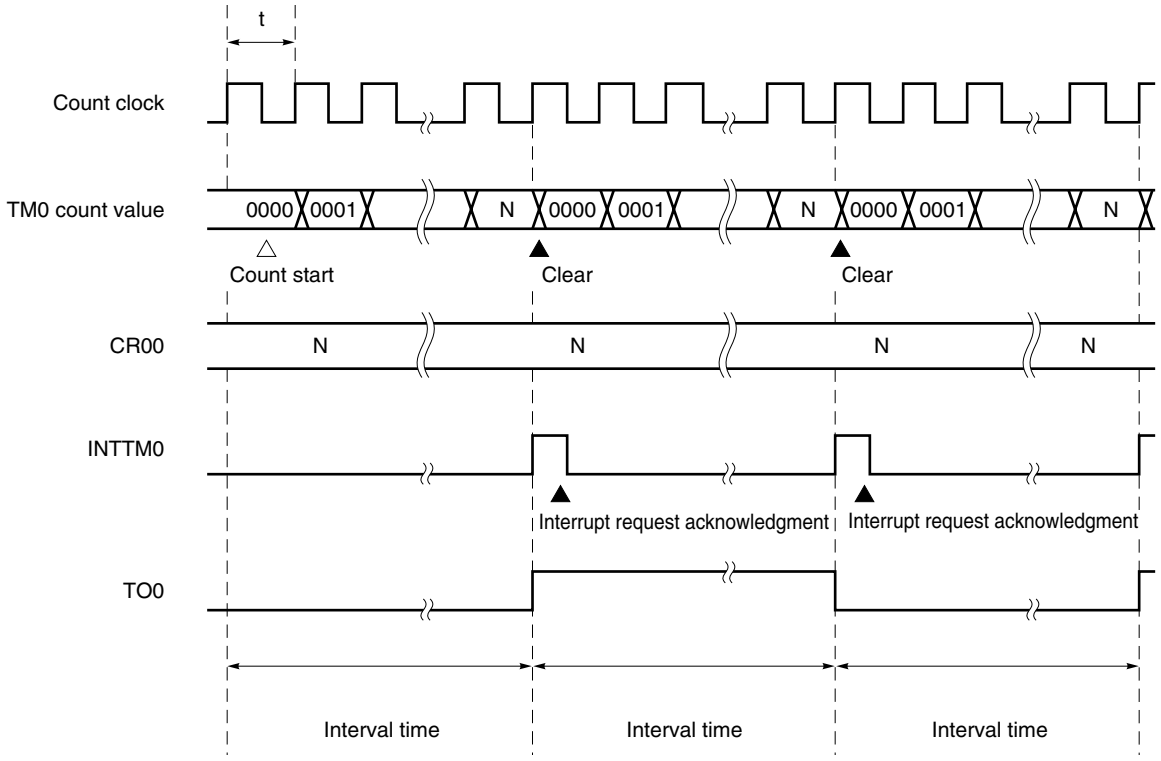


Figure 6-11. Interval Timer Operation Timing



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

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

TCL06	TCL05	TCL04	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	2 x TI0 input cycle	2^{16} x TI0 input cycle	TI0 input edge cycle
0	0	1	2 x 1/fx (400 ns)	2^{16} x 1/fx (13.1 ms)	1/fx (200 ns)
0	1	0	2^2 x 1/fx (800 ns)	2^{17} x 1/fx (26.2 ms)	2 x 1/fx (400 ns)
0	1	1	2^3 x 1/fx (1.6 μs)	2^{18} x 1/fx (52.4 ms)	2^2 x 1/fx (800 ns)
1	0	0	2^4 x 1/fx (3.2 μs)	2^{19} x 1/fx (104.9 ms)	2^3 x 1/fx (1.6 μs)
Other than above			Setting prohibited		

Remarks 1. fx: Main system clock oscillation frequency

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

6.5.2 PWM output operations

By setting bits 1 to 3 (TMC01 to TMC03) of the 16-bit timer mode control register (TMC0) to 1, 0, 0, the 16-bit timer/event counter operates as PWM output. Pulses with a duty determined by the value set in the 16-bit compare register (CR00) beforehand are output from the TO0/P30 pin.

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

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

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

Cautions 1. CR00 should be set after selecting the PWM operation mode.

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

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

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

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

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

V_{REF}: External switching circuit reference voltage

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

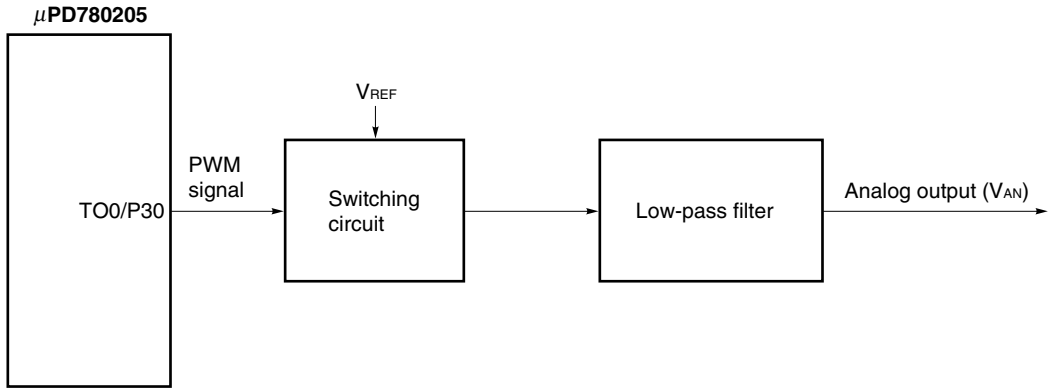
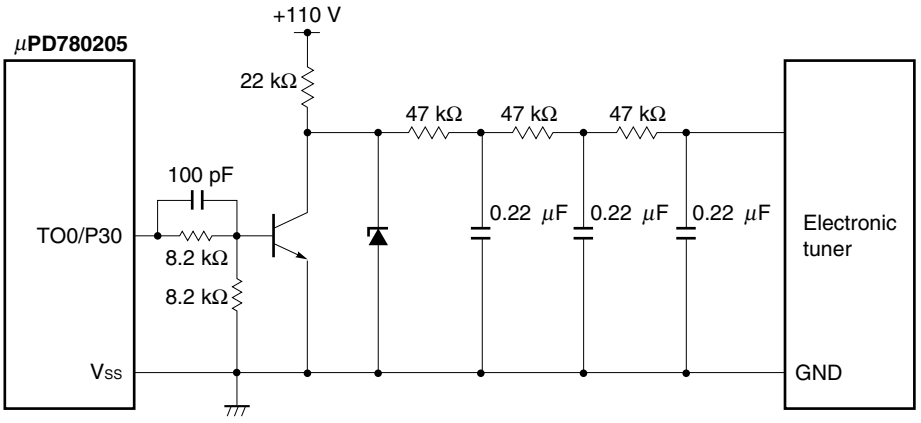


Figure 6-13 shows an example in which PWM output is converted to an analog voltage and used in a voltage synthesizer type TV tuner.

Figure 6-13. TV Tuner Application Circuit Example



6.5.3 Pulse width measurement operations

It is possible to measure the pulse width of the signals input to the TI0/P00 pin using the 16-bit timer register (TM0).

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

(1) Pulse width measurement in free-running mode

When the 16-bit timer register (TM0) is operated in free-running mode, if the edge specified by the external interrupt mode register (INTM0) is input, the value of TM0 is taken into the capture register (CR01) and an external interrupt request signal (INTP0) is set.

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

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

Figure 6-14. Configuration Diagram for Pulse Width Measurement in Free-Running Mode

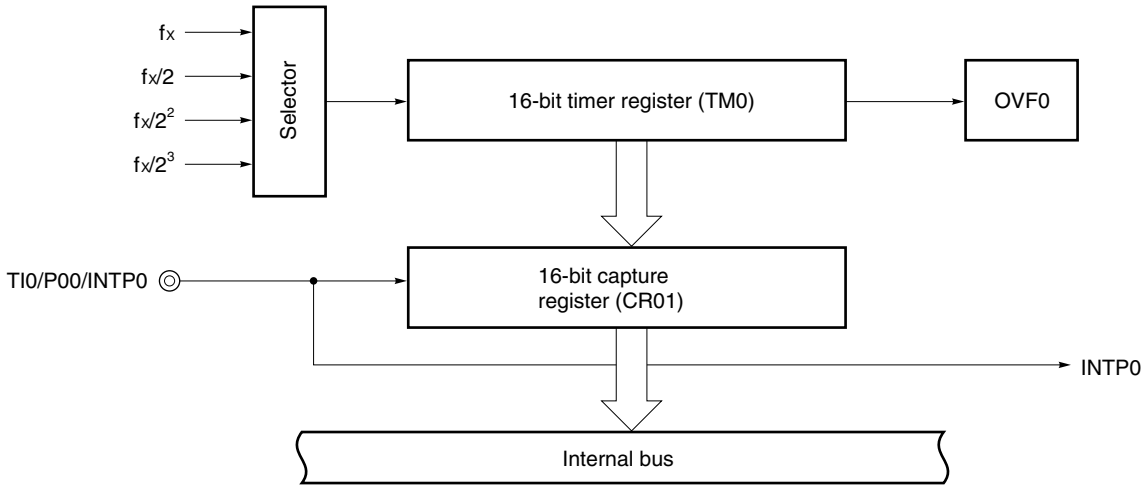
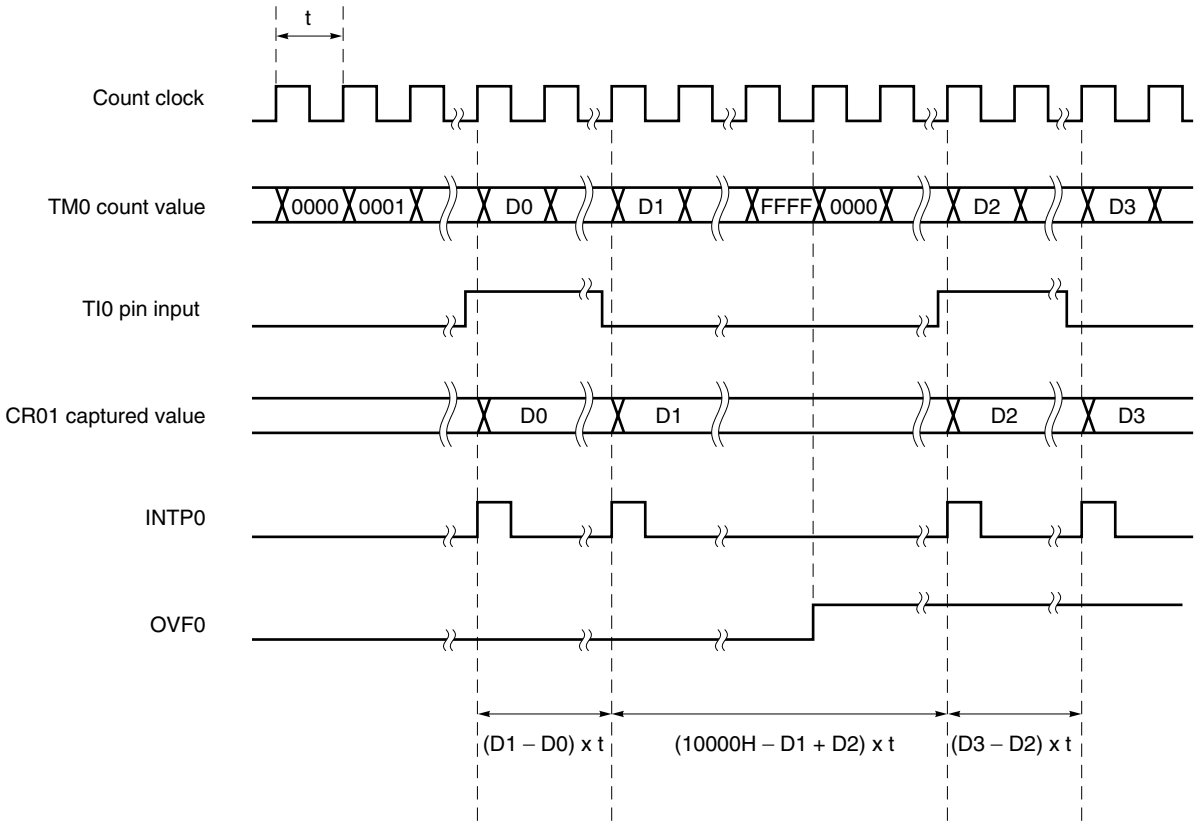


Figure 6-15. Timing of Pulse Width Measurement Operation in Free-Running Mode (with Both Edges Specified)



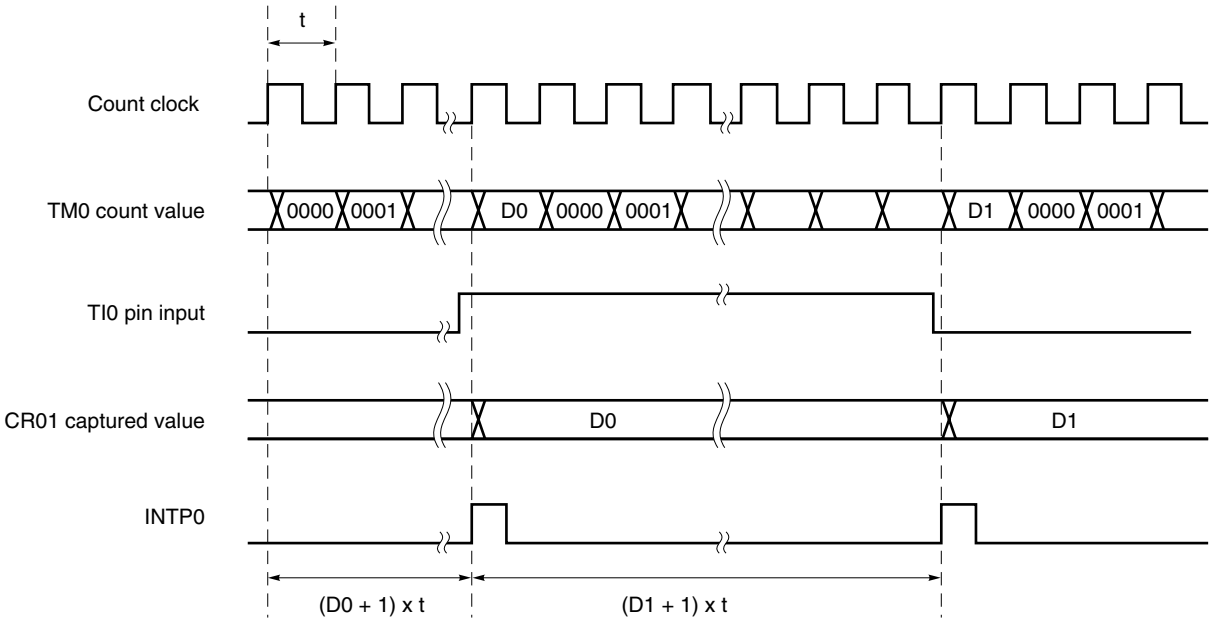
(2) Pulse width measurement by means of restart

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

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

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

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



6.5.4 External event counter operation

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

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

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

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

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

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

Figure 6-17. External Event Counter Configuration Diagram

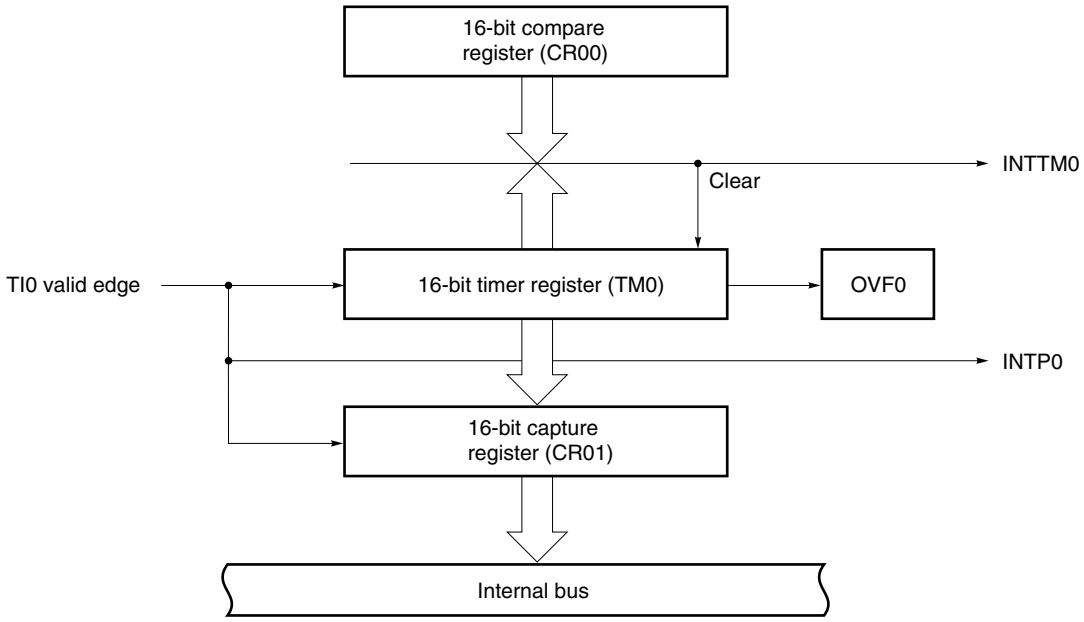
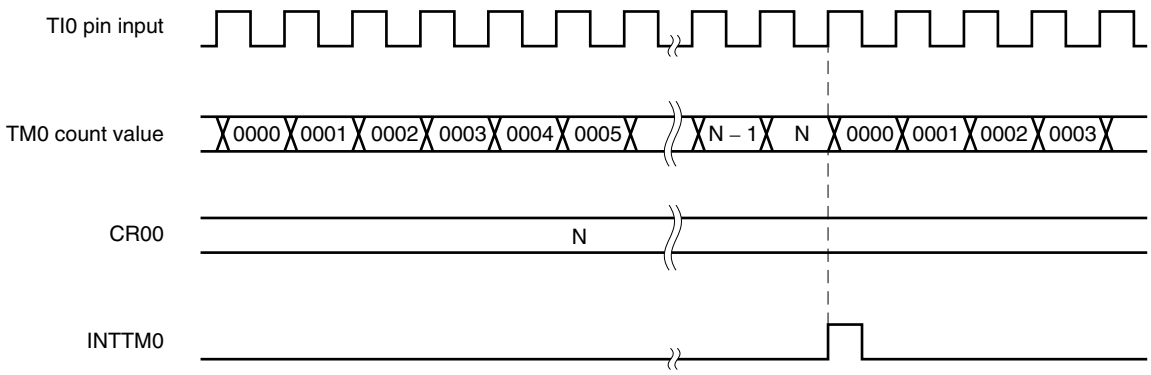


Figure 6-18. External Event Counter Operation Timing (with Rising Edge Specified)



6.5.5 Square-wave output operation

The 16-bit timer/event counter outputs a square-wave of any frequency with the value preset to the 16-bit compare register (CR00) as the interval.

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.

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

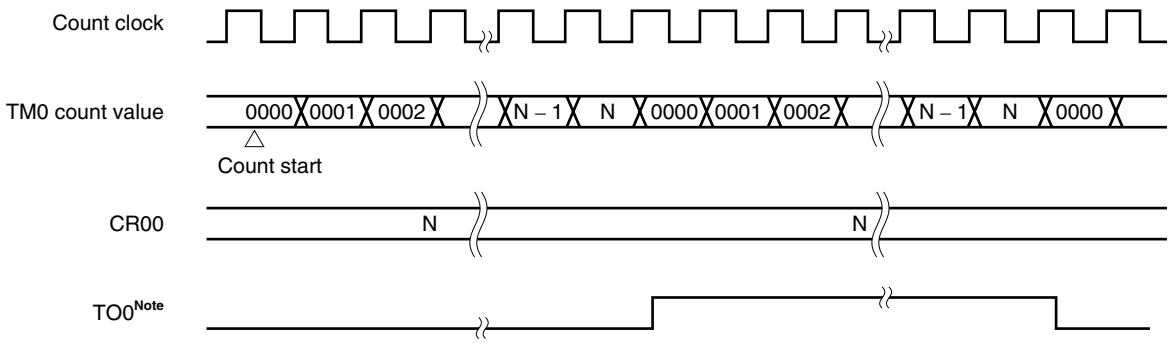
TCL06	TCL05	TCL04	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	0	2 x TI0 input cycle	2^{16} x TI0 input cycle	TI0 input edge cycle
0	0	1	2 x 1/fx (400 ns)	2^{16} x 1/fx (13.1 ms)	1/fx (200 ns)
0	1	0	2^2 x 1/fx (800 ns)	2^{17} x 1/fx (26.2 ms)	2 x 1/fx (400 ns)
0	1	1	2^3 x 1/fx (1.6 μs)	2^{18}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

Remarks 1. fx: Main system clock oscillation frequency

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

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

Figure 6-19. Square-Wave Output Operation Timing



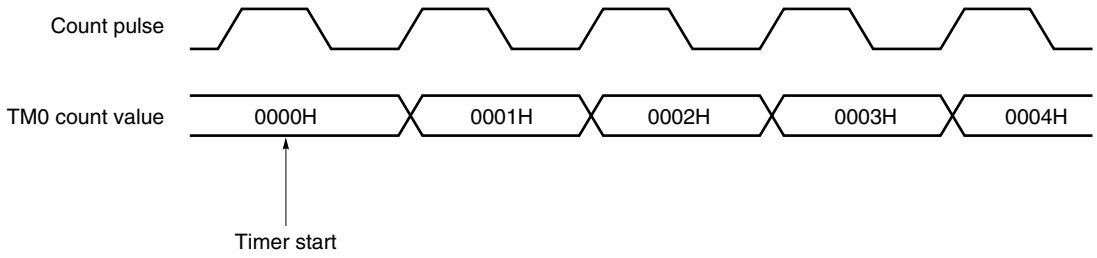
Note Initial value of TO0 output can be set by bits 2 and 3 (LVR0 and LVS0) of the 16-bit timer output control register (TOC0).

6.6 16-Bit Timer/Event Counter Operating Precautions

(1) Timer start errors

An error of up to one clock may occur in 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 to the count pulse.

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



(2) 16-bit compare register setting

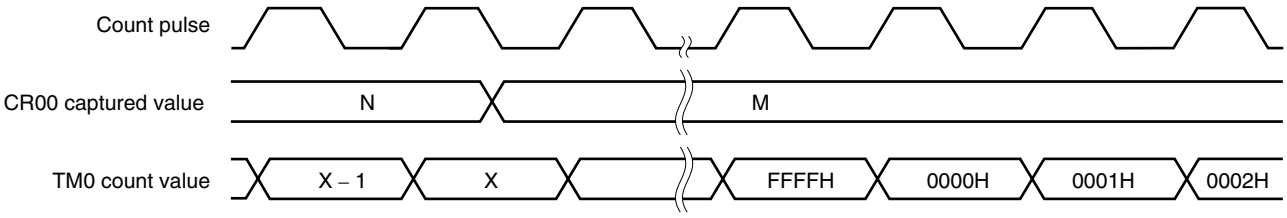
Set the 16-bit compare register (CR00) to a value other than 0000H.

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

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

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

Figure 6-21. Timing After Compare Register Change During Timer Count Operation

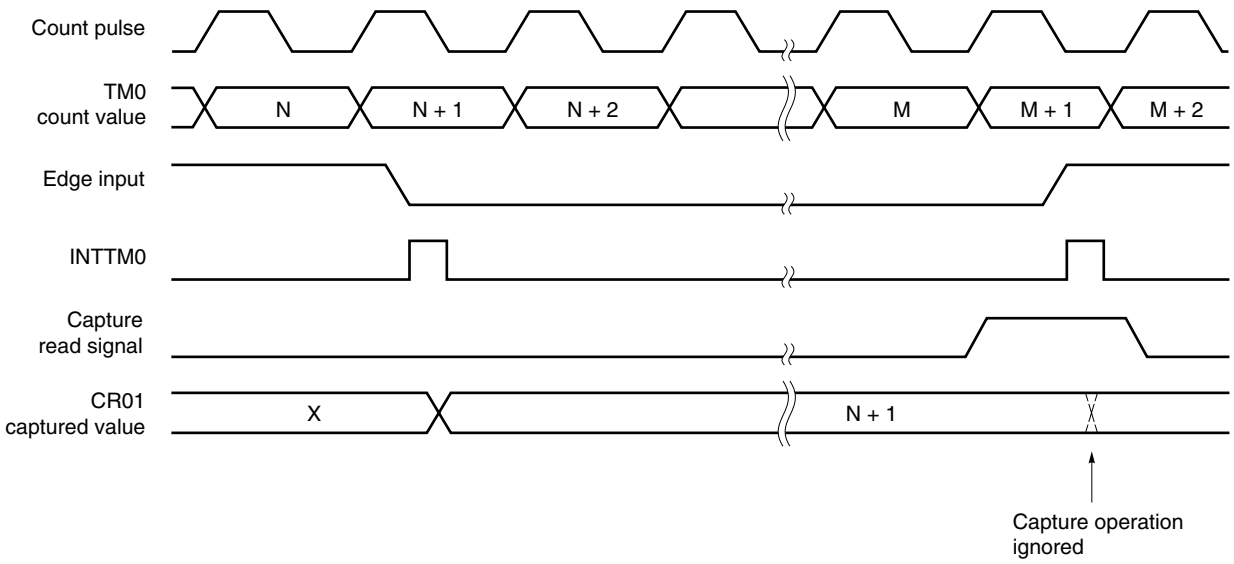


Remark N > X > M

(4) Capture register data retention timing

If the valid edge of the TI0/P00 pin is input during 16-bit capture register (CR01) read, CR01 holds the data without carrying out the capture operation. However, the interrupt request signal (INTTM0) is generated upon detection of the valid edge.

Figure 6-22. Capture Register Data Retention Timing



★ **(5) Valid edge setting**

When using the INTP0/TI0/P00 pin as a timer input pin (TI0), stop the operation of the 16-bit timer by clearing bits 1 to 3 (TMC01 to TMC03) of the 16-bit timer mode control register (TMC0) to 0, 0, 0, before setting the valid edge of TI0. When using the INTP0/TI0/P00 pin as an external interrupt input pin (INTP0), the valid edge of INTP0 may be set while the 16-bit timer is operating.

CHAPTER 7 8-BIT TIMER/EVENT COUNTER

7.1 8-Bit Timer/Event Counter Functions

The following two modes are available for the 8-bit timer/event counter incorporated in the μPD780208 Subseries.

- 8-bit timer/event counter mode: Two-channel 8-bit timer/event counter with each channel used separately
- 16-bit timer/event counter mode: Two-channel 8-bit timer/event counter used as 16-bit timer/event counter

7.1.1 8-bit timer/event counter mode

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

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

(1) 8-bit interval timer

Interrupt requests are generated at the preset time intervals.

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

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

Remarks 1. fx: Main system clock oscillation frequency

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

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

Remarks 1. fx: Main system clock oscillation frequency

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

7.1.2 16-bit timer/event counter mode

(1) 16-bit interval timer

Interrupt requests can be generated at the preset time intervals.

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

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

Remarks 1. fx: Main system clock oscillation frequency

2. 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 7-4. Square-Wave Output Ranges When 8-Bit Timer/Event Counter Is Used as 16-Bit Timer/Event Counter

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

Remarks 1. fx: Main system clock oscillation frequency

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

7.2 8-Bit Timer/Event Counter Configuration

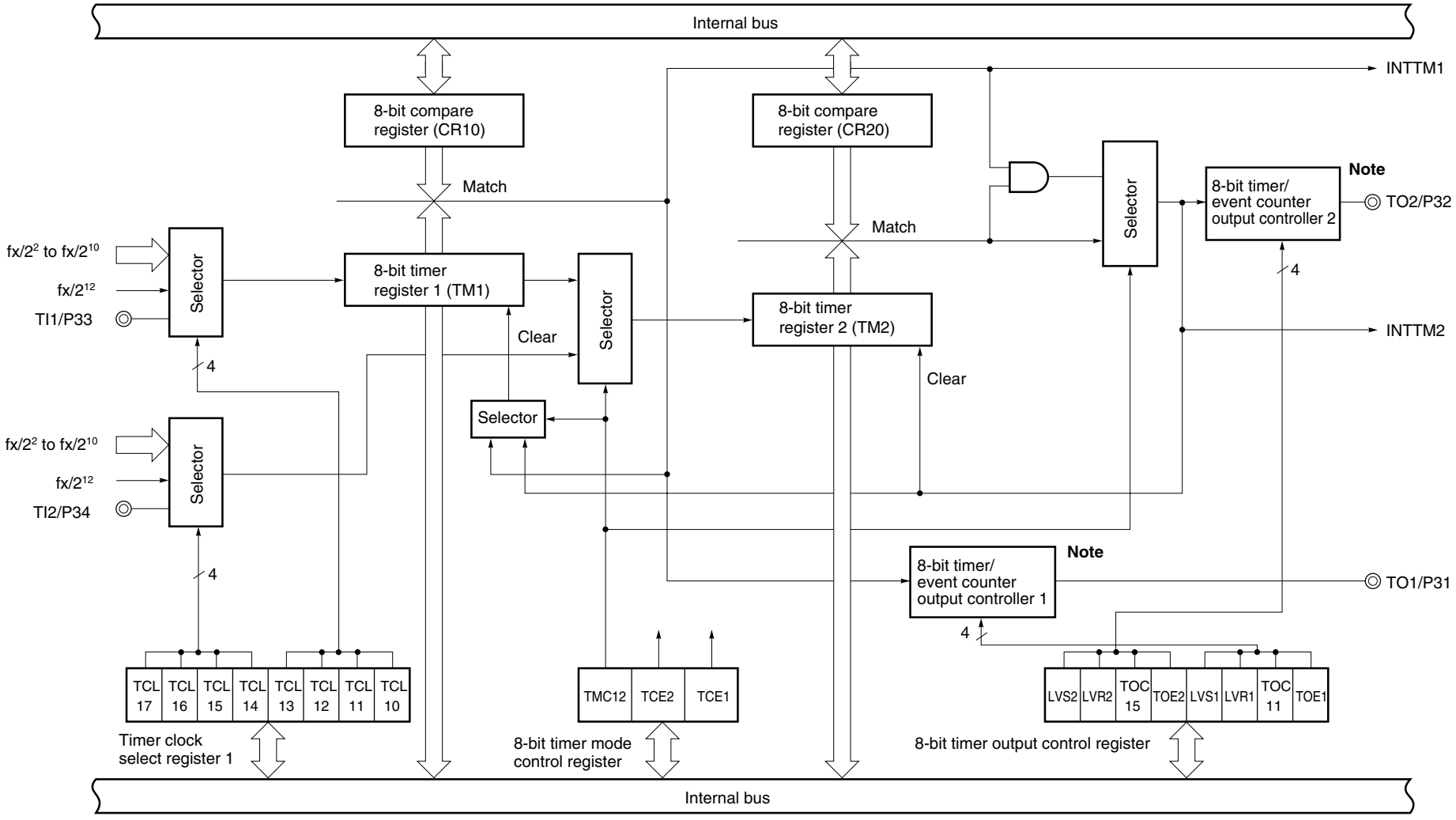
The 8-bit timer/event counter consists of the following hardware.

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

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

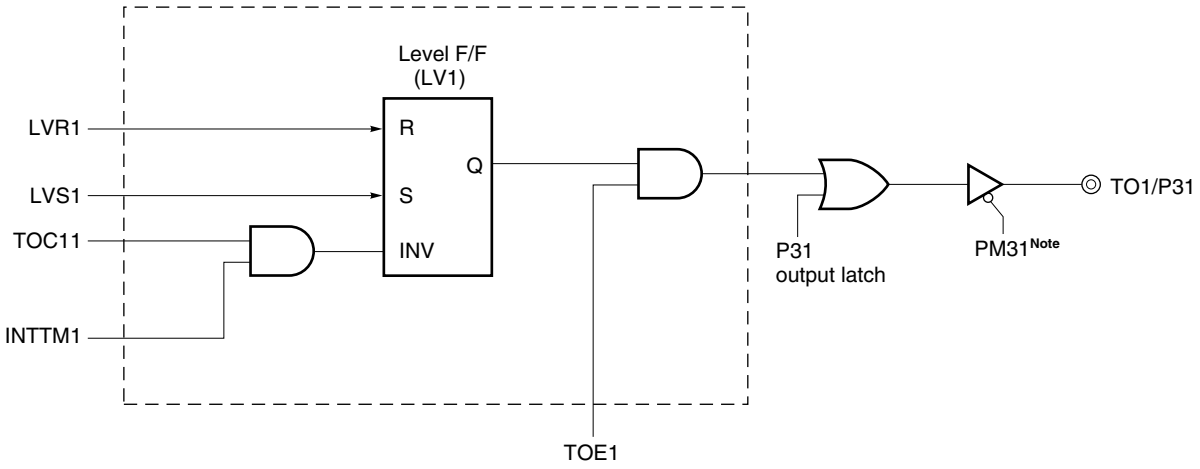
Note Refer to **Figure 4-7 Block Diagram of P30 to P37.**

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



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

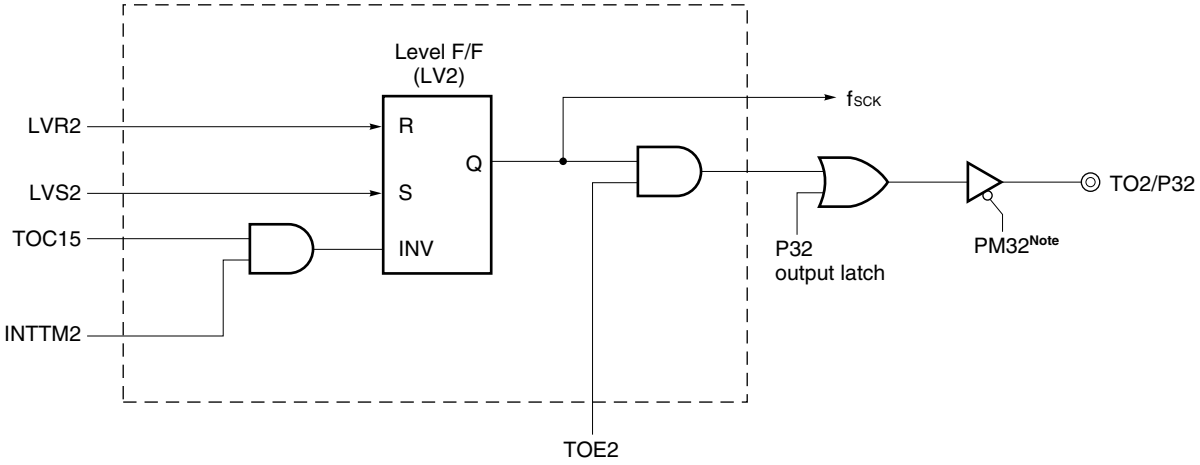
Figure 7-2. Block Diagram of 8-Bit Timer/Event Counter Output Controller 1



Note Bit 1 of port mode register 3 (PM3)

Remark The circuitry enclosed by the dotted line is the output controller.

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



Note Bit 2 of port mode register 3 (PM3)

Remarks 1. The circuitry enclosed by the dotted line is the output controller.

2. f_{SCK}: Serial clock frequency

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

These are 8-bit registers used to compare the value set to CR10 with the 8-bit timer register 1 (TM1) count value, and the value set to CR20 with 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 an 8-bit timer/event counter, values from 00H to FFH can be set. When the compare register is used as a 16-bit timer/event counter, values from 0000H to FFFFH can be set.

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

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

2. When the values of CR10 and CR20 after changing are smaller than those of the 8-bit timer registers (TM1, TM2), TM1 and TM2 continue to count. When they overflow, counting starts again from 0. Therefore, it is necessary to restart the timer after changing the values of CR10 and CR20 if the values of CR10 and CR20 are smaller than the values before changing.

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

These are 8-bit registers used to count count pulses.

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

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

7.3 8-Bit Timer/Event Counter Control Registers

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

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

(1) Timer clock select register 1 (TCL1)

This register sets the count clock 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 7-4. Format of Timer Clock Select Register 1

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

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

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

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

Remarks 1. f_X: Main system clock oscillation frequency

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

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

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

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

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

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

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

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

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

TCE1	8-bit timer register 1 operation control
0	Operation stopped (TM1 cleared to 0)
1	Operation enabled

TCE2	8-bit timer register 2 operation control
0	Operation stopped (TM2 cleared to 0)
1	Operation enabled

TMC12	Operating mode selection
0	8-bit timer register x 2-channel mode (TM1, TM2)
1	16-bit timer register x 1-channel mode (TMS)

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

2. When used as 16-bit timer register (TMS), 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 controllers 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 7-6. Format of 8-Bit Timer Output Control Register

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

TOE1	8-bit timer/event counter 1 output control
0	Output disabled (port mode)
1	Output enabled

TOC11	8-bit timer/event counter 1 timer output F/F control
0	Inverted operation disabled
1	Inverted operation enabled

LVS1	LVR1	8-bit timer/event counter 1 timer output F/F status setting
0	0	Unchanged
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TOE2	8-bit timer/event counter 2 output control
0	Output disabled (port mode)
1	Output enabled

TOC15	8-bit timer/event counter 2 timer output F/F control
0	Inverted operation disabled
1	Inverted operation enabled

LVS2	LVR2	8-bit timer/event counter 2 timer output F/F status setting
0	0	Unchanged
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

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

2. After data setting, 0 is 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 the output latches of P31 and P32 to 0.

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

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

Figure 7-7. Format of Port Mode Register 3

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

PM3n	P3n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

7.4 8-Bit Timer/Event Counter Operations

7.4.1 8-bit timer/event counter mode

(1) Interval timer operations

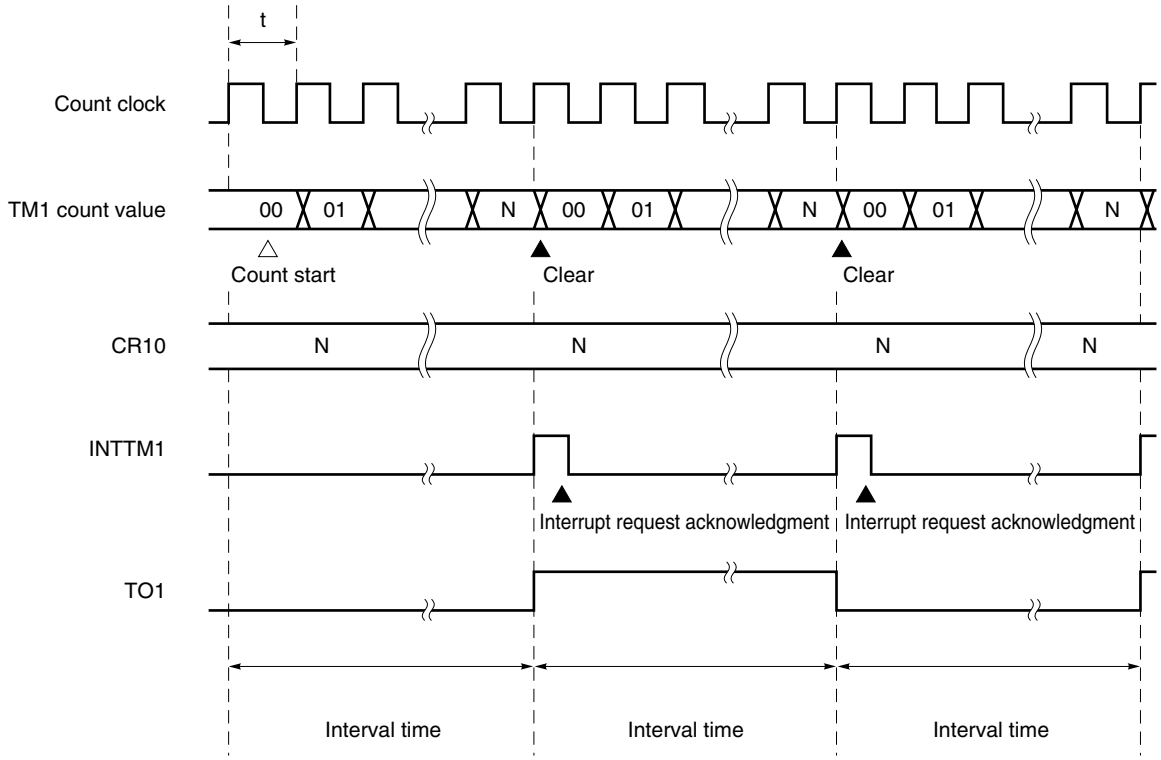
The 8-bit timer/event counter operates as an interval timer that generates interrupt requests repeatedly at intervals of the count value preset to the 8-bit compare registers (CR10 and CR20).

When the count values of 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 interrupt request signals (INTTM1 and INTTM2) are generated.

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

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

Figure 7-8. Interval Timer Operation Timing



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

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

TCL13	TCL12	TCL11	TCL10	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	0	TI1 input cycle	2^8 x TI1 input cycle	TI1 input edge cycle
0	0	0	1	TI1 input cycle	2^8 x TI1 input cycle	TI1 input edge cycle
0	1	0	1	2 x 1/fx (400 ns)	2^9 x 1/fx (102.4 μs)	2 x 1/fx (400 ns)
0	1	1	0	2^2 x 1/fx (800 ns)	2^{10} x 1/fx (204.8 μs)	2^2 x 1/fx (800 ns)
0	1	1	1	2^3 x 1/fx (1.6 μs)	2^{11} x 1/fx (409.6 μs)	2^3 x 1/fx (1.6 μs)
1	0	0	0	2^4 x 1/fx (3.2 μs)	2^{12} x 1/fx (819.2 μs)	2^4 x 1/fx (3.2 μs)
1	0	0	1	2^5 x 1/fx (6.4 μs)	2^{13} x 1/fx (1.64 ms)	2^5 x 1/fx (6.4 μs)
1	0	1	0	2^6 x 1/fx (12.8 μs)	2^{14} x 1/fx (3.28 ms)	2^6 x 1/fx (12.8 μs)
1	0	1	1	2^7 x 1/fx (25.6 μs)	2^{15} x 1/fx (6.55 ms)	2^7 x 1/fx (25.6 μs)
1	1	0	0	2^8 x 1/fx (51.2 μs)	2^{16} x 1/fx (13.1 ms)	2^8 x 1/fx (51.2 μs)
1	1	0	1	2^9 x 1/fx (102.4 μs)	2^{17} x 1/fx (26.2 ms)	2^9 x 1/fx (102.4 μs)
1	1	1	0	2^{10} x 1/fx (204.8 μs)	2^{18} x 1/fx (52.4 ms)	2^{10} x 1/fx (204.8 μs)
1	1	1	1	2^{12} x 1/fx (819.2 μs)	2^{20} x 1/fx (209.7 ms)	2^{12} x 1/fx (819.2 μs)
Other than above				Setting prohibited		

Remarks 1. fx: Main system clock oscillation frequency

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

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

TCL17	TCL16	TCL15	TCL14	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	0	TI2 input cycle	2^8 x TI2 input cycle	TI2 input edge cycle
0	0	0	1	TI2 input cycle	2^8 x TI2 input cycle	TI2 input edge cycle
0	1	0	1	2 x 1/fx (400 ns)	2^9 x 1/fx (102.4 μs)	2 x 1/fx (400 ns)
0	1	1	0	2^2 x 1/fx (800 ns)	2^{10} x 1/fx (204.8 μs)	2^2 x 1/fx (800 ns)
0	1	1	1	2^3 x 1/fx (1.6 μs)	2^{11} x 1/fx (409.6 μs)	2^3 x 1/fx (1.6 μs)
1	0	0	0	2^4 x 1/fx (3.2 μs)	2^{12} x 1/fx (819.2 μs)	2^4 x 1/fx (3.2 μs)
1	0	0	1	2^5 x 1/fx (6.4 μs)	2^{13} x 1/fx (1.64 ms)	2^5 x 1/fx (6.4 μs)
1	0	1	0	2^6 x 1/fx (12.8 μs)	2^{14} x 1/fx (3.28 ms)	2^6 x 1/fx (12.8 μs)
1	0	1	1	2^7 x 1/fx (25.6 μs)	2^{15} x 1/fx (6.55 ms)	2^7 x 1/fx (25.6 μs)
1	1	0	0	2^8 x 1/fx (51.2 μs)	2^{16} x 1/fx (13.1 ms)	2^8 x 1/fx (51.2 μs)
1	1	0	1	2^9 x 1/fx (102.4 μs)	2^{17} x 1/fx (26.2 ms)	2^9 x 1/fx (102.4 μs)
1	1	1	0	2^{10} x 1/fx (204.8 μs)	2^{18} x 1/fx (52.4 ms)	2^{10} x 1/fx (204.8 μs)
1	1	1	1	2^{12} x 1/fx (819.2 μs)	2^{20} x 1/fx (209.7 ms)	2^{12} x 1/fx (819.2 μs)
Other than above				Setting prohibited		

Remarks 1. fx: Main system clock oscillation frequency

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

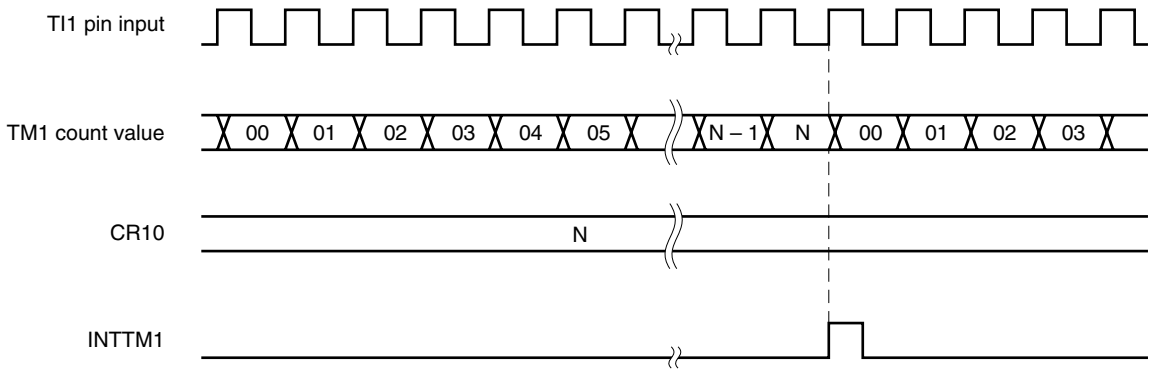
(2) External event counter operation

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

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

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

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



Remark N = 00H to FFH

(3) Square-wave output operation

The 8-bit timer/event counter outputs a square wave of any frequency with the value preset to the 8-bit compare register (CR10, CR20) as the interval.

The TO1/P31 or TO2/P32 pin output status is reversed at intervals of the count value preset to CR10 or CR20 by setting bit 0 (TOE1) or bit 4 (TOE2) of the 8-bit timer output control register (TOC1) to 1. This enables a square wave with any selected frequency to be output.

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

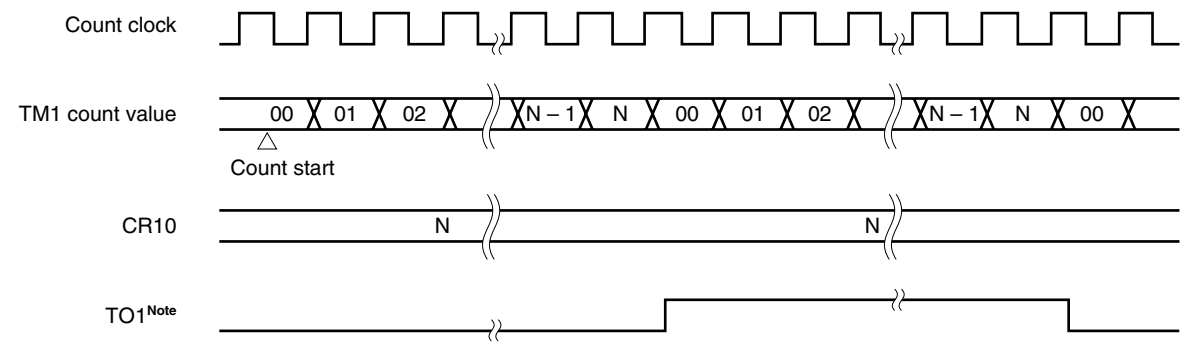
TCL13	TCL12	TCL11	TCL10	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	1	0	1	2 x 1/fx (400 ns)	2^9 x 1/fx (102.4 μs)	2 x 1/fx (400 ns)
0	1	1	0	2^2 x 1/fx (800 ns)	2^{10} x 1/fx (204.8 μs)	2^2 x 1/fx (800 ns)
0	1	1	1	2^3 x 1/fx (1.6 μs)	2^{11} x 1/fx (409.6 μs)	2^3 x 1/fx (1.6 μs)
1	0	0	0	2^4 x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

Remarks 1. fx: Main system clock oscillation frequency

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

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

Figure 7-10. Square-Wave Output Operation Timing



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

7.4.2 16-bit timer/event counter mode

When bit 2 (TMC12) of 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 using bits 0 to 3 (TCL10 to TCL13) of the timer clock select register (TCL1). The overflow signal of 8-bit timer/event counter 1 (TM1) is used as the count clock of 8-bit timer/event counter 2 (TM2).

Count operation enable/disable in this mode is selected using bit 0 (TCE1) of TMC1.

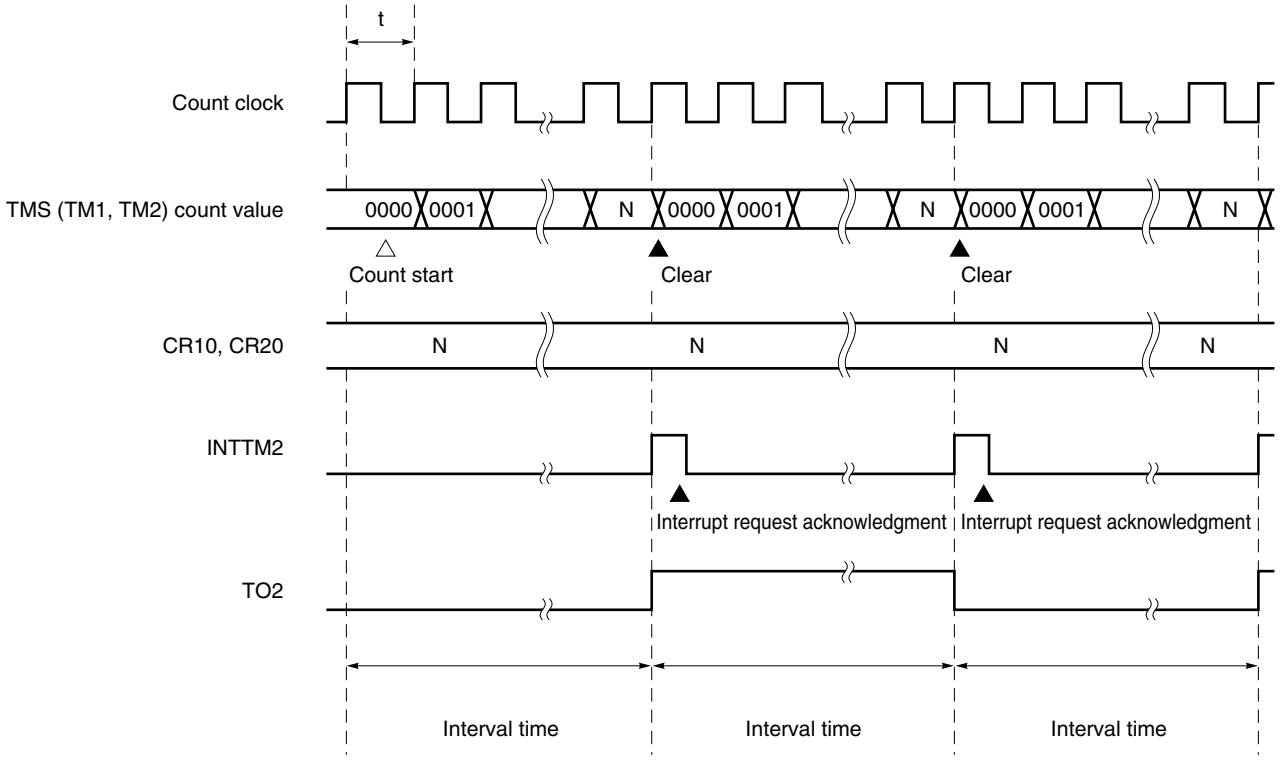
(1) Interval timer operations

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

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

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

Figure 7-11. Interval Timer Operation Timing



Remark Interval time = (N + 1) x 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, an interrupt request (INTTM1) is generated and the F/F of 8-bit timer/event counter output controller 1 is inverted. Thus, when using the 8-bit timer/event counter as a 16-bit interval timer, set the mask flag TMMK1 to 1 to disable INTTM1 acknowledgment.
When reading the 16-bit timer register (TMS) count value, use a 16-bit memory manipulation instruction.

Table 7-9. Interval Time When 2-Channel 8-Bit Timer/Event Counter (TM1 and TM2) Is Used as 16-Bit Timer/Event Counter

TCL13	TCL12	TCL11	TCL10	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	0	TI1 input cycle	2^8 x TI1 input cycle	TI1 input edge cycle
0	0	0	1	TI1 input cycle	2^8 x TI1 input cycle	TI1 input edge cycle
0	1	0	1	2 x 1/f_X (400 ns)	2^{17} x 1/f_X (26.2 ms)	2 x 1/f_X (400 ns)
0	1	1	0	2^2 x 1/f_X (800 ns)	2^{18} x 1/f_X (52.4 ms)	2^2 x 1/f_X (800 ns)
0	1	1	1	2^3 x 1/f_X (1.6 μs)	2^{19} x 1/f_X (104.9 ms)	2^3 x 1/f_X (1.6 μs)
1	0	0	0	2^4 x 1/f_X (3.2 μs)	2^{20} x 1/f_X (209.7 ms)	2^4 x 1/f_X (3.2 μs)
1	0	0	1	2^5 x 1/f_X (6.4 μs)	2^{21} x 1/f_X (419.4 ms)	2^5 x 1/f_X (6.4 μs)
1	0	1	0	2^6 x 1/f_X (12.8 μs)	2^{22} x 1/f_X (838.9 ms)	2^6 x 1/f_X (12.8 μs)
1	0	1	1	2^7 x 1/f_X (25.6 μs)	2^{23} x 1/f_X (1.7 s)	2^7 x 1/f_X (25.6 μs)
1	1	0	0	2^8 x 1/f_X (51.2 μs)	2^{24} x 1/f_X (3.4 s)	2^8 x 1/f_X (51.2 μs)
1	1	0	1	2^9 x 1/f_X (102.4 μs)	2^{25} x 1/f_X (6.7 s)	2^9 x 1/f_X (102.4 μs)
1	1	1	0	2^{10} x 1/f_X (204.8 μs)	2^{26} x 1/f_X (13.4 s)	2^{10} x 1/f_X (204.8 μs)
1	1	1	1	2^{12} x 1/f_X (819.2 μs)	2^{28} x 1/f_X (53.7 s)	2^{12} x 1/f_X (819.2 μs)
Other than above				Setting prohibited		

Remarks 1. f_X: Main system clock oscillation frequency
2. Figures in parentheses apply to operation with f_X = 5.0 MHz.

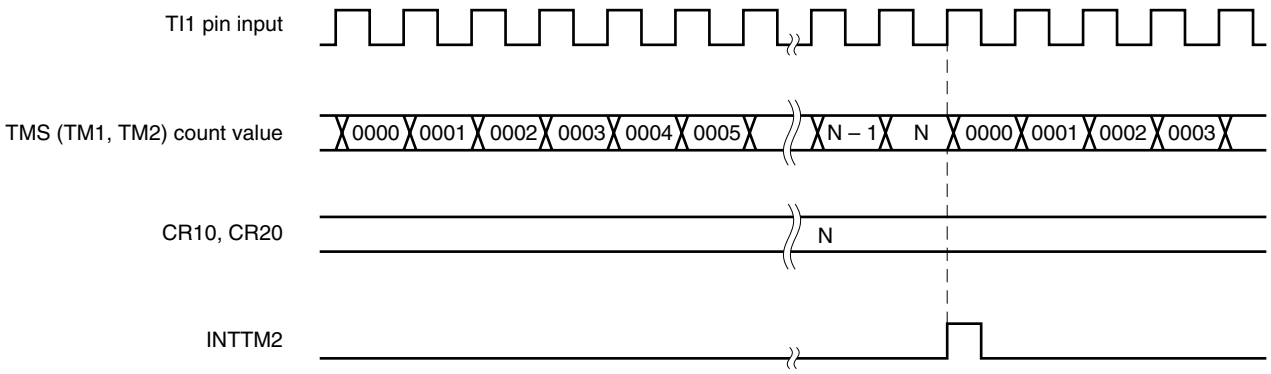
(2) External event counter operations

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

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

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

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



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

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

(3) Square-wave output operation

Square-wave signals can be generated at the user-specified frequency. The frequency or pulse interval is determined by the value preset in the 8-bit compare registers (CR10 and CR20). To set a count value, set the value of the higher 8 bits to CR20, and the value 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 7-10. Square-Wave Output Ranges When 2-Channel 8-Bit Timer/Event Counters (TM1 and TM2) Are Used as 16-Bit Timer/Event Counter

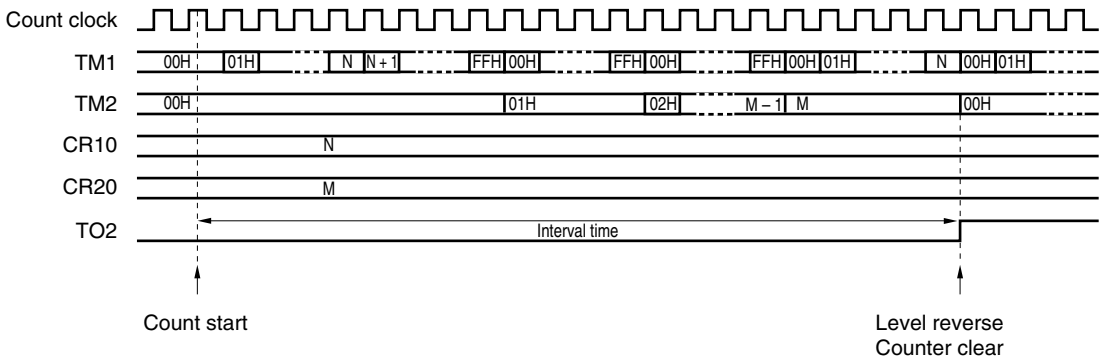
TCL13	TCL12	TCL11	TCL10	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	1	0	1	2 x 1/fx (400 ns)	2^{17} x 1/fx (26.2 ms)	2 x 1/fx (400 ns)
0	1	1	0	2^{2} x 1/fx (800 ns)	2^{18} x 1/fx (52.4 ms)	2^{2} x 1/fx (800 ns)
0	1	1	1	2^{3} x 1/fx (1.6 μs)	2^{19} x 1/fx (104.9 ms)	2^{3} x 1/fx (1.6 μs)
1	0	0	0	2^{4} x 1/fx (3.2 μs)	2^{20} x 1/fx (209.7 ms)	2^{4} x 1/fx (3.2 μs)
1	0	0	1	2^{5} x 1/fx (6.4 μs)	2^{21} x 1/fx (419.4 ms)	2^{5} x 1/fx (6.4 μs)
1	0	1	0	2^{6} x 1/fx (12.8 μs)	2^{22} x 1/fx (838.9 ms)	2^{6} x 1/fx (12.8 μs)
1	0	1	1	2^{7} x 1/fx (25.6 μs)	2^{23} x 1/fx (1.7 s)	2^{7} x 1/fx (25.6 μs)
1	1	0	0	2^{8} x 1/fx (51.2 μs)	2^{24} x 1/fx (3.4 s)	2^{8} x 1/fx (51.2 μs)
1	1	0	1	2^{9} x 1/fx (102.4 μs)	2^{25} x 1/fx (6.7 s)	2^{9} x 1/fx (102.4 μs)
1	1	1	0	2^{10} x 1/fx (204.8 μs)	2^{26} x 1/fx (13.4 s)	2^{10} x 1/fx (204.8 μs)
1	1	1	1	2^{12} x 1/fx (819.2 μs)	2^{28} x 1/fx (53.7 s)	2^{12} x 1/fx (819.2 μs)

Remarks 1. fx: Main system clock oscillation frequency

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

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

Figure 7-13. Square-Wave Output Operation Timing

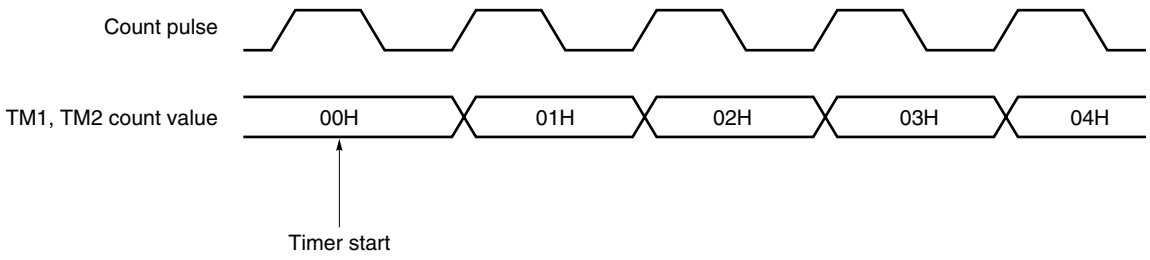


7.5 8-Bit Timer/Event Counter Operating Precautions

(1) Timer start error

An error of up to one clock may occur in 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 to the count pulse.

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



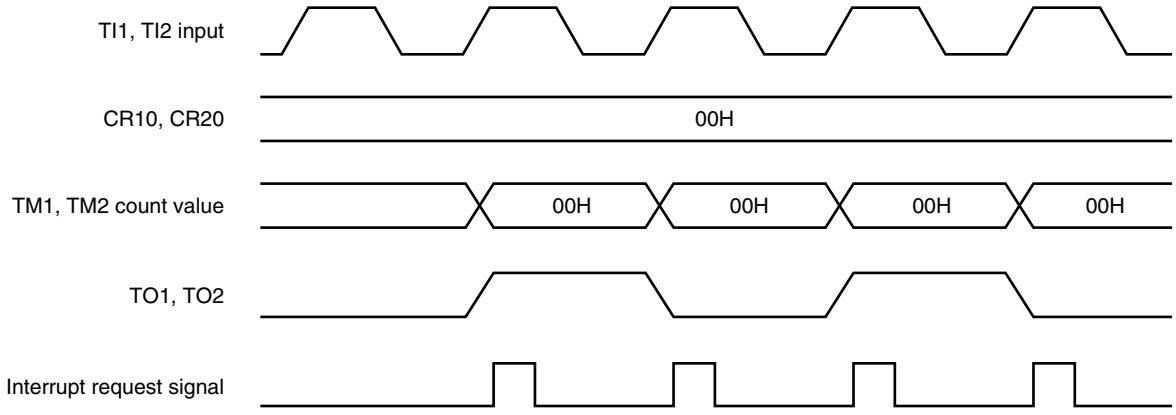
(2) 8-bit compare registers 1 and 2 settings

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

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

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

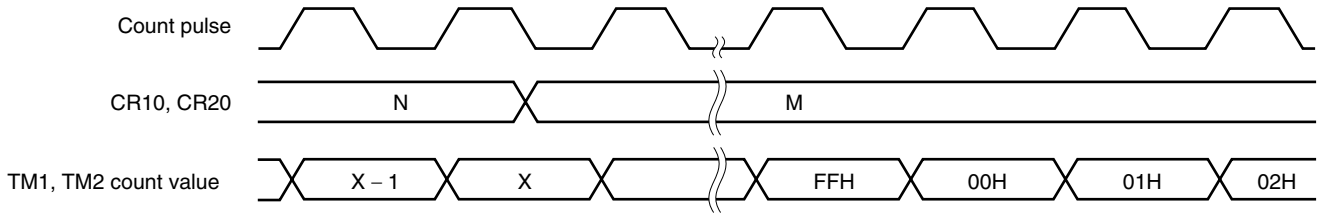
Figure 7-15. External Event Counter Operation Timing



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

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

Figure 7-16. Timing After Compare Register Change During Timer Count Operation



Remark N > X > M

CHAPTER 8 WATCH TIMER

8.1 Watch Timer Functions

The watch timer has the following functions.

- Watch timer
- Interval timer

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

(1) Watch timer

When the 32.768 kHz subsystem clock is used, a flag (WTIF) is set at 0.5-second or 0.25-second intervals. In addition, when the 4.19 MHz (standard: 4.194304 MHz) main system clock is used, a flag (WTIF) is set at 0.5-second or 1-second intervals.

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

(2) Interval timer

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

Table 8-1. Interval Timer Interval Time

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

f_X: Main system clock oscillation frequency
f_{XT}: Subsystem clock oscillation frequency

8.2 Watch Timer Configuration

The watch timer consists of the following hardware.

Table 8-2. Watch Timer Configuration

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

8.3 Watch Timer Control Registers

The following two 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) (See **Figure 8-2**)

This register sets the watch timer count clock.

TCL2 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TCL2 to 00H.

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

Figure 8-1. Watch Timer Block Diagram

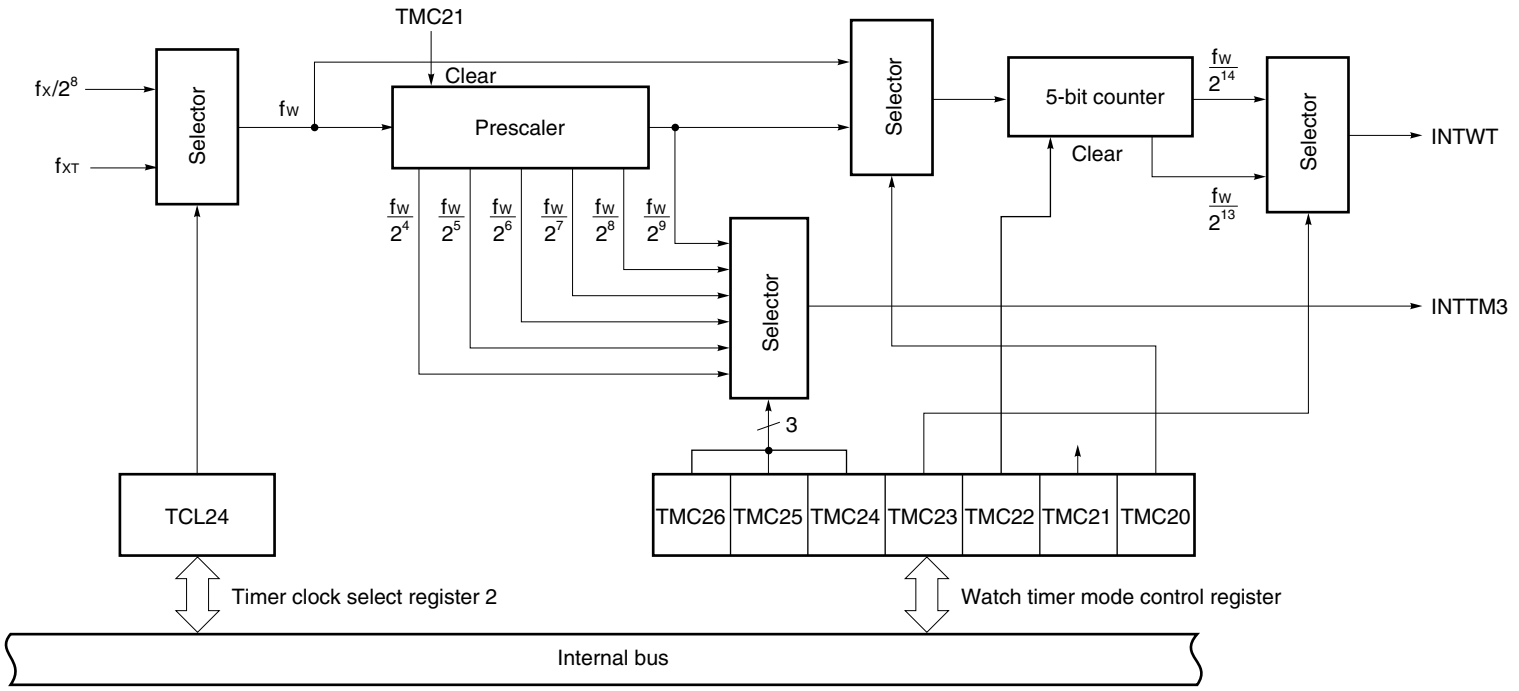


Figure 8-2. Format of Timer Clock Select Register 2

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

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

TCL24	Watch timer count clock selection[Note]
0	$f_X/2^8$ (19.5 kHz)
1	f_{XT} (32.768 kHz)

TCL27	TCL26	TCL25	Buzzer output frequency selection
0	×	×	Buzzer output disabled
1	0	0	$f_X/2^{10}$ (4.9 kHz)
1	0	1	$f_X/2^{11}$ (2.4 kHz)
1	1	0	$f_X/2^{12}$ (1.2 kHz)
1	1	1	Setting prohibited

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

★ **Caution When changing the count clock, be sure to stop operation of the watch timer before rewriting TCL2 (stopping operation is not necessary when rewriting the same data).**

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. x: don't care

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

(2) Watch timer mode control register (TMC2)

This register sets the watch timer operating mode, watch flag set time and prescaler interval time and enables/disables prescaler and 5-bit counter operations.

TMC2 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 8-3. Format of Watch Timer Mode Control Register

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

TMC23	TMC20	Watch flag set time selection
0	0	2^{14}/fw (0.5 s)
1		2^{13}/fw (0.25 s)
0	1	2^{5}/fw (977 μs)
1		2^{4}/fw (488 μs)

TMC21	Prescaler operation control[Note]
0	Clear after operation stops
1	Operation enable

TMC22	5-bit counter operation control
0	Clear after operation stops
1	Operation enable

TMC26	TMC25	TMC24	Prescaler interval time selection
0	0	0	2^{4}/fw (488 μs)
0	0	1	2^{5}/fw (977 μs)
0	1	0	2^{6}/fw (1.95 ms)
0	1	1	2^{7}/fw (3.91 ms)
1	0	0	2^{8}/fw (7.81 ms)
1	0	1	2^{9}/fw (15.6 ms)
Other than above			Setting prohibited

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

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

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

8.4 Watch Timer Operations

8.4.1 Watch timer operation

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

The watch timer sets the test input flag (WTIF) to 1 at a 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 1 again after setting TMC22 to 0 (maximum error: 26.2 ms when operated at 5.0 MHz).

8.4.2 Interval timer operation

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

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

Table 8-3. Interval Timer Interval Time

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

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

CHAPTER 9 WATCHDOG TIMER

9.1 Watchdog Timer Functions

The watchdog timer has the following functions.

- Watchdog timer
- Interval timer

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

(1) Watchdog timer mode

This mode detects an inadvertent program loop. Upon detection of the program loop, a non-maskable interrupt request or $\overline{\text{RESET}}$ can be generated.

Table 9-1. Watchdog Timer Program Loop Detection Time

Program Loop Detection Time	When Operated at f_X = 5.0 MHz	Program Loop Detection Time	When Operated at f_X = 5.0 MHz
$2^{11} \times 1/f_X$	410 μs	$2^{15} \times 1/f_X$	6.55 ms
$2^{12} \times 1/f_X$	819 μs	$2^{16} \times 1/f_X$	13.1 ms
$2^{13} \times 1/f_X$	1.64 ms	$2^{17} \times 1/f_X$	26.2 ms
$2^{14} \times 1/f_X$	3.28 ms	$2^{19} \times 1/f_X$	104.9 ms

fx: Main system clock oscillation frequency

(2) Interval timer mode

Interrupt requests are generated at the preset time intervals.

Table 9-2. Interval Time

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

fx: Main system clock oscillation frequency

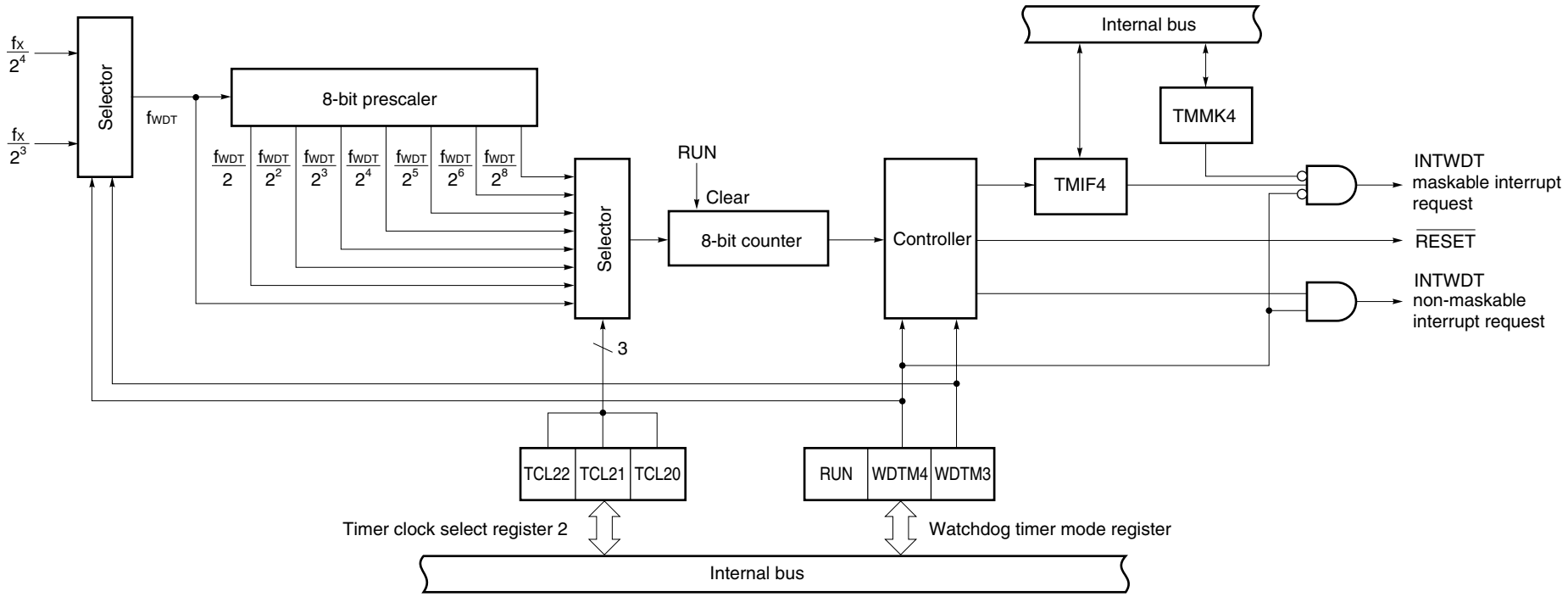
9.2 Watchdog Timer Configuration

The watchdog timer consists of the following hardware.

Table 9-3. Watchdog Timer Configuration

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

Figure 9-1. Watchdog Timer Block Diagram



9.3 Watchdog Timer Control Registers

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

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

(1) Timer clock select register 2 (TCL2)

This register sets the watchdog timer count clock.

TCL2 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TCL2 to 00H.

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

Figure 9-2. Format of Timer Clock Select Register 2

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

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

TCL24	Watch timer count clock selection[Note]
0	$f_X/2^8$ (19.5 kHz)
1	f_{XT} (32.768 kHz)

TCL27	TCL26	TCL25	Buzzer output frequency selection
0	x	x	Buzzer output disabled
1	0	0	$f_X/2^{10}$ (4.9 kHz)
1	0	1	$f_X/2^{11}$ (2.4 kHz)
1	1	0	$f_X/2^{12}$ (1.2 kHz)
1	1	1	Setting prohibited

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

★ **Caution Changing the count clock (rewriting TCL20 to TCL22) after watchdog timer operation has started is prohibited.**

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. x: don't care

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

(2) Watchdog timer mode register (WDTM)

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

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

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

Figure 9-3. Format of Watchdog Timer Mode Register

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0	FFF9H	00H	R/W

WDTM4	WDTM3	Watchdog timer operating mode selection[Note 1]
0	x	Interval timer mode[Note 2] (Maskable interrupt request occurs upon generation of an overflow.)
1	0	Watchdog timer mode 1 (Non-maskable interrupt request occurs upon generation of an overflow.)
1	1	Watchdog timer mode 2 (Reset operation is activated upon generation of an overflow.)

RUN	Watchdog timer operation selection[Note 3]
0	Count stop
1	Counter is cleared and counting starts.

Notes 1. Once set to 1, WDTM3 and WDTM4 cannot be cleared to 0 by software.

2. Starts operation as an interval timer as soon as RUN is set to 1.

3. Once set to 1, RUN cannot be cleared to 0 by software. Thus, once counting starts, it can only be stopped by $\overline{\text{RESET}}$ input.

Cautions 1. When RUN is set to 1 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).

2. When using watchdog timer mode 1 and 2, make sure that the interrupt request flag (TMIF4) is set to 0 before setting WDTM4 to 1. If WDTM4 is set to 1 while TMIF4 is 1, a non-maskable interrupt request occurs regardless of the contents of WDTM3.

Remark x: don't care

9.4 Watchdog Timer Operations

9.4.1 Watchdog timer operation

When bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 1, the watchdog timer operates to detect an inadvertent program loop.

The watchdog timer count clock (program loop detection time interval) can be selected using bits 0 to 2 (TCL20 to TCL22) of timer clock select register 2 (TCL2).

The 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 program loop 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 program loop detection time elapses, a system reset or a non-maskable interrupt request is generated according to the value of WDTM bit 3 (WDTM3).

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

2. When the subsystem clock is selected for the CPU clock, the watchdog timer count operation is stopped.

Table 9-4. Watchdog Timer Program Loop Detection Time

TCL22	TCL21	TCL20	Program Loop Detection Time	f_X = 5.0 MHz
0	0	0	$2^{11} \times 1/f_X$	410 μs
0	0	1	$2^{12} \times 1/f_X$	819 μs
0	1	0	$2^{13} \times 1/f_X$	1.64 ms
0	1	1	$2^{14} \times 1/f_X$	3.28 ms
1	0	0	$2^{15} \times 1/f_X$	6.55 ms
1	0	1	$2^{16} \times 1/f_X$	13.1 ms
1	1	0	$2^{17} \times 1/f_X$	26.2 ms
1	1	1	$2^{19} \times 1/f_X$	105.0 ms

fx: Main system clock oscillation frequency

9.4.2 Interval timer operation

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

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

When the watchdog timer operates as an interval timer, the interrupt mask flag (TMMK4) and priority specification flag (TMPR4) are validated and a maskable interrupt request (INTWDT) can be generated. Among the maskable interrupt requests, INTWDT has the highest default priority.

The interval timer continues operating in the HALT mode but it stops in the STOP mode. Thus, set RUN to 1 before the STOP mode is set, clear the interval timer and then execute the STOP instruction.

Cautions 1. Once bit 4 (WDTM4) of WDTM is set to 1 (with the watchdog timer mode selected), the interval timer mode is not set unless $\overline{\text{RESET}}$ input is applied.

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

3. When the subsystem clock is selected for the CPU clock, the watchdog timer count operation is stopped.

Table 9-5. Interval Timer Interval Time

TCL22	TCL21	TCL20	Interval Time	f_X = 5.0 MHz
0	0	0	$2^{12} \times 1/f_X$	819 μs
0	0	1	$2^{13} \times 1/f_X$	1.64 ms
0	1	0	$2^{14} \times 1/f_X$	3.28 ms
0	1	1	$2^{15} \times 1/f_X$	6.55 ms
1	0	0	$2^{16} \times 1/f_X$	13.1 ms
1	0	1	$2^{17} \times 1/f_X$	26.2 ms
1	1	0	$2^{18} \times 1/f_X$	52.4 ms
1	1	1	$2^{20} \times 1/f_X$	210.0 ms

f_X: Main system clock oscillation frequency

CHAPTER 10 CLOCK OUTPUT CONTROLLER

10.1 Clock Output Controller Functions

The clock output controller is used for carrier output during remote controlled transmission and clock output for supply to a peripheral LSI. The clock selected by timer clock select register 0 (TCL0) is output from the PCL/P35 pin.

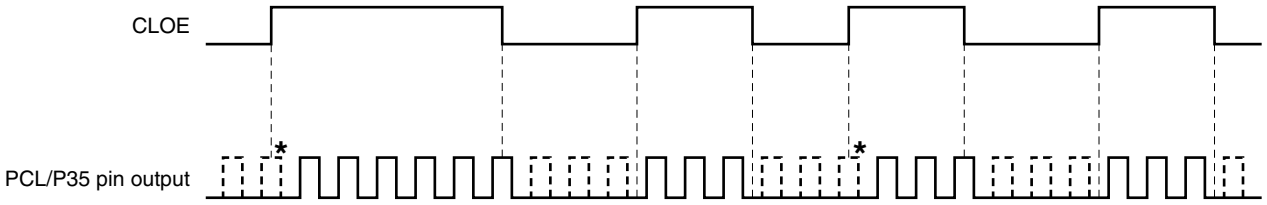
Follow the procedure below to output clock pulses.

[1] Select the clock pulse output frequency (with clock pulse output disabled) using bits 0 to 3 (TCL00 to TCL03) of TCL0.
[2] Set the P35 output latch to 0.
[3] Set bit 5 (PM35) of port mode register 3 (PM3) to 0 (set to output mode).
[4] Set bit 7 (CLOE) of TCL0 to 1.

Caution Clock output cannot be used when the P35 output latch is set to 1.

Remark When clock output enable/disable is switched, the clock output controller does not output pulses with small widths (see the mark * in **Figure 10-1**).

Figure 10-1. Remote Controlled Output Application Example



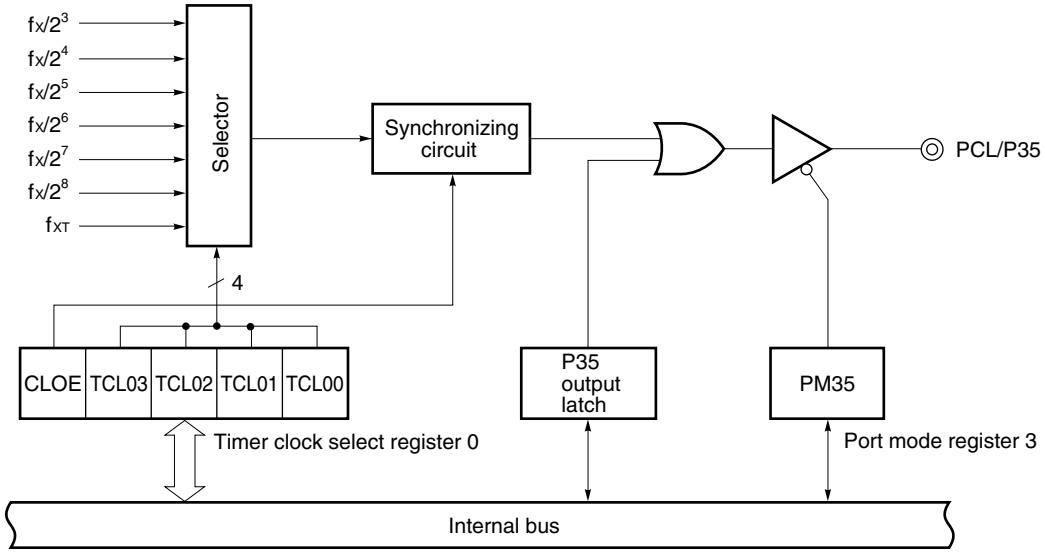
10.2 Clock Output Controller Configuration

The clock output controller consists of the following hardware.

Table 10-1. Clock Output Controller Configuration

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

Figure 10-2. Clock Output Controller Block Diagram



10.3 Clock Output Function Control Registers

The following two 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 the PCL output clock.

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

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

Remark Besides setting the PCL output clock, TCL0 sets the 16-bit timer register count clock.

Figure 10-3. Format of Timer Clock Select Register 0

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

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

TCL06	TCL05	TCL04	16-bit timer register count clock selection
0	0	0	TI0 (valid edge specifiable)
0	0	1	f_X (5.0 MHz)
0	1	0	$f_X/2$ (2.5 MHz)
0	1	1	$f_X/2^2$ (1.25 MHz)
1	0	0	$f_X/2^3$ (625 kHz)
Other than above			Setting prohibited

CLOE	PCL output control
0	Output disabled
1	Output enabled

Cautions 1. The TI0/P00/INTP0 pin valid edge is set by the external interrupt mode register (INTM0), and the sampling clock frequency is selected by the sampling clock select register (SCS).

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

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

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

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

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

4. TM0: 16-bit timer register

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

(2) Port mode register 3 (PM3)

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

When using the P35/PCL pin for clock output, set PM35 and the output latch of P35 to 0.

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

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

Figure 10-4. Format of Port Mode Register 3

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

PM3n	P3n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

CHAPTER 11 BUZZER OUTPUT CONTROLLER

11.1 Buzzer Output Controller Functions

The buzzer output controller outputs a 1.2 kHz, 2.4 kHz, or 4.9 kHz frequency square-wave. The buzzer frequency selected by 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 using 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 the P36 output latch is set to 1.

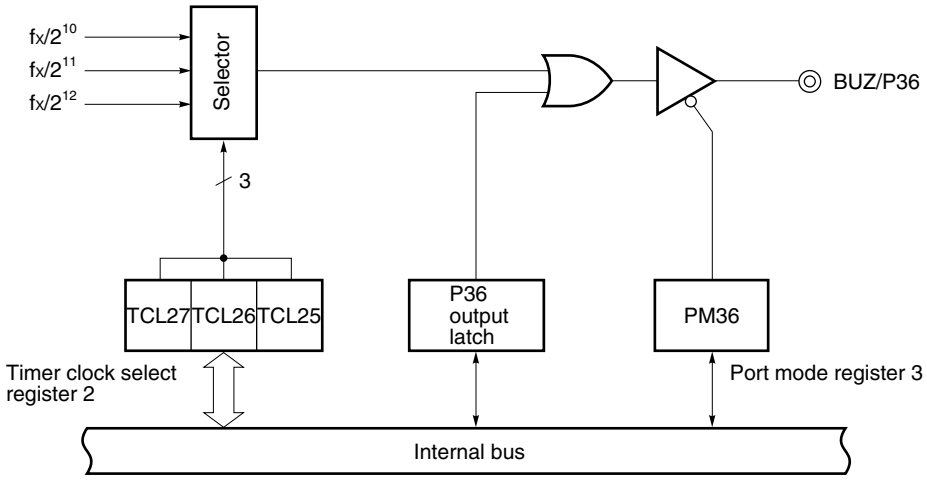
11.2 Buzzer Output Controller Configuration

The buzzer output controller consists of the following hardware.

Table 11-1. Buzzer Output Controller Configuration

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

Figure 11-1. Buzzer Output Controller Block Diagram



11.3 Buzzer Output Function Control Registers

The following two 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 11-2. Format of Timer Clock Select Register 2

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

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

TCL24	Watch timer count clock selection
0	$f_X/2^8$ (19.5 kHz)
1	f_{XT} (32.768 kHz)

TCL27	TCL26	TCL25	Buzzer output frequency selection
0	x	x	Buzzer output disabled
1	0	0	$f_X/2^{10}$ (4.9 kHz)
1	0	1	$f_X/2^{11}$ (2.4 kHz)
1	1	0	$f_X/2^{12}$ (1.2 kHz)
1	1	1	Setting prohibited

★ **Cautions 1. Be sure to stop operation of the watch timer or buzzer to be changed before rewriting TCL2 (stopping operation is not necessary when rewriting the same data).**
The operation is stopped by the following methods.

- **Buzzer output: Input 0 to bit 7 (TCL27) of TCL2**
- **Watch timer: Input 0 to bit 2 (TMC22) of the watch timer mode control register (TMC2)**

★ **2. Changing the count clock (rewriting TCL20 to TCL22) after watchdog timer operation has started is prohibited.**

Remarks 1. f_X: Main system clock oscillation frequency
2. f_{XT}: Subsystem clock oscillation frequency
3. x: don't care
4. Figures in parentheses apply to operation with f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

(2) Port mode register 3 (PM3)

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

When using the P36/BUZ pin for buzzer output, set PM36 and the output latch of P36 to 0.

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

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

Figure 11-3. Format of Port Mode Register 3

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

PM3n	P3n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

CHAPTER 12 A/D CONVERTER

12.1 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).

A/D conversion can be started in the following two ways.

(1) Hardware start

Conversion is started by trigger input (INTP3).

(2) Software start

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

Select one channel of analog input from ANI0 to ANI7 and carry out A/D conversion. In the case of a hardware start, when A/D conversion finishes, the A/D converter stops and an interrupt request (INTAD) is generated. In the case of a software start, the A/D conversion operation is repeated. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

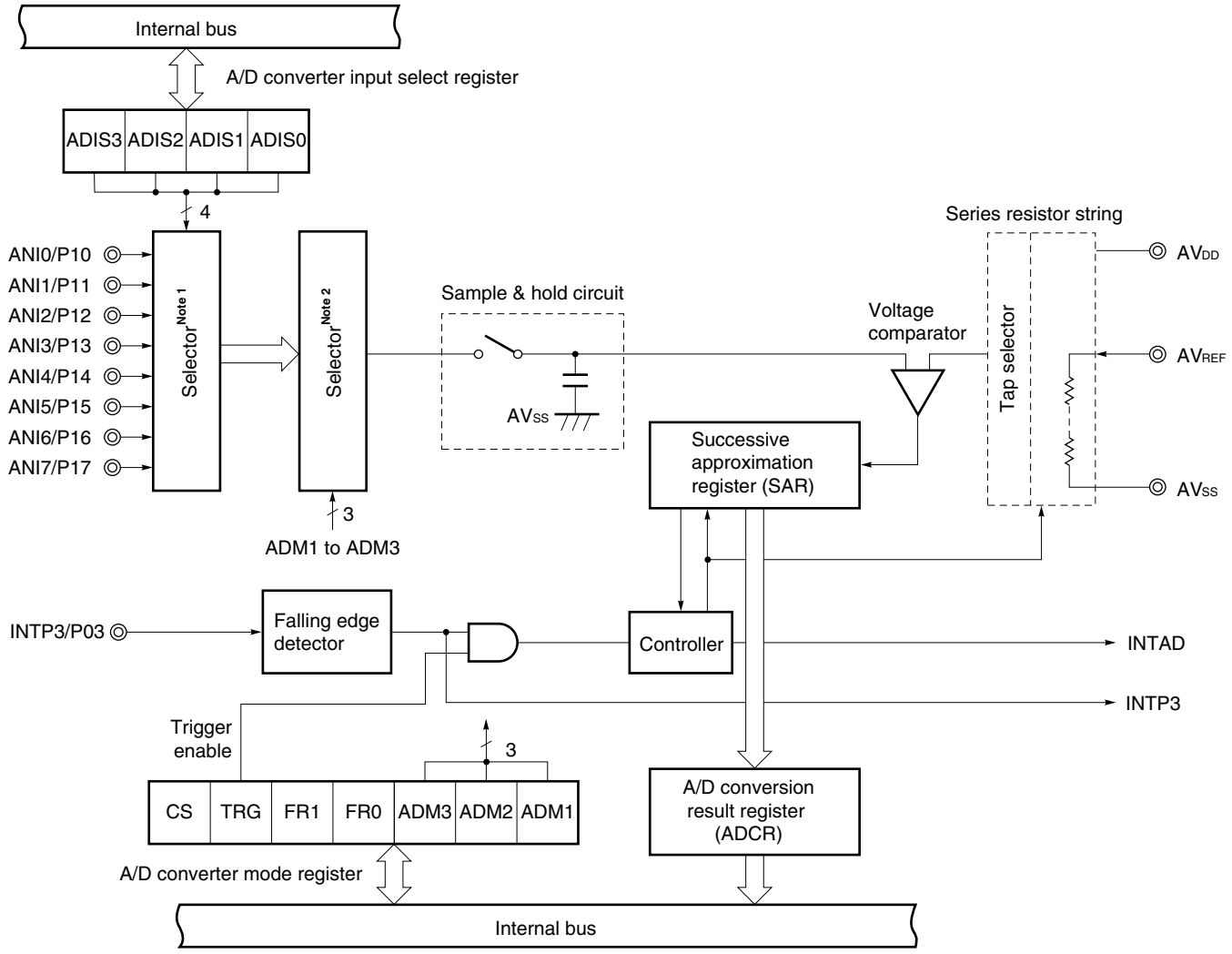
12.2 A/D Converter Configuration

The A/D converter consists of the following hardware.

Table 12-1. A/D Converter Configuration

Item	Configuration
Analog input	8 channels (ANI0 to ANI7)
Control registers	A/D converter mode register (ADM) A/D converter input select register (ADIS)
Registers	Successive approximation register (SAR) A/D conversion result register (ADCR)

Figure 12-1. A/D Converter Block Diagram



Notes 1. Selector to select the number of channels to be used for analog input

2. Selector to select the channel for A/D conversion

(1) Successive approximation register (SAR)

This register compares the analog input voltage value to the voltage tap (compare voltage) value applied from the series resistor string and holds the result from the most significant bit (MSB).

When up to the least significant bit (LSB) is held (end 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 ends, the conversion result is loaded from the successive approximation register (SAR).

ADCR is read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes ADCR undefined.

(3) Sample & hold circuit

The sample & hold circuit samples each analog input signal sequentially applied from the input circuit and sends it to the voltage comparator. This circuit holds the sampled analog input voltage value during A/D conversion.

(4) Voltage comparator

The voltage comparator compares the analog input with the series resistor string output voltage.

(5) Series resistor string

The series resistor string is connected between AV_{REF} and AV_{SS} and generates a voltage to be compared with the analog input.

(6) ANI0 to ANI7 pins

These are 8-channel analog input pins used to input the analog signals to undergo A/D conversion to the A/D converter.

Except for the analog input pins selected by the A/D converter input select register (ADIS), these pins can be used as I/O port pins.

Cautions 1. Use ANI0 to ANI7 input voltages within the specified range. If a voltage higher than or equal to AV_{REF} or lower than or equal to AV_{SS} is applied (even if within the absolute maximum ratings), the converted value of the corresponding channel becomes undefined and may adversely affect the converted values of other channels.

2. The analog input pins ANI0 to ANI7 also function as I/O port (port 1) pins. Pins used as analog inputs should be set to the input mode. When A/D conversion is performed with any of pins ANI0 to ANI7 selected, be sure not to execute an instruction that inputs data 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.

(7) AV_{REF} pin

This pin inputs the A/D converter reference voltage.

It converts signals input to ANI0 to ANI7 into digital signals according to the voltage applied between AV_{REF} and AV_{SS}.

Caution A series resistor string of approximately 10 kΩ is connected between the AV_{REF} pin and the AV_{SS} pin. Therefore, if the output impedance of the reference voltage source is high, this will result in series connection to the series resistor string between the AV_{REF} pin and the AV_{SS} pin, and there will be a large reference voltage error.

(8) AV_{SS} pin

Ground potential pin of the A/D converter. It must be at the same level as the V_{SS} pin even if the A/D converter is not used.

(9) AV_{DD} pin

Analog power supply pin of the A/D converter. It must be at the same level as the V_{DD} pin even if the A/D converter is not used.

12.3 A/D Converter Control Registers

The following two registers are used to control the A/D converter.

- A/D converter mode register (ADM)
- A/D converter input select register (ADIS)

(1) A/D converter mode register (ADM)

This register sets the analog input channel for A/D conversion, conversion time, conversion start/stop, and external trigger.

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

$\overline{\text{RESET}}$ input sets ADM to 01H.

Figure 12-2. Format of A/D Converter Mode Register

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ADM	CS	TRG	FR1	FR0	ADM3	ADM2	ADM1	1	FF80H	01H	R/W

ADM3	ADM2	ADM1	Analog input channel selection
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

FR1	FR0	A/D conversion time selection[Note 1]	
		When operated at f_X = 5.0 MHz	When operated at f_X = 4.19 MHz
0	0	160/f_X (32.0 μs)	160/f_X (38.1 μs)
0	1	80/f_X (setting prohibited[Note 2])	80/f_X (19.1 μs)
1	0	200/f_X (40.0 μs)	200/f_X (47.7 μs)
1	1	Setting prohibited	

TRG	External trigger selection
0	No external trigger (software start mode)
1	Conversion started by external trigger (hardware start mode)

CS	A/D conversion operation control
0	Operation stop
1	Operation start

Notes 1. Set so that the A/D conversion time is 19.1 μs or more.

2. Setting prohibited because the A/D conversion time is less than 19.1 μs.

Cautions 1. Bit 0 must be set to 1.

2. In order to reduce the power consumption of the A/D converter when the standby function is working, clear bit 7 (CS) of this register to 0 to stop the A/D conversion operation before executing the HALT or STOP instruction.

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

Remark fx: Main system clock oscillation frequency

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

This register determines whether the ANI0/P10 to ANI7/P17 pins should be used for analog input channels or ports. The pins that are not selected for analog input pins can be used as I/O port pins.

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

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

Cautions 1. Set the analog input channel using the following procedure.

[1] Set the number of analog input channels using ADIS.

[2] Using the A/D converter mode register (ADM), select the channel to undergo A/D conversion among the channels which were set to analog input using ADIS.

2. No internal pull-up resistor can be connected to the channels set to analog input using ADIS, irrespective of the value of bit 1 (PUO1) of the pull-up resistor option register (PUO).

Figure 12-3. Format of A/D Converter Input Select Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ADIS	0	0	0	0	ADIS3	ADIS2	ADIS1	ADIS0	FF84H	00H	R/W

ADIS3	ADIS2	ADIS1	ADIS0	Analog input channel count selection
0	0	0	0	No analog input channels (P10 to P17)
0	0	0	1	1 channel (ANI0, P11 to P17)
0	0	1	0	2 channels (ANI0, ANI1, P12 to P17)
0	0	1	1	3 channels (ANI0 to ANI2, P13 to P17)
0	1	0	0	4 channels (ANI0 to ANI3, P14 to P17)
0	1	0	1	5 channels (ANI0 to ANI4, P15 to P17)
0	1	1	0	6 channels (ANI0 to ANI5, P16, P17)
0	1	1	1	7 channels (ANI0 to ANI6, P17)
1	0	0	0	8 channels (ANI0 to ANI7)
Other than above				Setting prohibited

12.4 A/D Converter Operations

12.4.1 Basic operations of A/D converter

[1] Set the number of analog input channels using the A/D converter input select register (ADIS).

[2] From among the analog input channels set by ADIS, select the channel for A/D conversion using the A/D converter mode register (ADM).

[3] The sample & hold circuit samples the voltage input to the selected analog input channel.

[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 the end of A/D conversion.

[5] Bit 7 of the successive approximation register (SAR) is set. The tap selector sets the series resistor string voltage tap to (1/2) AV_{REF}.

[6] The voltage difference between the series resistor string voltage tap and analog input is compared by the voltage comparator. If the analog input is larger than (1/2) AV_{REF}, the MSB of the SAR remains set. If the input is smaller than (1/2) AV_{REF}, the MSB is reset.

[7] Next, bit 6 of the SAR is automatically set and the operation proceeds to the next comparison. In this case, the series resistor string voltage tap is selected according to the preset value of bit 7 as described below.

- Bit 7 = 1: (3/4) AV_{REF}
- Bit 7 = 0: (1/4) AV_{REF}

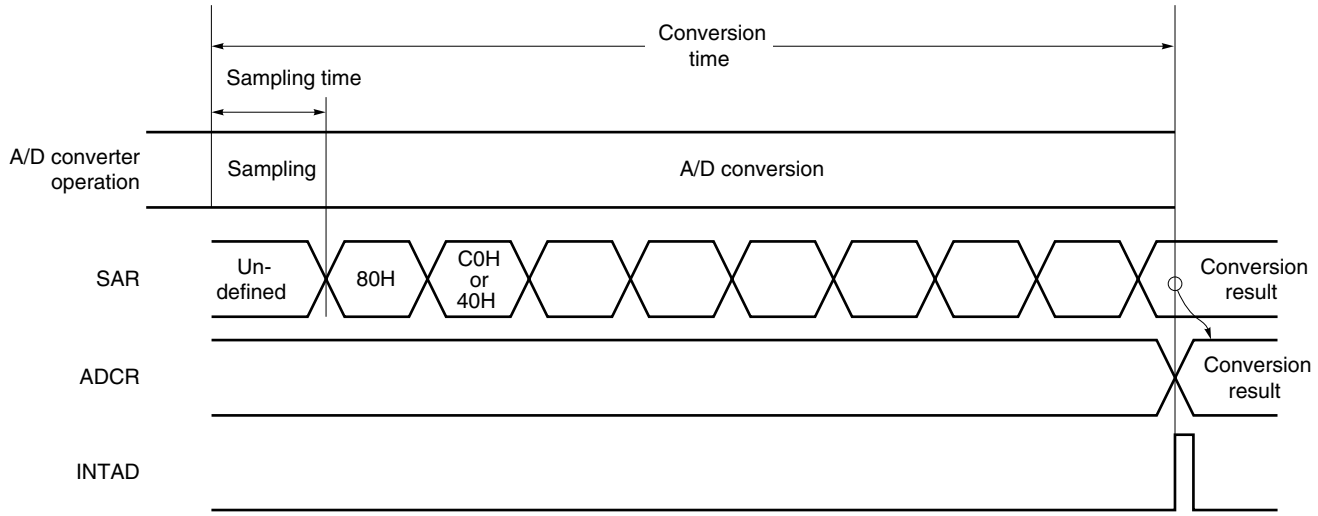
The voltage tap and analog input voltage are compared and bit 6 of the SAR is manipulated using the result as follows.

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

[8] Comparison of this sort continues up to bit 0 of the SAR.

[9] Upon completion of the comparison of 8 bits, an effective digital result value remains in the SAR and the result value is transferred to and latched in the A/D conversion result register (ADCR).
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

Figure 12-4. Basic Operation of A/D Converter



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 CS is set (1), conversion starts again from the beginning.

$\overline{\text{RESET}}$ input makes ADCR undefined.

12.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI7) and the A/D conversion result (the value stored in the A/D conversion result register (ADCR)) is shown by the following expression.

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

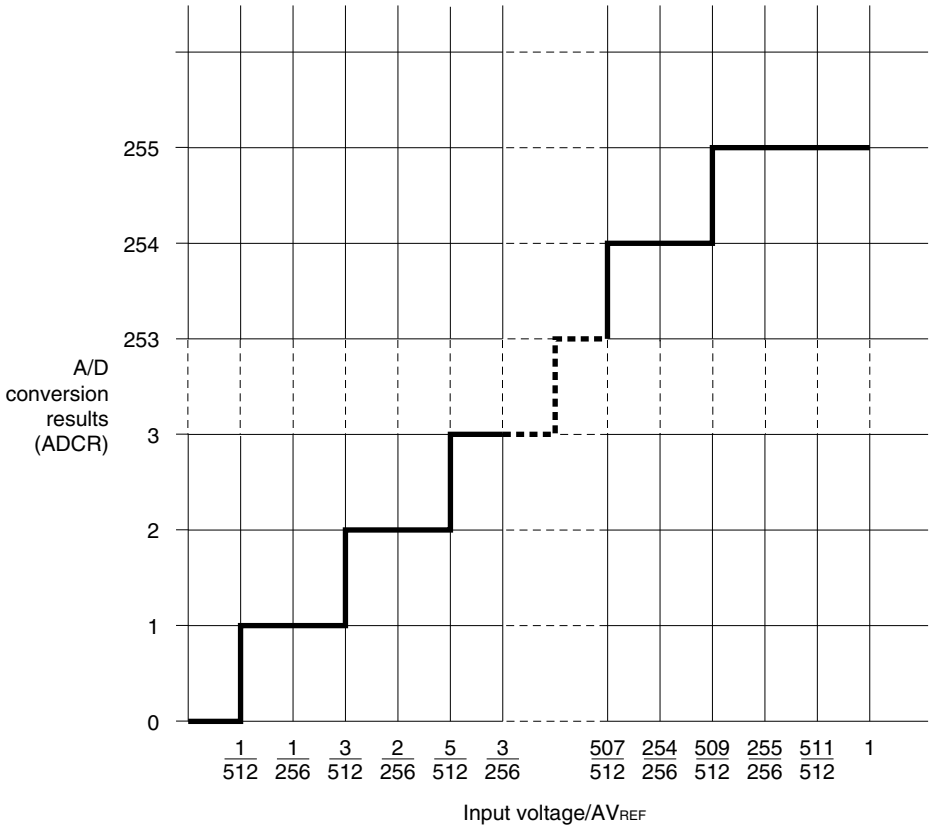
or

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

Remark INT ( ): Function which returns the integer part of the value in parentheses
V_{IN}: Analog input voltage
AV_{REF}: AV_{REF} pin voltage
ADCR: A/D conversion result register (ADCR) value

Figure 12-5 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 12-5. Relationship Between Analog Input Voltage and A/D Conversion Result



12.4.3 A/D converter operating mode

Select one analog input channel from among ANI0 to ANI7 using the A/D converter input select register (ADIS) and A/D converter mode register (ADM) and start A/D conversion.

A/D conversion can be started in the following two ways.

- Hardware start: Conversion is started by trigger input (INTP3).
- Software start: Conversion is started by setting ADM.

The 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

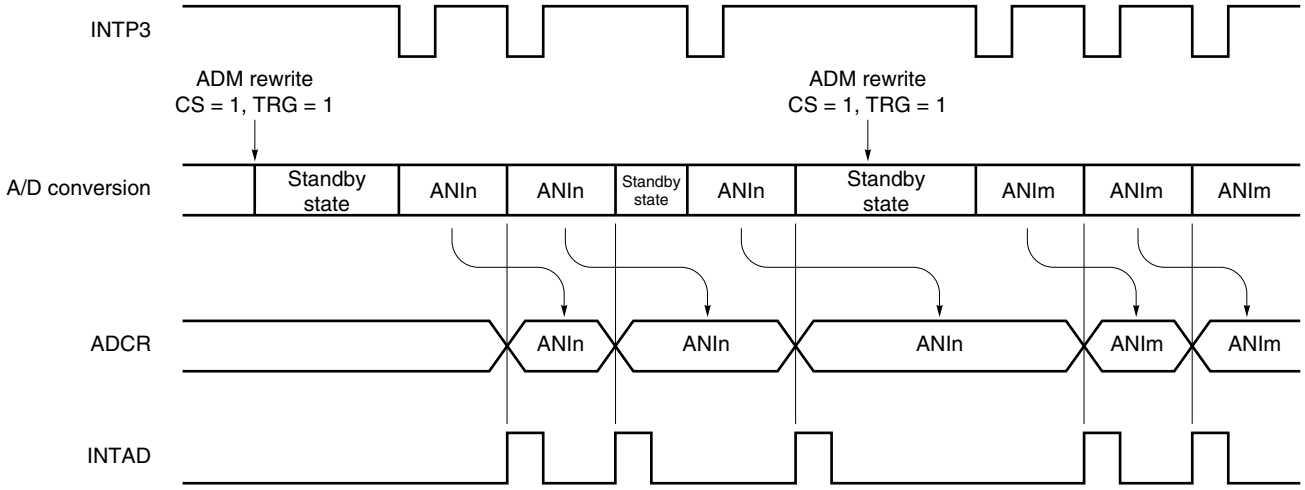
When bit 6 (TRG) and bit 7 (CS) of the A/D converter mode register (ADM) are set to 1, the A/D conversion standby state is set. When the external trigger signal (INTP3) is input, A/D conversion starts on the voltage applied to the analog input pins specified by bits 1 to 3 (ADM1 to ADM3) of ADM.

At the end of 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 12-6. A/D Conversion by Hardware Start



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

(2) A/D conversion by software start

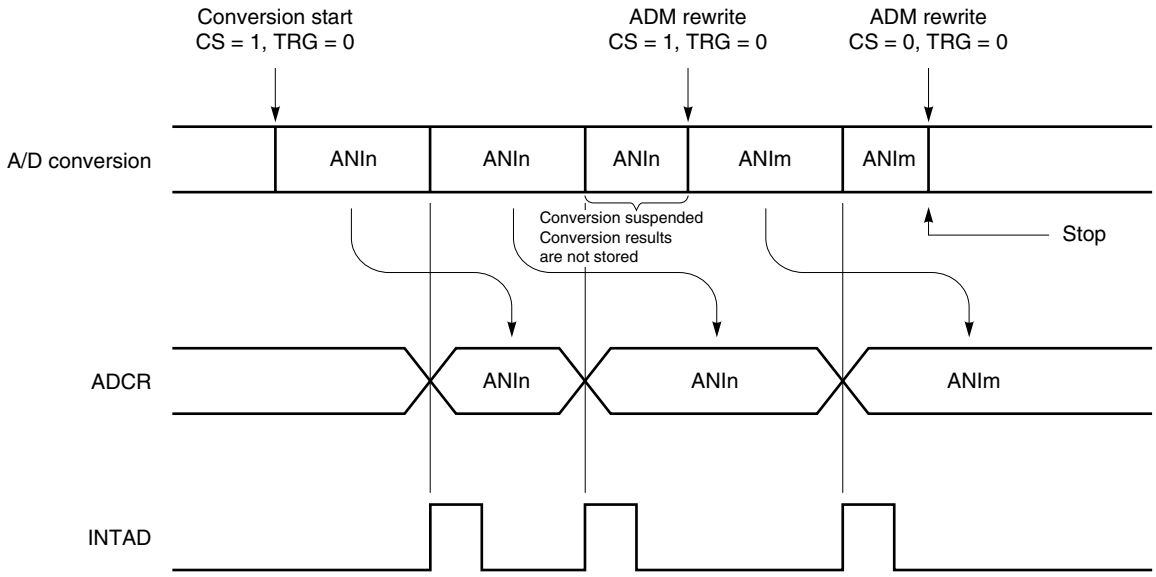
When bit 6 (TRG) and bit 7 (CS) of the A/D converter mode register (ADM) are set to 0 and 1, respectively, A/D conversion starts on the voltage applied to the analog input pins specified by bits 1 to 3 (ADM1 to ADM3) of ADM.

At the end of 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. A/D conversion 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 12-7. A/D Conversion by Software Start



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

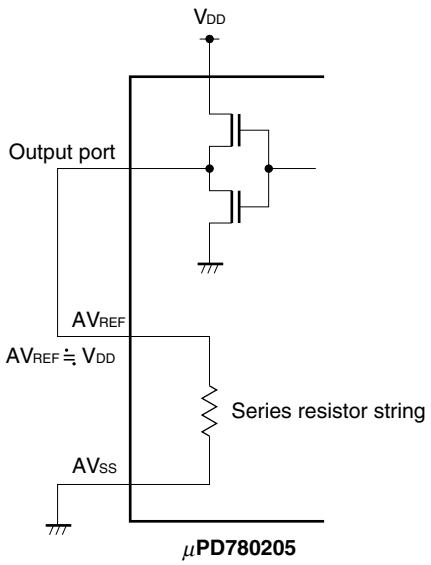
12.5 A/D Converter Precautions

(1) Power consumption in standby mode

The A/D converter operates on the main system clock. Therefore, its operation stops in STOP mode or in HALT mode with the subsystem clock. As a current still flows in the AV_{REF} pin at this time, this current must be cut in order to minimize the overall system power consumption.

In this example, the power consumption can be reduced if a low level is output to the output port in the standby mode. However, the actual AV_{REF} voltage is not so accurate and, accordingly, the converted value is not accurate and should be used for relative comparison only.

Figure 12-8. Example of Method of Reducing Power Consumption in Standby Mode



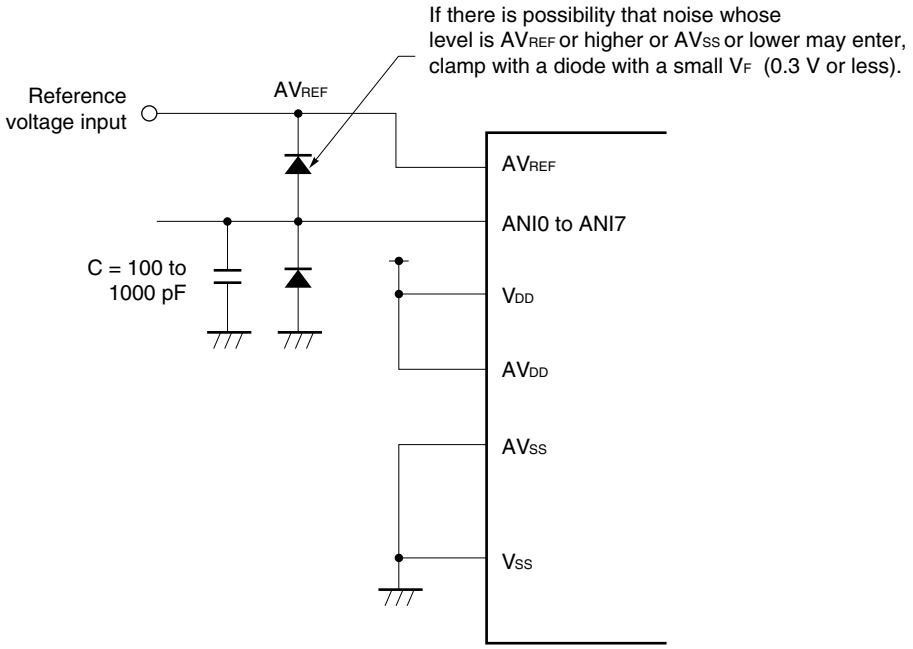
(2) Input range of ANI0 to ANI7

The input voltages of ANI0 to ANI7 should be within the specification range. In particular, if a voltage greater than or equal to AV_{REF} or less than or equal to AV_{SS} is input (even if within the absolute maximum rating range), the conversion value for that channel will be undefined, and 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 the AV_{REF} and ANI0 to ANI7 pins. 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 12-9 in order to reduce noise.

Figure 12-9. Analog Input Pin Processing



(4) Pins ANI0/P10 to ANI7/P17

The analog input pins ANI0 to ANI7 also function as I/O port (port 1) pins.

Pins used as analog inputs should be set to the input mode.

When A/D conversion is performed with any of pins ANI0 to ANI7 selected, be sure not to execute an instruction that inputs data 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_{REF} pin input impedance

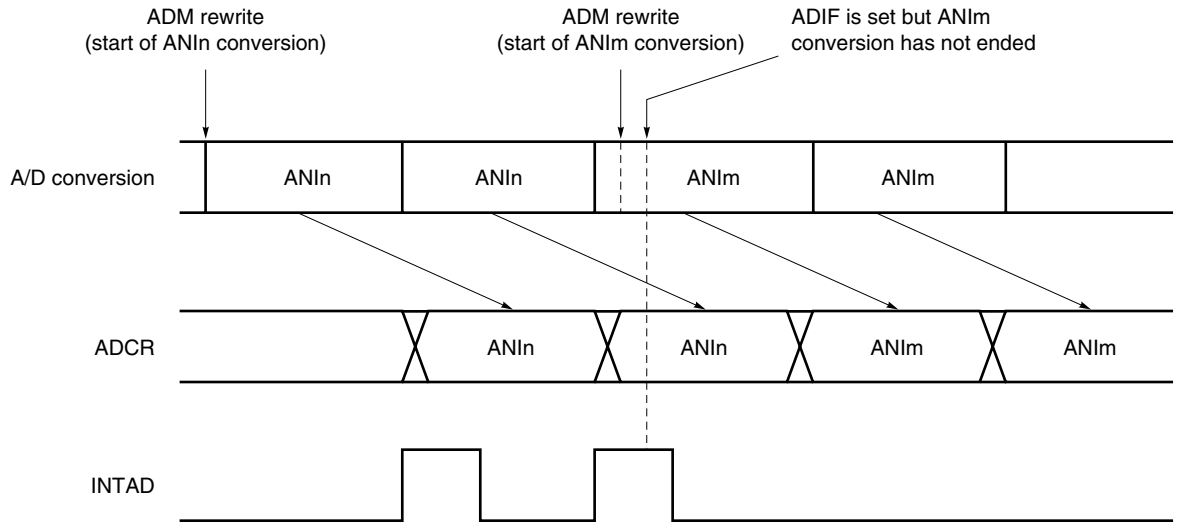
A series resistor string of approximately 10 kΩ is connected between the AV_{REF} pin and the AV_{SS} pin. Therefore, if the output impedance of the reference voltage source is high, this will result in series connection to the series resistor string between the AV_{REF} pin and the AV_{SS} pin, and there will be a large reference voltage error.

(6) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the A/D converter mode register (ADM) is changed. Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result for the analog input before changing and ADIF may be set immediately before rewriting ADM. In this case, if ADIF is read immediately after the rewriting of ADM, ADIF is set despite the fact that A/D conversion of the analog input after changing has not been completed (refer to **Figure 12-10**).

When A/D conversion is stopped, ADIF must be cleared before restarting.

Figure 12-10. A/D Conversion End Interrupt Request Generation Timing



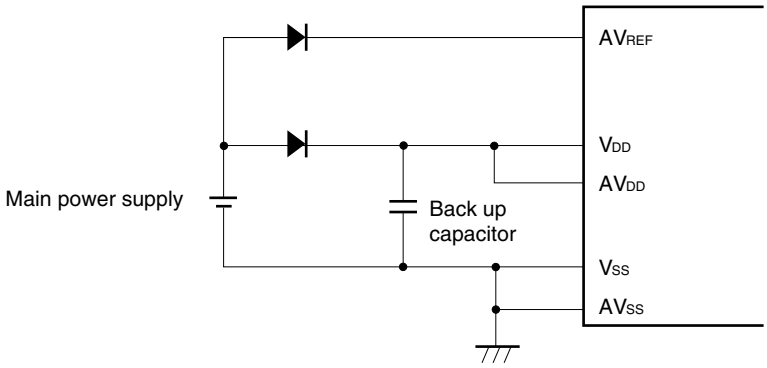
Remark n = 0, 1, ... , 7
m = 0, 1, ... , 7

(7) AV_{DD} pin

The AV_{DD} pin is the analog circuit power supply pin, and supplies power to the input circuits of ANI0/P10 to ANI7/P17.

Therefore, be sure to apply the voltage at the same level as V_{DD} as shown in Figure 12-11 even in an application where the power supply is switched to the back-up power supply.

Figure 12-11. AV_{DD} Pin Connection



CHAPTER 13 SERIAL INTERFACE CHANNEL 0

The μPD780208 Subseries incorporates two clocked serial interface channels.

The differences between channels 0 and 1 are as follows (refer to **CHAPTER 14 SERIAL INTERFACE CHANNEL 1** for details of serial interface channel 1).

Table 13-1. Differences Between Channels 0 and 1

Serial Transfer Mode		Channel 0	Channel 1
3-wire serial I/O	Clock selection	$f_X/2^2$, $f_X/2^3$, $f_X/2^4$, $f_X/2^5$, $f_X/2^6$, $f_X/2^7$, $f_X/2^8$, $f_X/2^9$, external clock, TO2 output	$f_X/2^2$, $f_X/2^3$, $f_X/2^4$, $f_X/2^5$, $f_X/2^6$, $f_X/2^7$, $f_X/2^8$, $f_X/2^9$, external clock, TO2 output
	Transfer method	MSB/LSB switchable as the start bit	MSB/LSB switchable as the start bit Automatic transmit/ receive function
	Transfer end flag	Serial transfer end interrupt request flag (CSIIF0)	Serial transfer end interrupt request flag (CSIIF1)
SBI (serial bus interface)		Use possible	None
2-wire serial I/O			

13.1 Functions of Serial Interface Channel 0

Serial interface channel 0 has the following four modes.

Table 13-2. Modes of Serial Interface Channel 0

Operation Mode	Pins Used	Features	Usage
Operation stop mode	–	• Used when serial transfer is not carried out. • Power consumption can be reduced.	–
3-wire serial I/O mode	$\overline{\text{SCK0}}$ (serial clock), SO0 (serial output), SI0 (serial input)	• Input and output lines are independent and they can transfer/receive at the same time, so the data transfer processing time is short. • The start bit of 8-bit data to undergo serial transfer is switchable between MSB and LSB.	These modes are used for connection of peripheral ICs and display controllers that incorporate a clocked serial interface.
SBI mode	$\overline{\text{SCK0}}$ (serial clock), SB0 or SB1 (serial data bus)	• Enables configuration of serial bus with two signal lines, thus, even when connected to some microcontrollers, the number of ports can be cut and the wiring on the board reduced. • High-speed serial interface complying with the NEC Electronics standard bus format. • Address, command, and data information sent on the serial bus • The wakeup function for handshake and acknowledge and busy signal output function can also be used.	
2-wire serial I/O mode	$\overline{\text{SCK0}}$ (serial clock), SB0 or SB1 (serial data bus)	• Can cope with any data transfer format by program. Thus, the handshake lines previously necessary for connection of two or more devices can be removed.	

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

13.2 Configuration of Serial Interface Channel 0

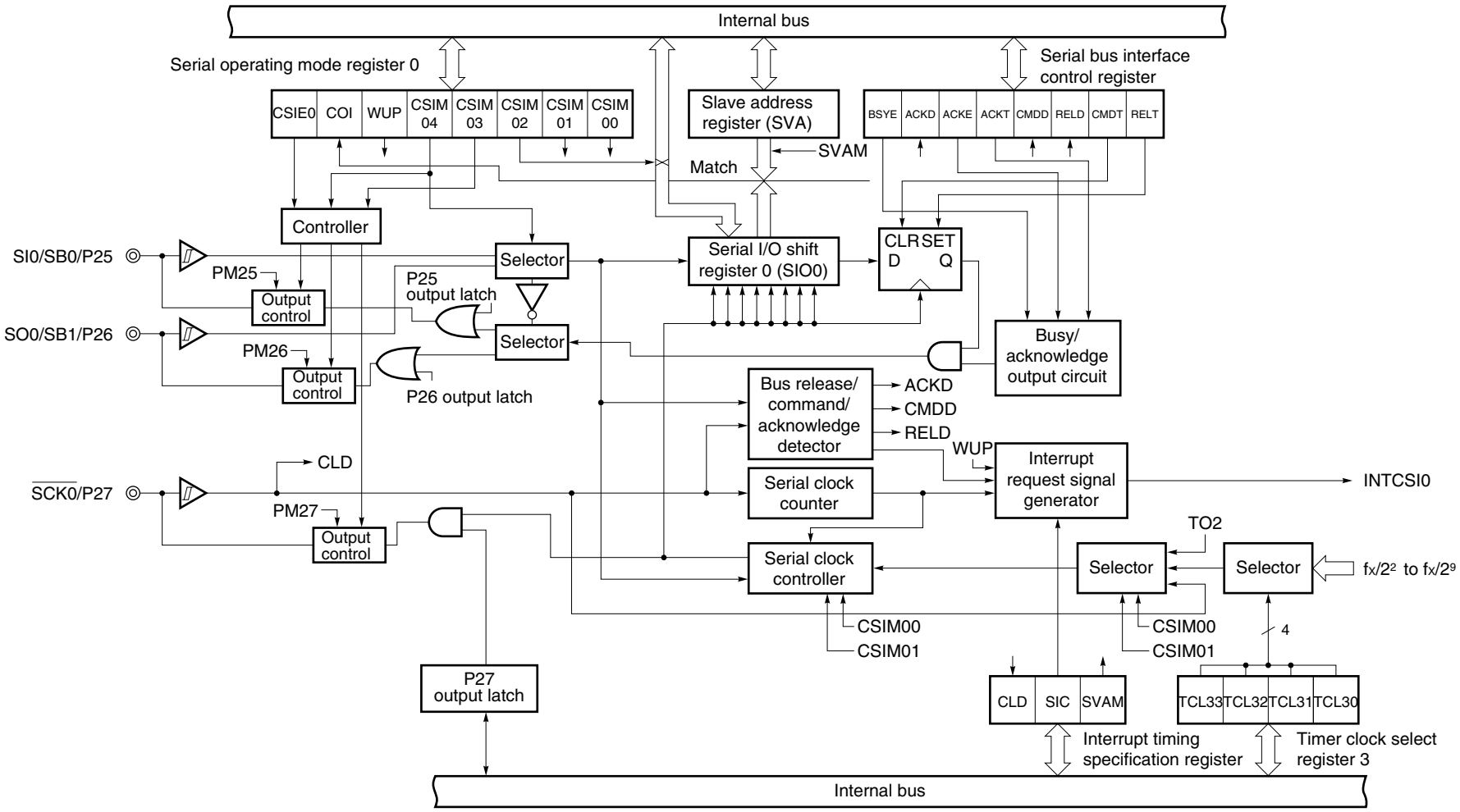
Serial interface channel 0 consists of the following hardware.

Table 13-3. Configuration of Serial Interface Channel 0

Item	Configuration
Registers	Serial I/O shift register 0 (SIO0) Slave address register (SVA)
Control registers	Timer clock select register 3 (TCL3) Serial operating mode register 0 (CSIM0) Serial bus interface control register (SBIC) Interrupt timing specification register (SINT) Port mode register 2 (PM2)[Note]

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

Figure 13-1. Block Diagram of Serial Interface Channel 0



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 used to carry out parallel/serial conversion and serial transmission/reception (shift operations) in synchronization with the serial clock.

SIO0 is set with an 8-bit memory manipulation instruction.

When bit 7 (CSIE0) of serial operating mode register 0 (CSIM0) is 1, writing data to SIO0 starts serial operation.

In transmission, data written to SIO0 is output to the serial output (SO0) or serial data bus (SB0/SB1).

In reception, data is read from the serial input (SI0) or SB0/SB1 to SIO0.

The bus configuration in SBI mode and 2-wire serial I/O mode enables the pin to function as both an input and output pin. Thus, when a device is receiving, 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 used to set the slave address value for connection of a slave device to the serial bus. This register is not used in the 3-wire serial I/O mode.

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

The master device outputs a slave address to the connected slave devices for selection of a particular slave device. These two data (the slave address output from the master device and the SVA value) are compared by the 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 the LSB-masked higher 7 bits by setting bit 4 (SVAM) of the interrupt timing specification register (SINT) to 1.

If no matching is detected in address reception, bit 2 (RELD) of the serial bus interface control register (SBIC) is cleared to 0. The wakeup function can be used by setting bit 5 (WUP) of CSIM0 to 1. In this case, the interrupt request signal (INTCSI0) is generated only when the slave address output by the master matches the value of SVA, and it can be ascertained by this interrupt request that the master is requesting communication. If bit 5 (SIC) of the interrupt timing specification register (SINT) is set to 1, the wakeup function cannot be used even if WUP is set to 1 (an interrupt request signal is generated when bus release is detected). To use the wakeup function, clear SIC to 0.

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

$\overline{\text{RESET}}$ input makes SVA undefined.

(3) SO0 latch

This latch holds the SI0/SB0/P25 and SO0/SB1/P26 pin levels. It can be directly controlled by software.

In the SBI mode, this latch is set at the end of the 8th serial clock.

(4) Serial clock counter

This counter counts the serial clocks to be output and input during transmission/reception and checks whether 8-bit data has been transmitted/received.

(5) Serial clock controller

This circuit controls serial clock supply to 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 an 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 the wakeup function specification bit. It is bit 5 of serial operating mode register 0 (CSIM0). To use the wakeup function (WUP = 1), clear bit 5 (SIC) of the interrupt timing specification register (SINT) to 0.

(7) Busy/acknowledge output circuit and bus release/command/acknowledge detector

These two circuits output and detect various control signals in the SBI mode.
These do not operate in the 3-wire serial I/O mode and 2-wire serial I/O mode.

(8) P27 output latch

This latch generates a serial clock by software at the end of eight serial clocks.
When using serial interface channel 0, set the P27 output latch to 1.
$\overline{\text{RESET}}$ input sets the latch to 0.

13.3 Control Registers of Serial Interface Channel 0

The following four 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 specification register (SINT)

(1) Timer clock select register 3 (TCL3) (See **Figure 13-2**.)

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

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

$\overline{\text{RESET}}$ input sets TCL3 to 88H.

Remark Besides setting the serial clock of serial interface channel 0, TCL3 sets the serial clock of serial interface channel 1.

(2) Serial operating mode register 0 (CSIM0) (See **Figure 13-3**.)

This register sets the serial interface channel 0 serial clock, operating mode, operation enable/stop wakeup 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, or SBI) while the operation of serial interface channel 0 is enabled. Stop the serial operation before changing the operation mode.

Figure 13-2. Format of Timer Clock Select Register 3

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33	TCL32	TCL31	TCL30	Serial interface channel 0 serial clock selection
0	1	1	0	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_X/2^9$ (9.8 kHz)
Other than above				Setting prohibited

TCL37	TCL36	TCL35	TCL34	Serial interface channel 1 serial clock selection
0	1	1	0	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_X/2^9$ (9.8 kHz)
Other than above				Setting prohibited

Caution If TCL3 is to be rewritten with data other than identical data, stop the serial transfer first.

Remarks 1. fx: Main system clock oscillation frequency

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

Figure 13-3. Format of Serial Operating Mode Register 0

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	After reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM 04	CSIM 03	CSIM 02	CSIM 01	CSIM 00	FF60H	00H	R/W[Note 1]

R/W

CSIM 01	CSIM 00	Serial interface channel 0 clock selection
0	x	Input clock to $\overline{SCK0}$ pin from off-chip
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operating mode	Start bit	SI0/P25 pin function	SO0/P26 pin function	$\overline{SCK0}$/P27 pin function
0	×	0	1	×	0	0	0	1	3-wire serial I/O mode	MSB	SI0[Note 2] (input)	SO0 (CMOS output)	$\overline{SCK0}$ (CMOS I/O)
		1								LSB			
1	0	0	[Note 3] ×	[Note 3] ×	0	0	0	1	SBI mode	MSB	P25 (CMOS I/O)	SB1 (N-ch open-drain I/O)	$\overline{SCK0}$ (CMOS I/O)
		1	0	0	[Note 3] ×	[Note 3] ×	0	1			SB0 (N-ch open-drain I/O)	P26 (CMOS I/O)	
1	1	0	[Note 3] ×	[Note 3] ×	0	0	0	1	2-wire serial I/O mode	MSB	P25 (CMOS I/O)	SB1 (N-ch open-drain I/O)	$\overline{SCK0}$ (N-ch open-drain I/O)
		1	0	0	[Note 3] ×	[Note 3] ×	0	1			SB0 (N-ch open-drain I/O)	P26 (CMOS I/O)	

R/W

WUP	Wakeup 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 (SVA) in SBI mode

R

COI	Slave address comparison result flag[Note 5]
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 P25 (CMOS input) when used only for transmission.

3. Can be used freely as port function.

4. To use the wakeup function (WUP = 1), clear bit 5 (SIC) of the interrupt timing specification register (SINT) to 0.

5. COI becomes 0 when CSIE0 = 0.

Remark ×: don't care
PM××: Port mode register
P××: Port output latch

(3) Serial bus interface control register (SBIC)

This register sets the 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 13-4. Format of Serial Bus Interface Control Register (1/2)

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

R/W		
R/W	RELT	Used for bus release signal output. When RELT = 1, the SO latch is set to 1. After SO latch setting, RELT is automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W		
R/W	CMDT	Used for command signal output. When CMDT = 1, the SO latch is cleared to 0. After SO latch clearance, CMDT is automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R	RELD	Bus release detection	
	Clear conditions (RELD = 0)		Set conditions (RELD = 1)
	• When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied		• When bus release signal (REL) is detected

R	CMDD	Command detection	
	Clear conditions (CMDD = 0)		Set conditions (CMDD = 1)
	• When transfer start instruction is executed • When bus release signal (REL) is detected • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied		• When command signal (CMD) is detected

R/W		
R/W	ACKT	The 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, ACKT is automatically cleared to 0. Also cleared to 0 upon start of serial interface transfer or when CSIE0 = 0.

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

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

Figure 13-4. Format of Serial Bus Interface Control Register (2/2)

R/W

ACKE	Acknowledge signal output control	
0	Acknowledge signal automatic output disabled (output with ACKT enabled)	
1	Before completion of transfer	The acknowledge signal is output in synchronization with the 9th clock falling edge of $\overline{SCK0}$ (automatically output when ACKE = 1).
	After completion of transfer	The acknowledge signal is output in synchronization with the falling edge of $\overline{SCK0}$ just after execution of the instruction to be set to 1 (automatically output when ACKE = 1). However, ACKE is not automatically cleared to 0 after acknowledge signal output.

R

ACKD	Acknowledge detection	
Clear conditions (ACKD = 0)		Set conditions (ACKD = 1)
• At the falling edge of $\overline{SCK0}$ immediately after the busy mode has been released when a transfer start instruction is executed • When CSIE0 = 0 • When $\overline{RESET}$ input is applied		• When acknowledge signal ($\overline{ACK}$) is detected at the rising edge of $\overline{SCK0}$ clock after completion of transfer

R/W

BSYE[Note]	Synchronizing busy signal output control
0	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 is disabled.
1	Busy signal is output 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, the BSYE flag is not cleared to 0.

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)

(4) Interrupt timing specification register (SINT)

This register sets the bus release interrupt and address mask functions and displays the $\overline{SCK0}$ pin level status.

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

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

Figure 13-5. Format of Interrupt Timing Specification Register

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

R/W

SVAM	SVA bit to be used as slave address
0	Bits 0 to 7
1	Bits 1 to 7

R/W

SIC	INTCSI0 interrupt source selection[Note 2]
0	CSIIF0 is set upon termination of serial interface channel 0 transfer
1	CSIIF0 is set upon bus release detection or termination of serial interface channel 0 transfer

R

CLD	$\overline{SCK0}$ pin level[Note 3]
0	Low level
1	High level

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

2. When using wakeup function, set SIC to 0.

3. When CSIE0 = 0, CLD becomes 0.

Caution Be sure to set bits 0 to 3 to 0.

Remark SVA: Slave address register

CSIIF0: Interrupt request flag for INTCSI0

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

13.4 Operations of Serial Interface Channel 0

The following four operating modes are available for serial interface channel 0.

- Operation stop mode
- 3-wire serial I/O mode
- SBI mode
- 2-wire serial I/O mode

13.4.1 Operation stop mode

Serial transfer is not carried out in the operation stop mode. Thus, power consumption can be reduced. Serial I/O shift register 0 (SIO0) does not carry out shift operations and can be used as an ordinary 8-bit register.

In the operation stop mode, the P25/SI0/SB0, P26/SO0/SB1, and P27/$\overline{\text{SCK0}}$ pins can be used as ordinary I/O ports.

(1) Register setting

The operation stop mode is set by 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

13.4.2 3-wire serial I/O mode operation

The 3-wire serial I/O mode is used for connection of peripheral ICs and display controllers that incorporate a clocked serial interface.

Communication is carried out using three lines: a serial clock ($\overline{SCK0}$), serial output (SO0), and serial input (SI0).

(1) Register setting

The 3-wire serial I/O mode is set by 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.

$\overline{RESET}$ input sets CSIM0 to 00H.

Symbol

	<7>	<6>	<5>	4	3	2	1	0	Address	After reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM 04	CSIM 03	CSIM 02	CSIM 01	CSIM 00	FF60H	00H	R/W[Note 1]

R/W

CSIM 01	CSIM 00	Serial interface channel 0 clock selection
0	×	Input clock to $\overline{SCK0}$ pin from off-chip
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operating mode	Start bit	SI0/P25 pin function	SO0/P26 pin function	$\overline{SCK0}$/P27 pin function
0	×	0	1	×	0	0	0	1	3-wire serial I/O mode	MSB	SI0[Note 2] (input)	SO0 (CMOS output)	$\overline{SCK0}$ (CMOS I/O)
		1								LSB			
1	0	SBI mode (refer to **13.4.3 SBI mode operation**)											
1	1	2-wire serial I/O mode (refer to **13.4.4 2-wire serial I/O mode operation**)											

R/W

WUP	Wakeup 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) 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 P25 (CMOS input) when used only for transmission.

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

4. When CSIE0 = 0, COI becomes 0.

Remark ×: don't care

PM××: Port mode register

P××: 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, the SO latch is set to 1. After SO latch setting, RELT is automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W		
	CMDT	When CMDT = 1, the SO latch is cleared to 0. After SO latch clearance, CMDT is automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

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

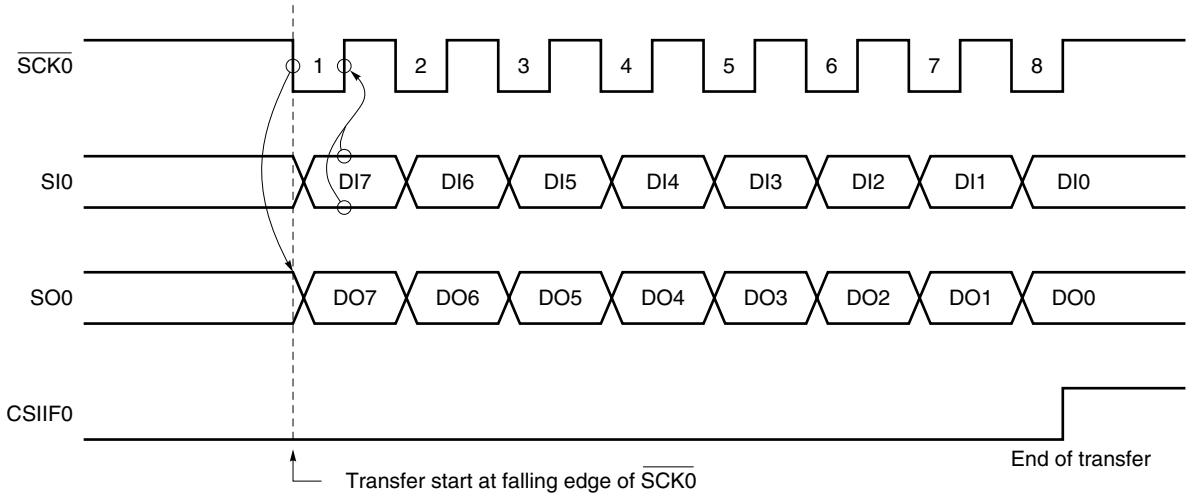
(2) Communication operation

The 3-wire serial I/O mode is used for data transmission/reception in 8-bit units. Data transmission/reception is carried out bit by bit in synchronization with the serial clock.

Shift operations of serial I/O shift register 0 (SIO0) are carried out at the falling edge of the serial clock ($\overline{SCK0}$). The transmit data is held in the SO0 latch and is output from the SO0 pin. The receive data input to the SI0 pin is latched into SIO0 at the rising edge of $\overline{SCK0}$.

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

Figure 13-6. 3-Wire Serial I/O Mode Timing



The SO0 pin is used for CMOS output and generates the SO0 latch status. 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).

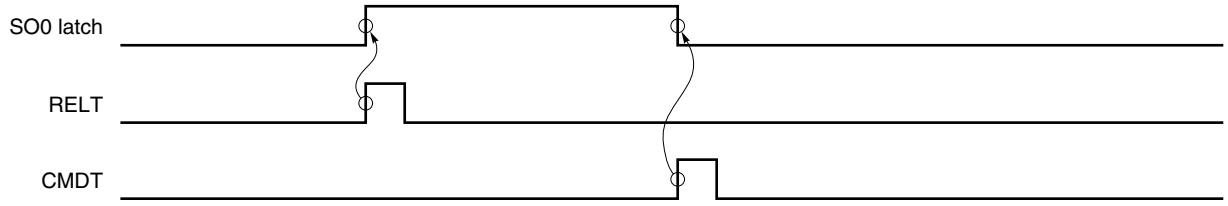
However, do not carry out this manipulation during serial transfer.

Control the $\overline{SCK0}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **13.4.5 $\overline{SCK0}$/P27 pin output manipulation**).

(3) Signals

Figure 13-7 shows the RELT and CMDT operations.

Figure 13-7. RELT and CMDT Operations



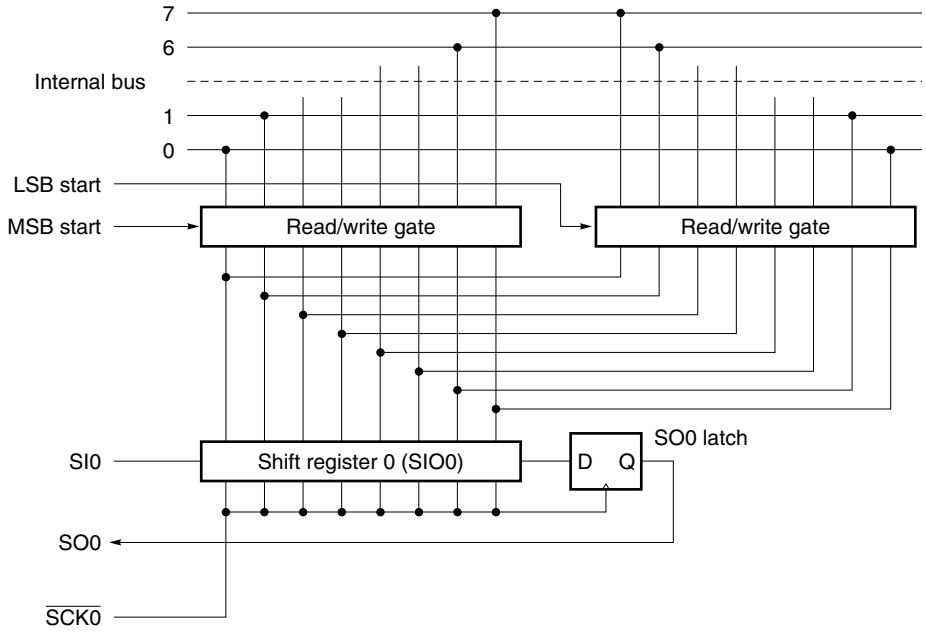
(4) MSB/LSB switching as the start bit

In the 3-wire serial I/O mode, transfer can be selected to start from the MSB or LSB.

Figure 13-8 shows the configuration of serial I/O shift register 0 (SIO0) and the internal bus. As shown in the figure, the MSB/LSB can be read/written in reverse form.

MSB/LSB switching as the start bit can be specified using bit 2 (CSIM02) of serial operating mode register 0 (CSIM0).

Figure 13-8. Circuit for Switching Transfer Bit Order



Start bit switching is realized by switching the bit order for data write to SIO0. The SIO0 shift order remains unchanged.

Thus, switch the MSB/LSB start bit before writing data to the shift register.

(5) Transfer start

Serial transfer is started by setting transfer data to serial I/O shift register 0 (SIO0) when the following two conditions are satisfied.

- Serial interface channel 0 operation control bit (CSIE0) = 1
- Internal serial clock is stopped or $\overline{SCK0}$ is at high level after 8-bit serial transfer.

Caution If CSIE0 is set to "1" after data write to SIO0, transfer does not start.

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

13.4.3 SBI mode operation

SBI (Serial Bus Interface) is a high-speed serial interface that complies with the NEC Electronics serial bus format.

SBI is a single-master high-speed serial bus with a format in which a bus configuration function has been added to the clocked serial I/O method so that it can carry out communication with two or more devices using two signal lines. Thus, when configuring a serial bus with two or more microcontrollers or peripheral ICs, the number of ports to be used and the number of wires on the board can be decreased.

The master device can output to the serial data bus of the slave device "addresses" for selection of the serial communication target device, "commands" to instruct the target device and actual "data". The slave device can identify the received data as an "address", "command", or "data", by hardware. This function can simplify the application program which controls serial interface channel 0.

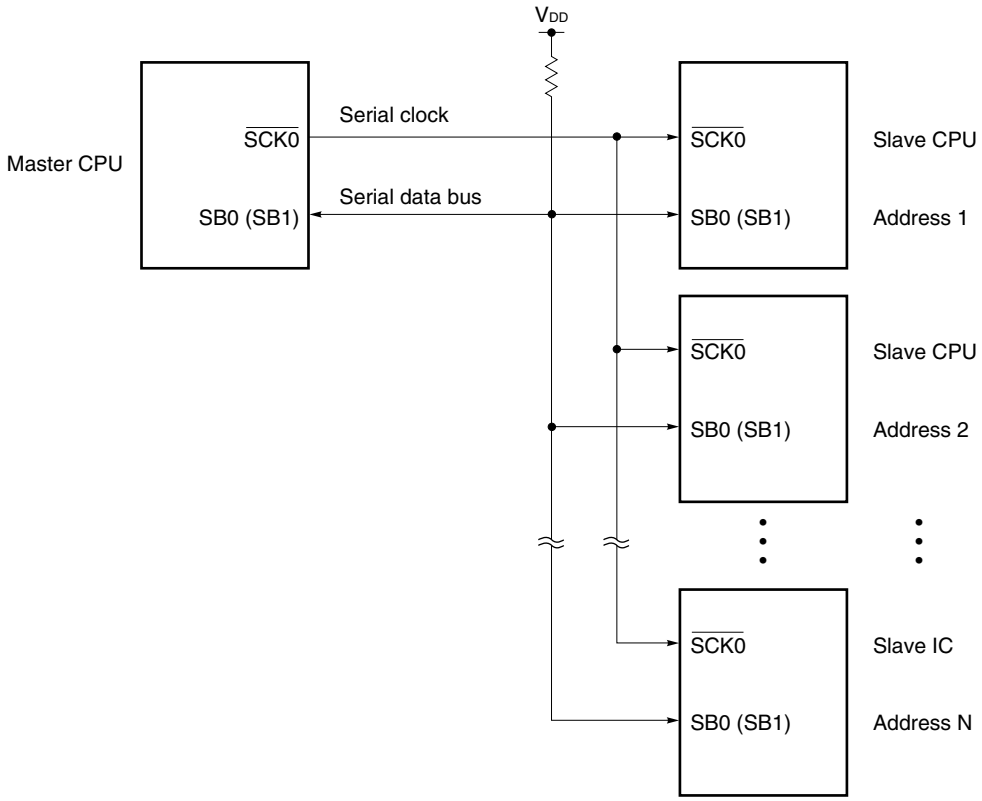
The SBI function is incorporated into various devices including the 75XL Series and 78K Series.

Figure 13-9 shows a serial bus configuration example when a CPU having a serial interface compliant with SBI and peripheral ICs are used.

In SBI, the SB0 (SB1) serial data bus pin is an open-drain output and so the serial data bus line is in a wired-OR state. A pull-up resistor is therefore necessary for the serial data bus line.

Refer to **(11) SBI mode precautions (d)** described later when the SBI mode is used.

Figure 13-9. Example of Serial Bus Configuration with SBI



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

(1) SBI functions

With the conventional serial I/O method, when a serial bus is configured by connecting two or more devices, many ports and wiring are necessary to distinguish chip select signals and command/data and to judge the busy state because only a data transfer function is available. Controlling these operations by software places a heavy load on the software.

In SBI, a serial bus can be configured with two signal lines: a serial clock $\overline{\text{SCK0}}$ and serial data bus SB0 (SB1). Thus, SBI is effective to decrease the number of microcontroller ports and wiring and routing on the board.

The SBI functions are described below.

(a) Address/command/data identification 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) Wakeup function

The slave can easily judge address reception (chip select judgment) using the wakeup function (which can be set/reset by software).

When the wakeup function is set, the interrupt request signal (CSIIF0) is generated upon reception of a match address. Thus, when communication is executed with two or more devices, the CPUs of other than the selected slave device can operate irrespective of serial communication.

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

The acknowledge signal to check serial data reception is controlled.

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

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

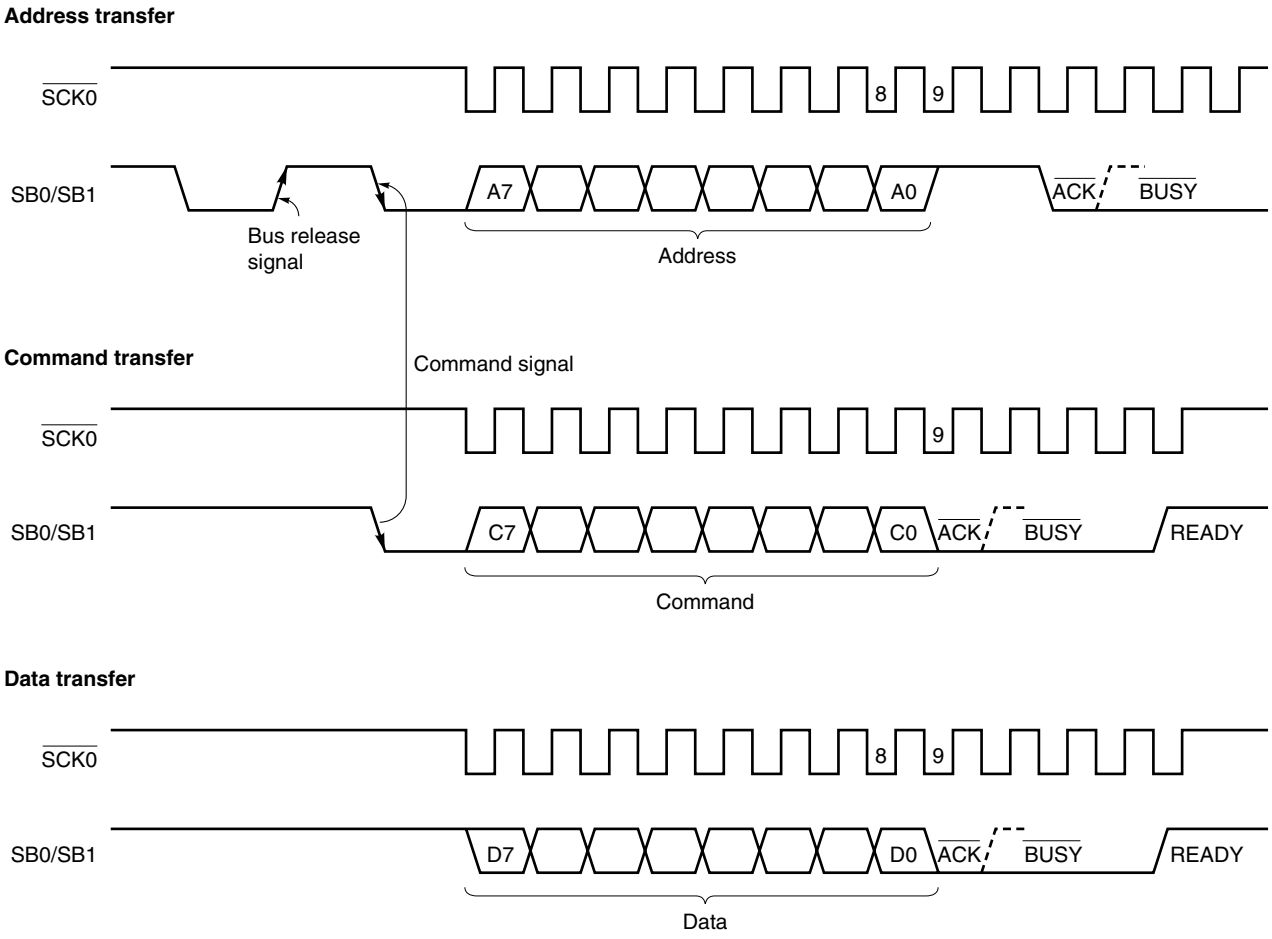
(2) SBI definition

The SBI serial data format is defined as follows.

Serial data to be transferred with SBI is distinguished into three types, “address”, “command”, and “data”.

Figure 13-10 shows the address, command, and data transfer timing.

Figure 13-10. SBI Transfer Timing



Remark The broken line indicates the READY status.

The bus release signal and the command signal are output by the master device. $\overline{\text{BUSY}}$ is output by the slave. $\overline{\text{ACK}}$ can be output by either the master or slave device (normally, the 8-bit data receiver outputs $\overline{\text{ACK}}$).

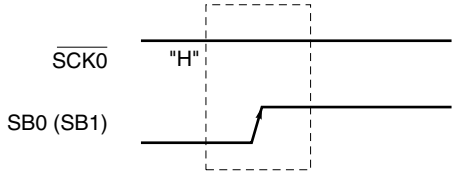
Serial clocks continue to be output by the master device from 8-bit data transfer start to $\overline{\text{BUSY}}$ reset.

(a) Bus release signal (REL)

The bus release signal is identified when the SB0 (SB1) line has changed from low level to high level while the $\overline{SCK0}$ line is high level (without serial clock output).

This signal is output by the master device.

Figure 13-11. Bus Release Signal



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

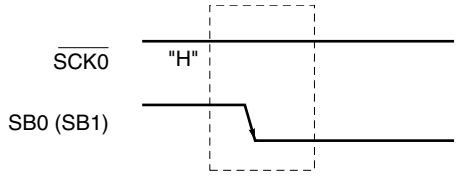
The slave device incorporates hardware to detect the bus release signal.

Caution If the SB0 (SB1) line changes from low level to high level while the $\overline{SCK0}$ line is high level, it is recognized as a bus release signal. Therefore, if the changing timing of the bus fluctuates because of substrate capacitance, etc., it may be recognized as a bus release signal even while data is being transmitted. Care should therefore be taken in the wiring.

(b) Command signal (CMD)

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

Figure 13-12. Command Signal



The command signal indicates that from this point, the master will send a command to the slave (however, command signals following bus release signals indicate that an address will be sent).

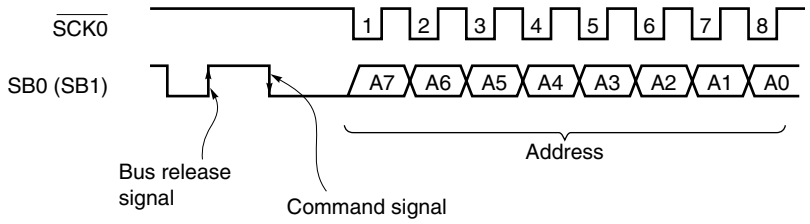
The slave incorporates hardware to detect command signals.

Caution If the SB0 (SB1) line changes from high level to low level while the $\overline{SCK0}$ line is high level, it is recognized as a command signal. Therefore, if the changing timing of the bus fluctuates because of substrate capacitance, etc., it may be recognized as a command signal even while data is being transmitted. Care should therefore be taken in the wiring.

(c) Address

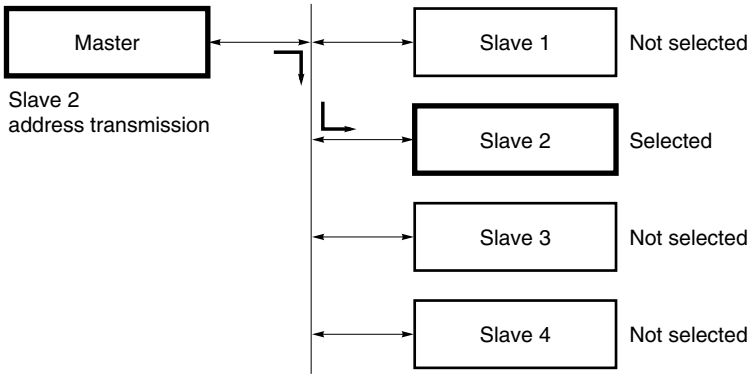
An address is 8-bit data which the master device outputs to the slave device connected to the bus line in order to select a particular slave device.

Figure 13-13. Address



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 the 8-bit data matches the slave’s 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 13-14. Slave Selection by 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 13-15. Commands

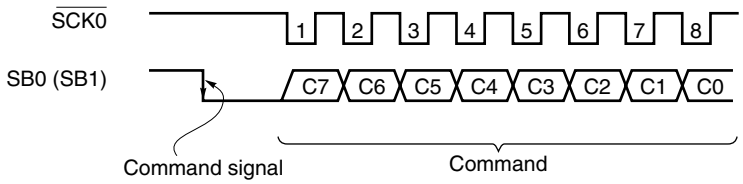
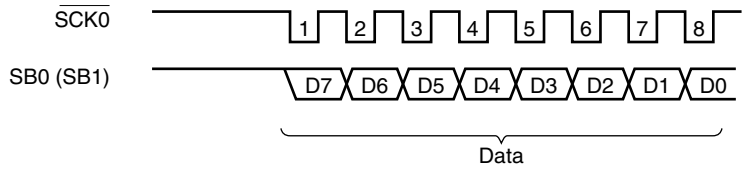


Figure 13-16. Data



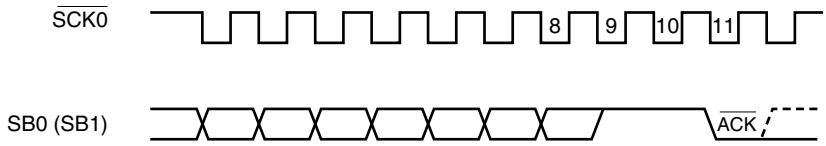
8-bit data following a command signal is defined as “command” data. 8-bit data without a command signal is defined as “data”. Command and data operation procedures can be determined by the user according to their communication specifications.

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

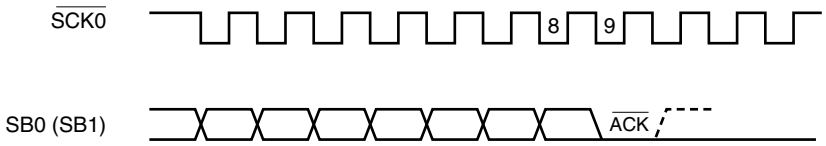
The acknowledge signal is used to check serial data reception between the transmitter and receiver.

Figure 13-17. Acknowledge Signal

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



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



Remark The broken line indicates the READY status.

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

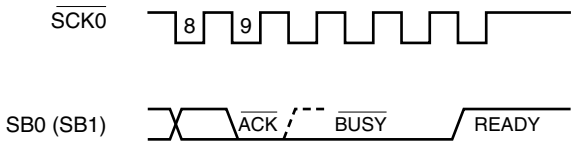
After 8-bit data transmission, the transmitter checks whether the receiver has returned the acknowledge signal. If the acknowledge signal is not returned for the preset period of time after data transmission, it can be judged that data reception has not been carried out correctly.

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

The $\overline{\text{BUSY}}$ signal is used to report to the master device that the slave device is not ready for data transmission/reception.

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

Figure 13-18. $\overline{\text{BUSY}}$ and READY Signals



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

$\overline{\text{BUSY}}$ signal output follows 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 the $\overline{\text{SCK0}}$ 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 SBI, after specifying reset of $\overline{\text{BUSY}}$, the $\overline{\text{BUSY}}$ signal is output until the fall of the next serial clock ($\overline{\text{SCK0}}$). If WUP = 1 is set during this interval by mistake, it will be impossible to reset $\overline{\text{BUSY}}$. Therefore, after $\overline{\text{BUSY}}$ is released, make sure that the SB0 (SB1) pin is high level before setting WUP = 1.

(3) Register setting

The SBI mode is set by serial operating mode register 0 (CSIM0), the serial bus interface control register (SBIC), and the interrupt timing specification 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	CSIM 04	CSIM 03	CSIM 02	CSIM 01	CSIM 00	FF60H	00H	R/W Note 1

R/W

CSIM 01	CSIM 00	Serial interface channel 0 clock selection
0	×	Input clock to $\overline{\text{SCK0}}$ pin from off-chip
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operating mode	Start bit	SI0/P25 pin function	SO0/P26 pin function	$\overline{\text{SCK0}}$/P27 pin function
0	×	3-wire serial I/O mode (refer to **13.4.2 3-wire serial I/O mode operation**)											
1	0	0	Note 2 ×	Note 2 ×	0	0	0	1	SBI mode	MSB	P25 (CMOS I/O)	SB1 (N-ch open-drain I/O)	$\overline{\text{SCK0}}$ (CMOS I/O)
		1	0	0	Note 2 ×	Note 2 ×	0	1			SB0 (N-ch open-drain I/O)	P26 (CMOS I/O)	
1	1	2-wire serial I/O mode (refer to **13.4.4 2-wire serial I/O mode operation**)											

R/W

WUP	Wakeup 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) 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. To use the wakeup function (WUP = 1), clear bit 5 (SIC) of the interrupt timing specification register (SINT) to 0.

4. When CSIE0 = 0, COI becomes 0.

Remark ×: don't care

PM××: Port mode register

P××: 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, the SO latch is set to 1. After SO latch setting, RELT is automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W

CMDT	Used for command signal output. When CMDT = 1, the SO latch is cleared to 0. After SO latch clearance, CMDT is automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R

RELD	Bus release detection	
Clear conditions (RELD = 0)		Set conditions (RELD = 1)
• When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied		• When bus release signal (REL) is detected

R

CMDD	Command detection	
Clear conditions (CMDD = 0)		Set conditions (CMDD = 1)
• When transfer start instruction is executed • When bus release signal (REL) is detected • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied		• When command signal (CMD) is detected

R/W

ACKT	The 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, ACKT is automatically cleared to 0. Used as ACKE = 0. Also cleared to 0 upon start of serial interface transfer or when CSIE0 = 0.

(continued)

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

Remarks 1. Bits 0, 1, and 4 (RELT, CMDT, and ACKT) are 0 when read after data setting.

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

(continued)

R/W

ACKE	Acknowledge signal output control	
0	Acknowledge signal automatic output disabled (output with ACKT enabled)	
1	Before completion of transfer	The acknowledge signal is output in synchronization with the 9th clock falling edge of $\overline{SCK0}$ (automatically output when ACKE = 1).
	After completion of transfer	The acknowledge signal is output in synchronization with the falling edge of $\overline{SCK0}$ just after execution of the instruction to be set to 1 (automatically output when ACKE = 1). However, ACKE is not automatically cleared to 0 after acknowledge signal output.

R

ACKD	Acknowledge detection	
Clear conditions (ACKD = 0)		Set conditions (ACKD = 1)
• At the falling edge of $\overline{SCK0}$ immediately after the busy mode has been released when a transfer start instruction is executed • When CSIE0 = 0 • When $\overline{RESET}$ input is applied		• When acknowledge signal ($\overline{ACK}$) is detected at the rising edge of $\overline{SCK0}$ clock after completion of transfer

R/W

BSYE[Note]	Synchronizing busy signal output control
0	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) is disabled.
1	Busy signal is output 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, the BSYE flag is not cleared to 0.

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

(c) Interrupt timing specification 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

SVAM	SVA bit to be used as slave address
0	Bits 0 to 7
1	Bits 1 to 7

R/W

SIC	INTCSI0 interrupt source selection[Note 2]
0	CSIIF0 is set upon termination of serial interface channel 0 transfer
1	CSIIF0 is set upon bus release detection or termination of serial interface channel 0 transfer

R

CLD	$\overline{\text{SCK0}}$ pin level[Note 3]
0	Low level
1	High level

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

2. When using the wakeup function in the SBI mode, set SIC to 0.

3. When CSIE0 = 0, CLD becomes 0.

Caution Be sure to set bits 0 to 3 to 0.

Remark SVA: Slave address register

CSIIF0: Interrupt request flag for INTCSI0

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

(4) Signals

Figures 13-19 to 13-24 show the signals and operations of the flags of the serial bus interface control register (SBIC) in SBI. Table 13-4 lists the signals in SBI.

Figure 13-19. RELT, CMDT, RELD, and CMDD Operations (Master)

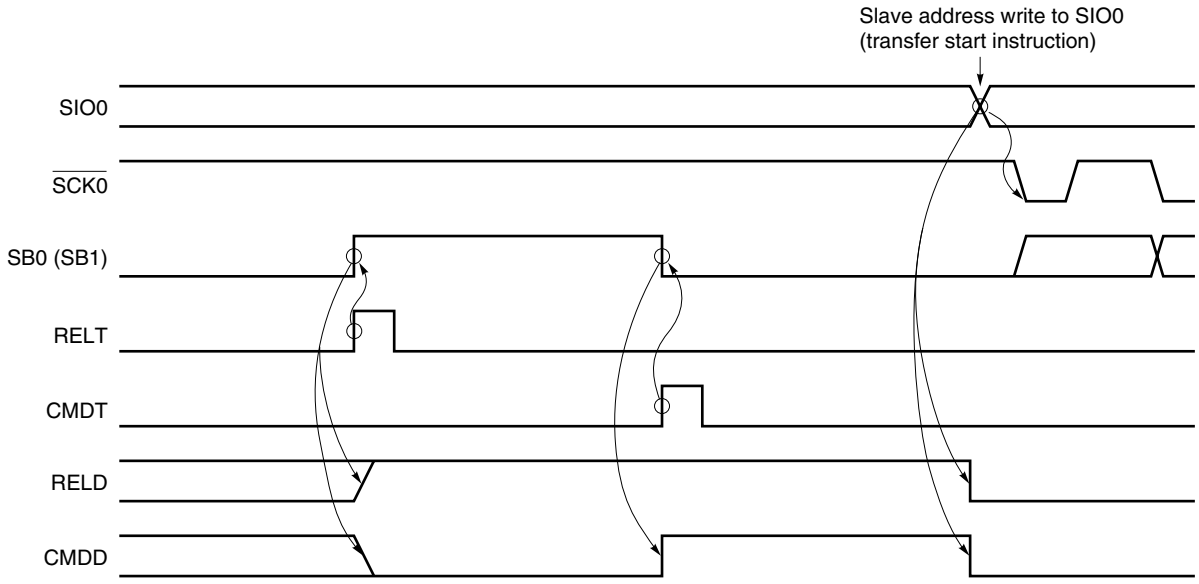


Figure 13-20. RELD and CMDD Operations (Slave)

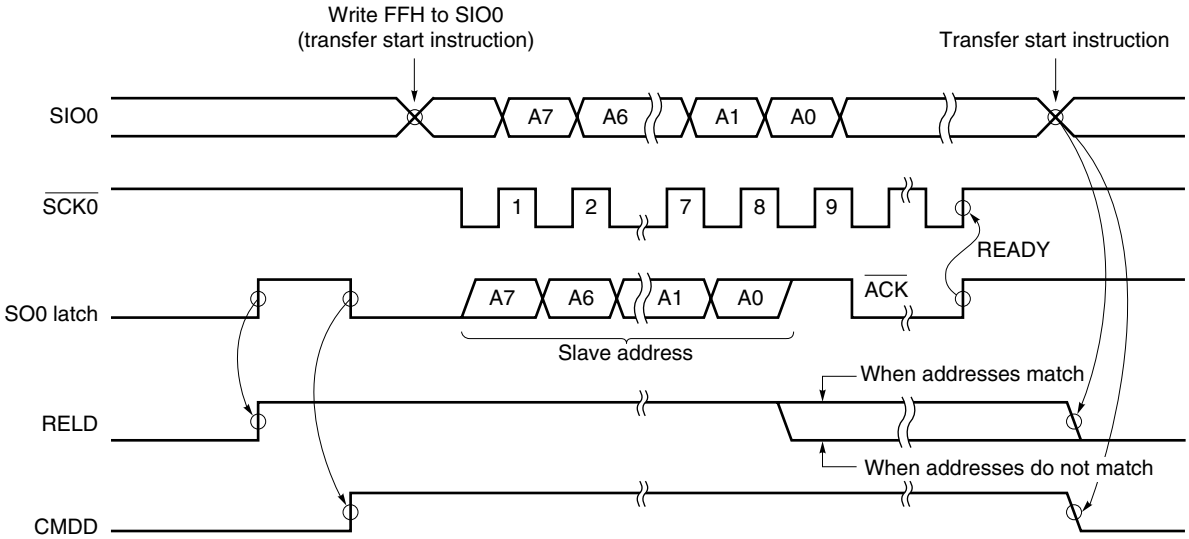
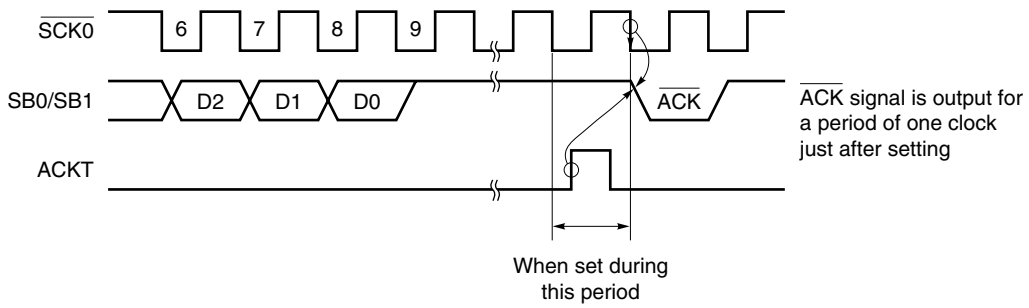


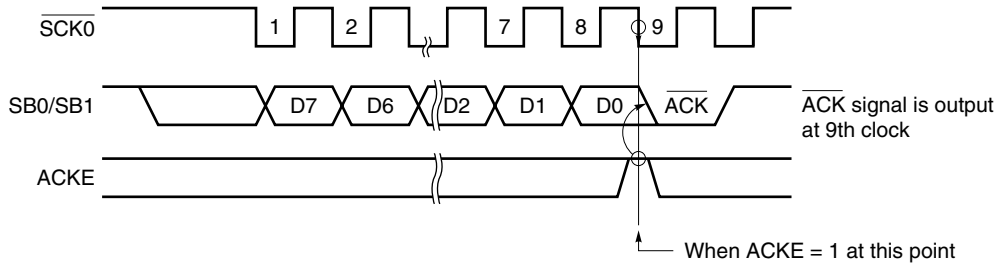
Figure 13-21. ACKT Operation



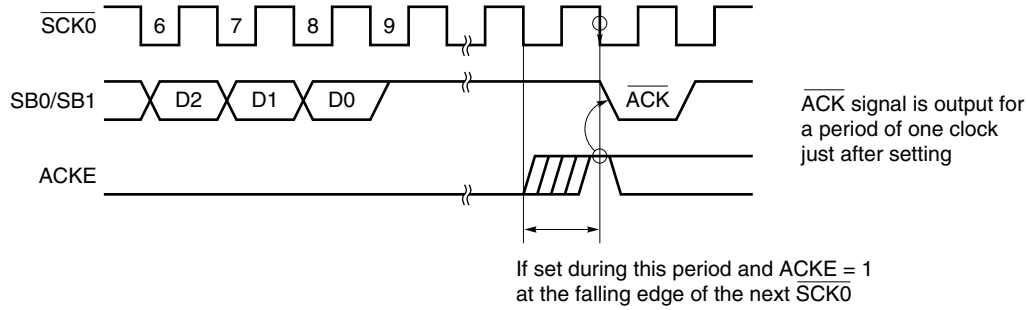
Caution Do not set ACKT before termination of transfer.

Figure 13-22. ACKE Operations

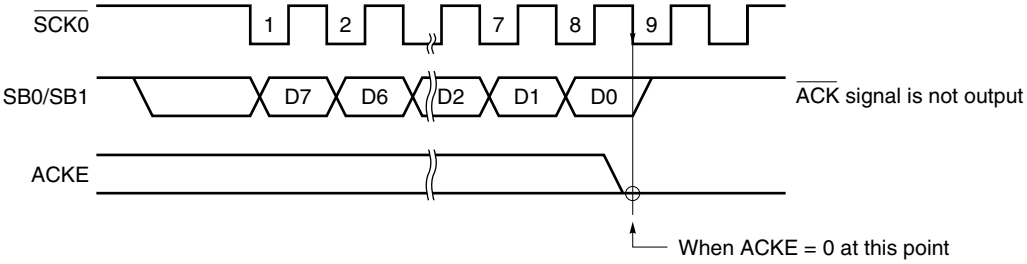
(a) When ACKE = 1 upon completion of transfer



(b) When set after completion of transfer



(c) When ACKE = 0 upon completion of transfer



(d) When ACKE = 1 period is short

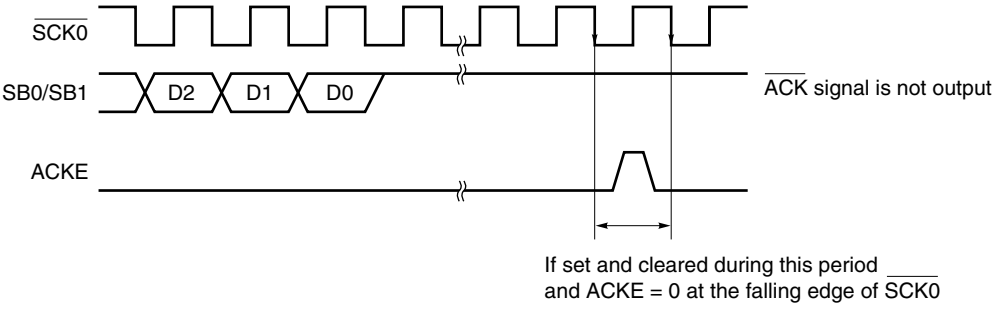
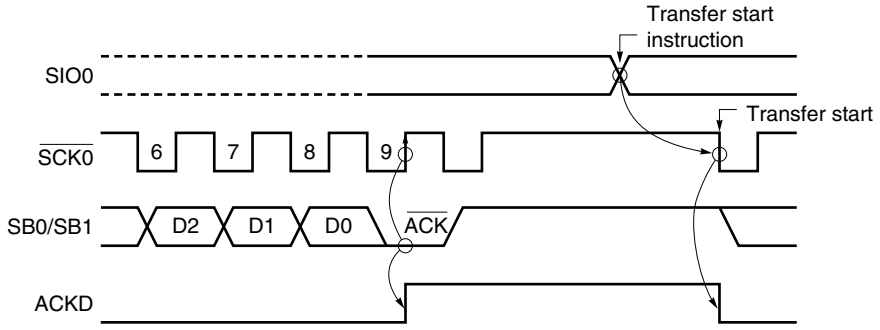
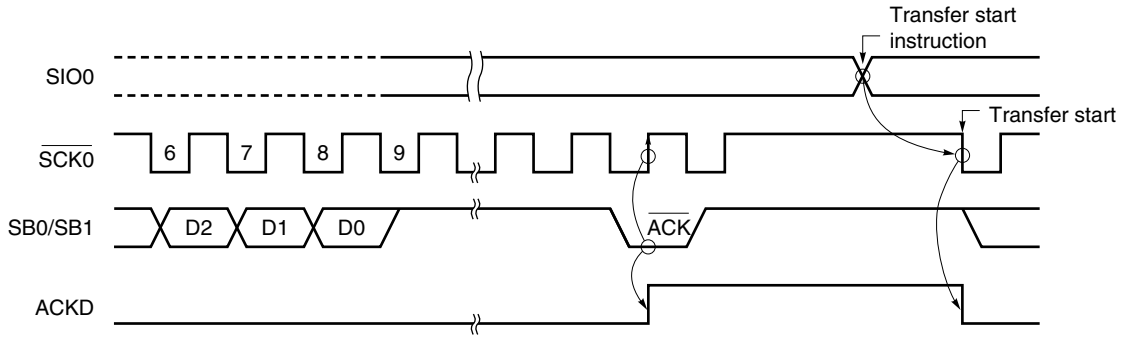


Figure 13-23. ACKD Operations

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



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



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

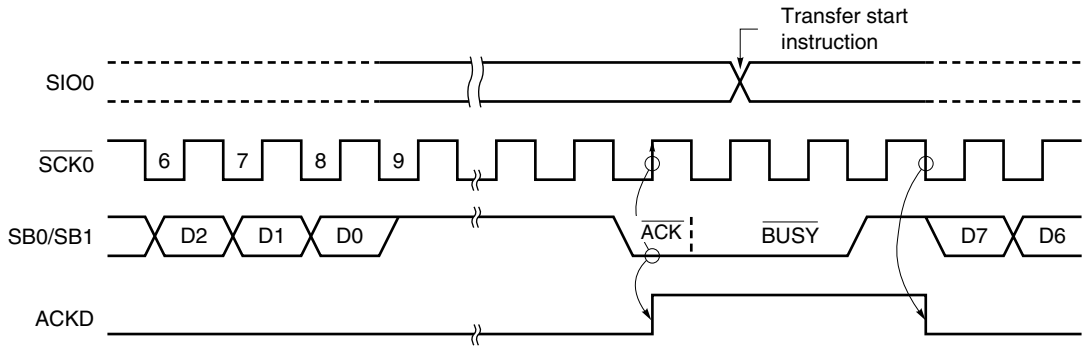


Figure 13-24. BSYE Operation

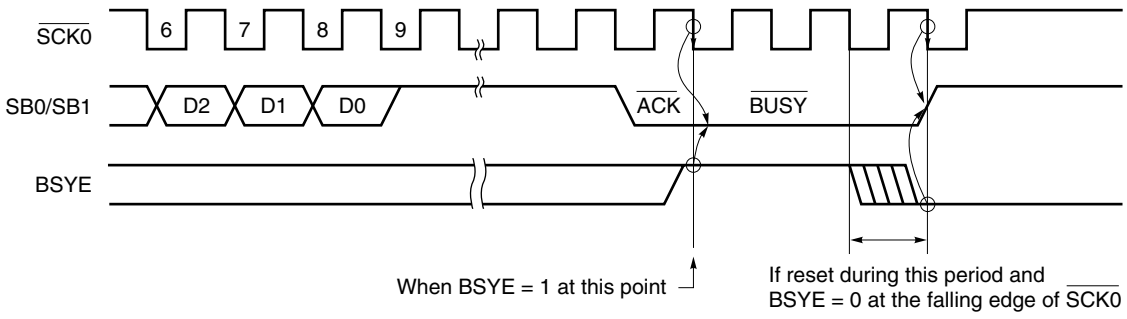


Table 13-4. Signals in SBI Mode (1/2)

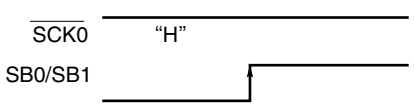
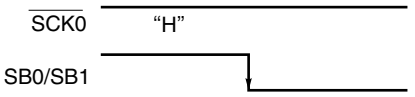
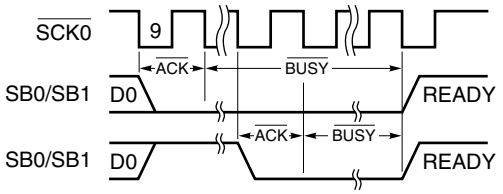
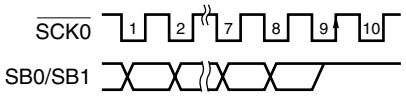
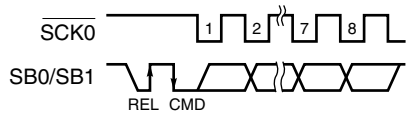
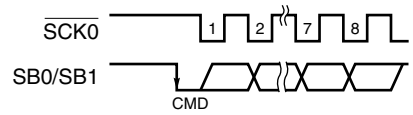
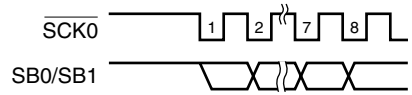
Signal Name	Output Device	Definition	Timing Chart	Output Condition	Effect on Flag	Meaning of Signal
Bus release signal (REL)	Master	SB0/SB1 rising edge when $\overline{SCK0}$ = 1		• RELT set	• RELD set • CMDD clear	CMD signal is output to indicate that transmit data is an address.
Command signal (CMD)	Master	SB0/SB1 falling edge when $\overline{SCK0}$ = 1		• CMDT set	• CMDD set	i) Transmit data is an address after REL signal output. ii) REL signal is not output and transmit data is a command.
Acknowledge signal ($\overline{ACK}$)	Master/ slave	Low-level signal 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	• ACKD set	Completion of reception
Busy signal ($\overline{BUSY}$)	Slave	[Synchronous BUSY signal] Low-level signal output to SB0/SB1 following acknowledge signal		• BSYE = 1	—	Serial receive disabled because of processing
Ready signal (READY)	Slave	High-level signal output to SB0/SB1 before serial transfer start and after completion of serial transfer		[1] BSYE = 0 [2] Execution of instruction data write to SIO0 (transfer start instruction)	—	Serial receive enabled

Table 13-4. Signals in SBI Mode (2/2)

Signal Name	Output Device	Definition	Timing Chart	Output Condition	Effect on Flag	Meaning of Signal
Serial clock ($\overline{SCK0}$)	Master	Synchronous clock to output address/command/data, $\overline{ACK}$ signal, synchronous $\overline{BUSY}$ signal, etc. Address/command/data are transferred with the first eight synchronous clocks.		When CSIE0 = 1, execution of instruction for data write to SIO0 (serial transfer start instruction)[Note 2]	CSIIF0 set (rising edge of 9th clock of $\overline{SCK0}$)[Note 1]	Timing of signal output to serial data bus
Address (A7 to A0)	Master	8-bit data transferred in synchronization with $\overline{SCK0}$ after output of REL and CMD signals				Address value of slave device on the serial bus
Command (C7 to C0)	Master	8-bit data transferred in synchronization with $\overline{SCK0}$ after output of only CMD signal without REL signal output				Instructions and messages to the slave device
Data (D7 to D0)	Master/ slave	8-bit data transferred in synchronization with $\overline{SCK0}$ without output of REL and CMD signals				Numeric values to be processed with slave or master device

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

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

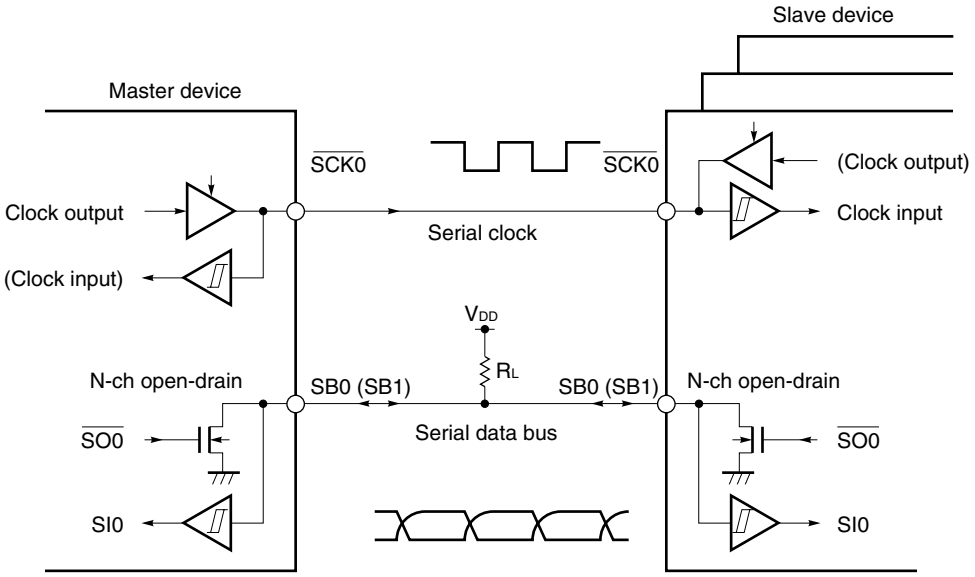
(5) Pin configuration

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

(a) $\overline{SCK0}$: Serial clock I/O pin
  [1] Master: CMOS and push-pull output
  [2] Slave: Schmitt input
(b) SB0 (SB1): Serial data I/O alternate-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 13-25. Pin Configuration



Caution Because the N-ch open-drain output must be high impedance at the time of data reception, write FFH to serial I/O shift register 0 (SIO0) in advance. The N-ch open-drain output can be high impedance throughout transfer. However, when the wakeup function specification bit (WUP) = 1, the N-ch open-drain output is always high impedance. Thus, it is not necessary to write FFH to SIO0.

(6) Address match detection method

In the SBI mode, the master transmits a slave address to select a specific slave device.

A match of the addresses can be automatically detected by hardware. CSIIF0 is set only when the slave address transmitted by the master matches the address set to SVA when the wakeup function specification bit (WUP) = 1.

If bit 5 (SIC) of the interrupt timing specification register (SINT) is set to 1, the wakeup function cannot be used even if WUP is set to 1 (an interrupt request signal is generated when bus release is detected). To use the wakeup 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, the match interrupt request (CSIIF0) of the address to be generated with WUP = 1 is normally used. Thus, execute selection/non-selection detection by slave address when WUP = 1.
2. When detecting selection/non-selection without the use of an 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, serial I/O shift register 0 (SIO0). Thus, transmit errors can be detected in the following two ways.

(a) Comparison of SIO0 data before transmission to that after transmission

In this case, if the two data differ, a transmit error is judged to have occurred.

(b) Use of the slave address register (SVA)

Transmit data is set to both SIO0 and SVA and is transmitted. After termination of transmission, the COI bit (match signal coming from the address comparator) of 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 normally selects one slave device as the communication target from among two or more devices by outputting an “address” to the serial bus.

After the communication target device has been determined, commands and data are transmitted/ received and serial communication is realized between the master and slave devices.

Figures 13-26 to 13-29 show data communication timing charts.

Shift operations of serial I/O shift register 0 (SIO0) are carried out at the falling edge of the serial clock ($\overline{SCK0}$). Transmit data is latched into the SO0 latch and is output with the MSB set as the start 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 SIO0.

Figure 13-26. Address Transmission from Master Device to Slave Device (WUP = 1)

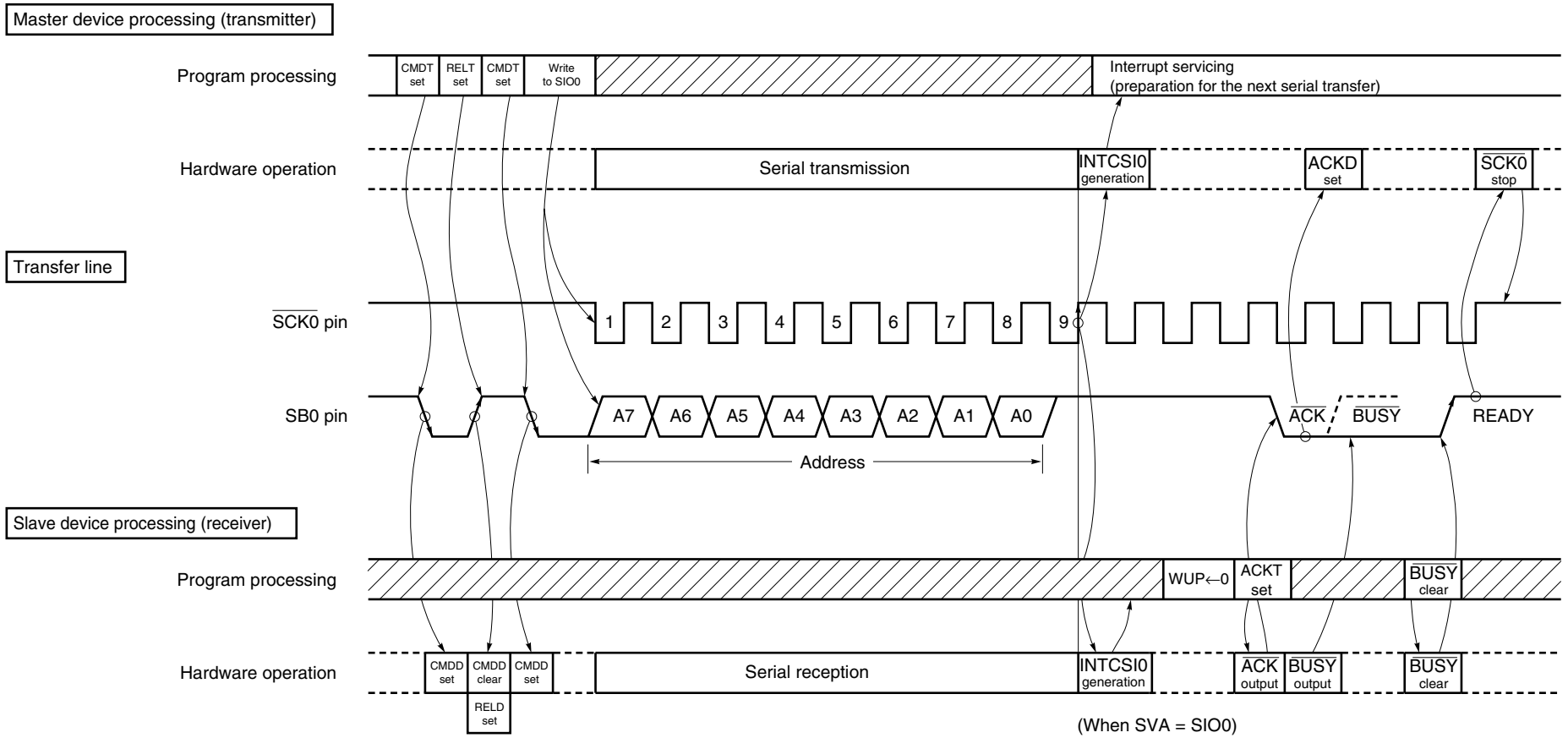


Figure 13-27. Command Transmission from Master Device to Slave Device

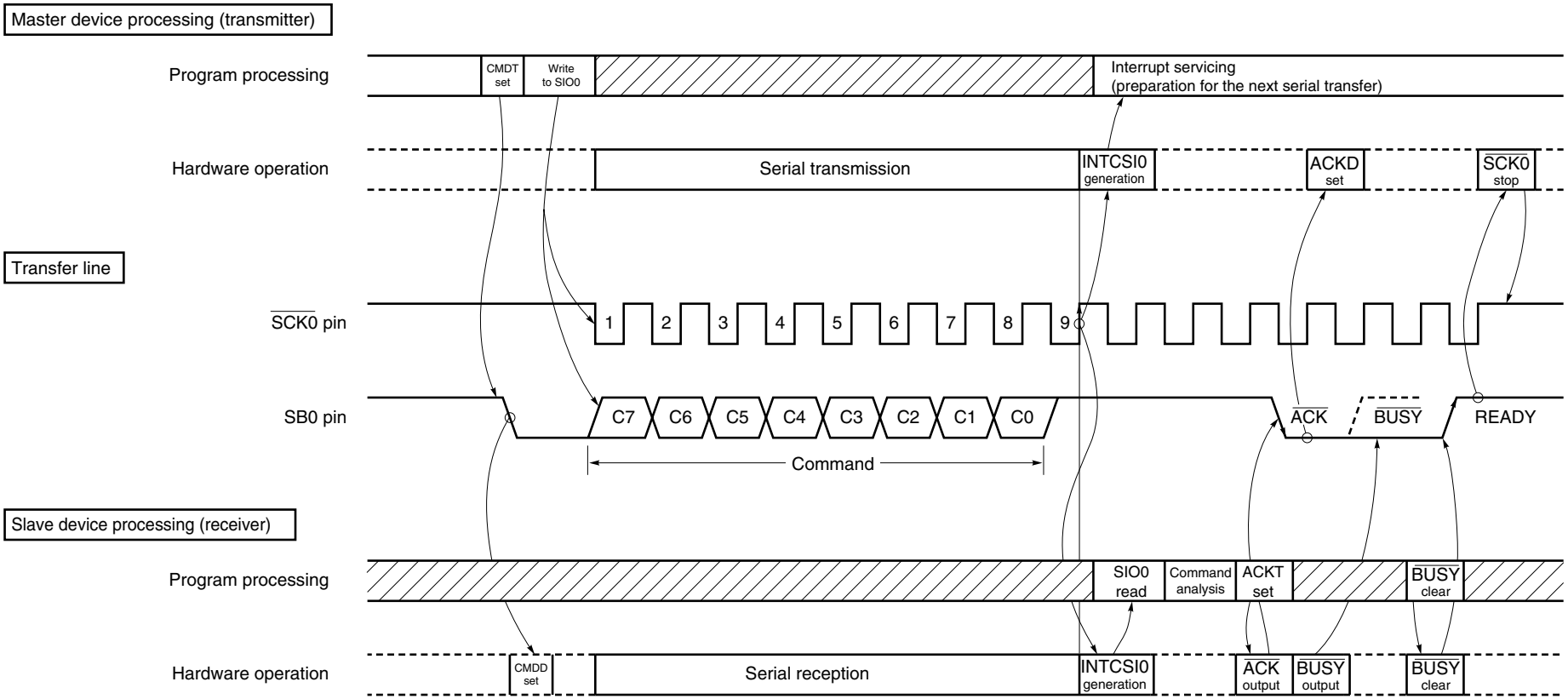


Figure 13-28. Data Transmission from Master Device to Slave Device

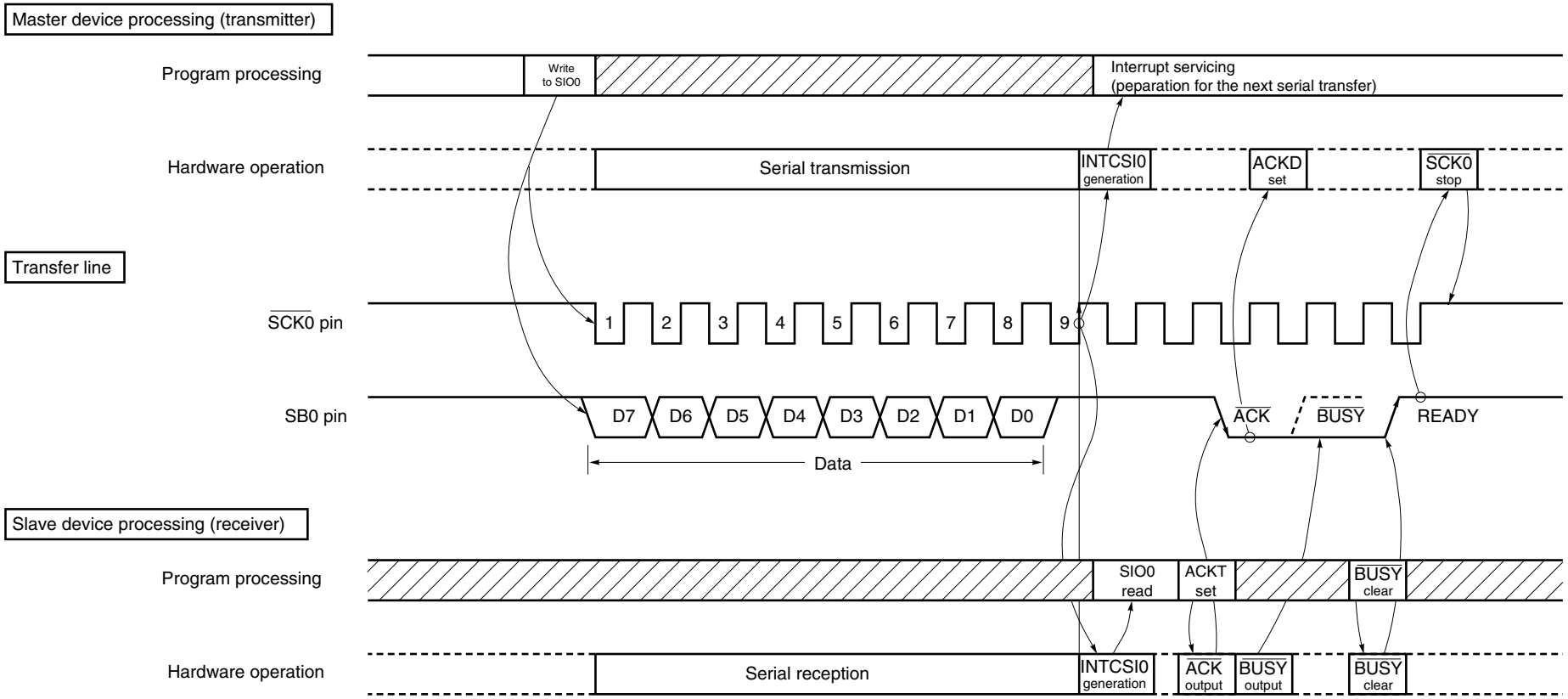
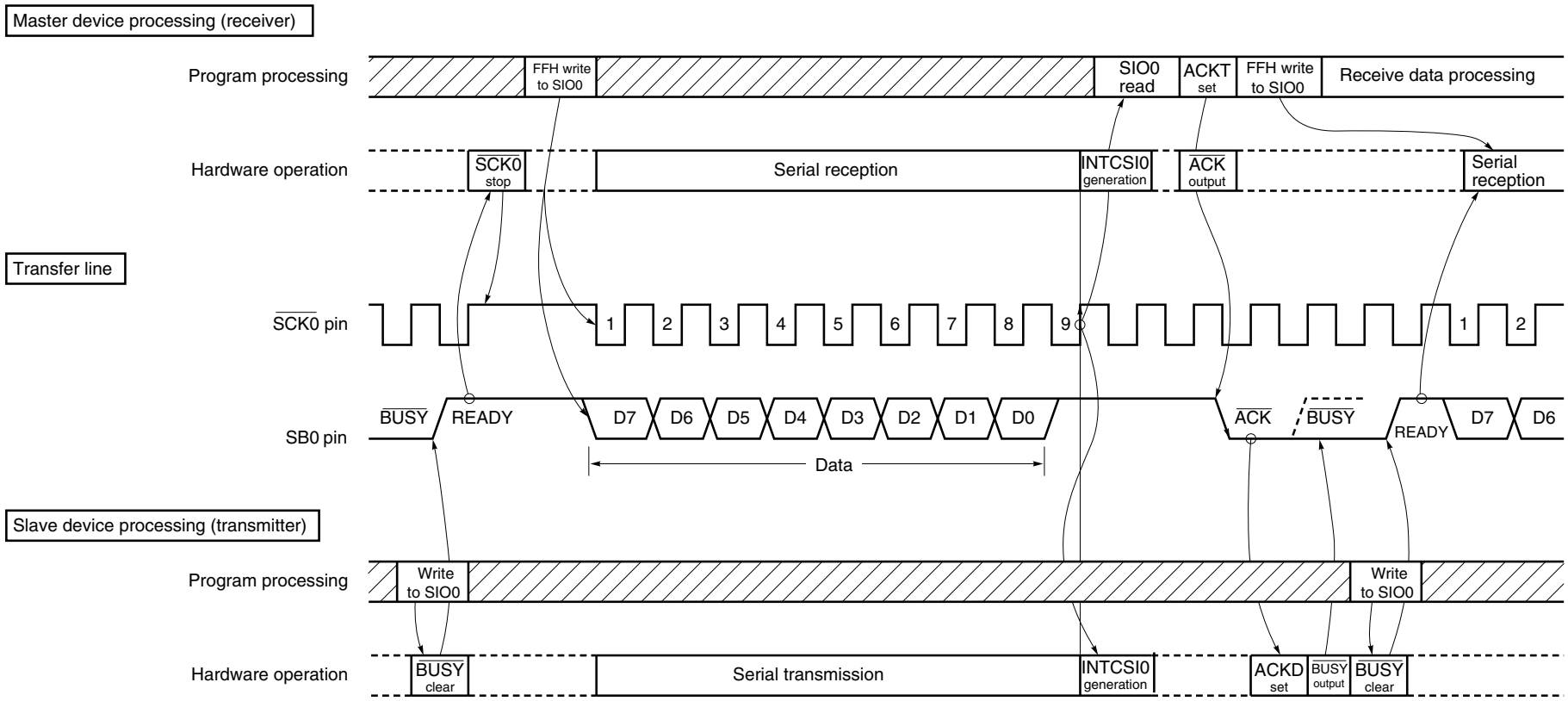


Figure 13-29. Data Transmission from Slave Device to Master Device



(9) Transfer start

Serial transfer is started by setting transfer data to serial I/O shift register 0 (SIO0) when the following two conditions are satisfied.

- Serial interface channel 0 operation control bit (CSIE0) = 1
- Internal serial clock is stopped or $\overline{SCK0}$ is at high level after 8-bit serial transfer.

Cautions 1. If CSIE0 is set to "1" after data write to SIO0, transfer does not start.

2. Because the N-ch open-drain output must be made to go into a high-impedance state during data reception, write FFH to SIO0 in advance. However, when the wakeup function specification bit (WUP) = 1, the N-ch open-drain output always goes into a high-impedance state. Thus, it is not necessary to write FFH to SIO0.

3. If data is written to SIO0 when the slave is busy, the data is not lost.
When the busy state is cleared and SB0 (or SB1) input is set to the high level (READY) state, transfer starts.

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

For the pin that is to be used for data I/O (SB0 or SB1), be sure to set as follows before serial transfer of the 1st byte after $\overline{RESET}$ input.

[1] Set the P25 and P26 output latches to 1.

[2] Set bit 0 (RELT) of the serial bus interface control register (SBIC) to 1.

[3] Reset the P25 and P26 output latches from 1 to 0.

(10) Judging busy status of slave

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

[1] Detect acknowledge signal ($\overline{ACK}$) or interrupt request signal generation.

[2] Set the port mode register PM25 (or PM26) of the SB0/P25 (or SB1/P26) pin to the input mode.

[3] Read out the pin state (when the pin level is high, the READY state is set).

After 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, the match interrupt (CSIIF0) of the address to be generated with WUP = 1 is normally used. Thus, execute selection/non-selection detection by slave address when WUP = 1.

(b) When detecting selection/non-selection without the use of an 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 SBI, after specifying reset of $\overline{BUSY}$, the $\overline{BUSY}$ signal is output until the fall of the next serial clock ($\overline{SCK0}$). If WUP = 1 is set during this interval by mistake, it will be impossible to reset $\overline{BUSY}$. Therefore, after $\overline{BUSY}$ is released, make sure that the SB0 (SB1) pin is high level before setting WUP = 1.

(d) For the pin that is to be used for data I/O, be sure to set as follows before serial transfer of the 1st byte after $\overline{\text{RESET}}$ input.

[1] Set the P25 and P26 output latches to 1.

[2] Set bit 0 (RELT) of the serial bus interface control register (SBIC) to 1.

[3] Reset the P25 and P26 output latches from 1 to 0.

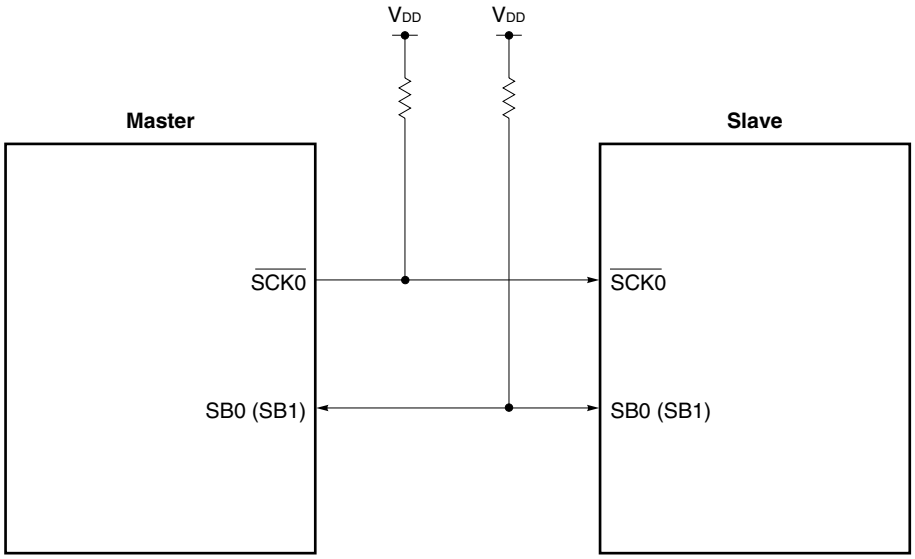
(e) If the SB0 (SB1) line changes from low level to high level or from high level to low level while the $\overline{\text{SCK0}}$ line is high level, it is recognized as either a bus release signal or a command signal. Therefore, if the changing timing of the bus fluctuates because of substrate capacitance, etc., it may be recognized as a bus release signal (or a command signal) even while data is being transmitted. Care should therefore be taken in the wiring.

13.4.4 2-wire serial I/O mode operation

The 2-wire serial I/O mode can handle any communication format by program.

Communication is basically carried out using two lines: a serial clock ($\overline{\text{SCK0}}$) and serial data input/output (SB0 or SB1).

Figure 13-30. Serial Bus Configuration Example Using 2-Wire Serial I/O Mode



(1) Register setting

The 2-wire serial I/O mode is set by serial operating mode register 0 (CSIM0), the serial bus interface control register (SBIC), and the interrupt timing specification 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	CSIM 04	CSIM 03	CSIM 02	CSIM 01	CSIM 00	FF60H	00H	R/W[Note 1]

R/W

CSIM 01	CSIM 00	Serial interface channel 0 clock selection
0	×	Input clock to $\overline{\text{SCK0}}$ pin from off-chip
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operating mode	Start bit	SI0/P25 pin function	SO0/P26 pin function	$\overline{\text{SCK0}}$/P27 pin function
0	×	3-wire serial I/O mode (refer to **13.4.2 3-wire serial I/O mode operation**)											
1	0	SBI mode (refer to **13.4.3 SBI mode operation**)											
1	1	0	Note 2 ×	Note 2 ×	0	0	0	1	2-wire serial I/O mode	MSB	P25 (CMOS I/O)	SB1 (N-ch open-drain I/O)	$\overline{\text{SCK0}}$ (N-ch open-drain I/O)
		1	0	0	Note 2 ×	Note 2 ×	0	1			SB0 (N-ch open-drain I/O)	P26 (CMOS I/O)	

R/W

WUP	Wakeup 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) 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 ×: don't care

PM××: Port mode register

P××: 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, the SO latch is set to 1. After SO latch setting, RELT is automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W		
	CMDT	When CMDT = 1, the SO latch is cleared to 0. After SO latch clearance, CMDT is automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

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

(c) Interrupt timing specification 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

SVAM	SVA bit to be used as slave address
0	Bits 0 to 7
1	Bits 1 to 7

R/W

SIC	INTCSI0 interrupt source selection
0	CSIIF0 is set upon termination of serial interface channel 0 transfer
1	CSIIF0 is set upon bus release detection or termination of serial interface channel 0 transfer

R

CLD	$\overline{\text{SCK0}}$ pin level[Note 2]
0	Low level
1	High level

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

2. When CSIE0 = 0, CLD becomes 0.

Caution Be sure to set bits 0 to 3 to 0.

Remark SVA: Slave address register

CSIIF0: Interrupt request flag for INTCSI0

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

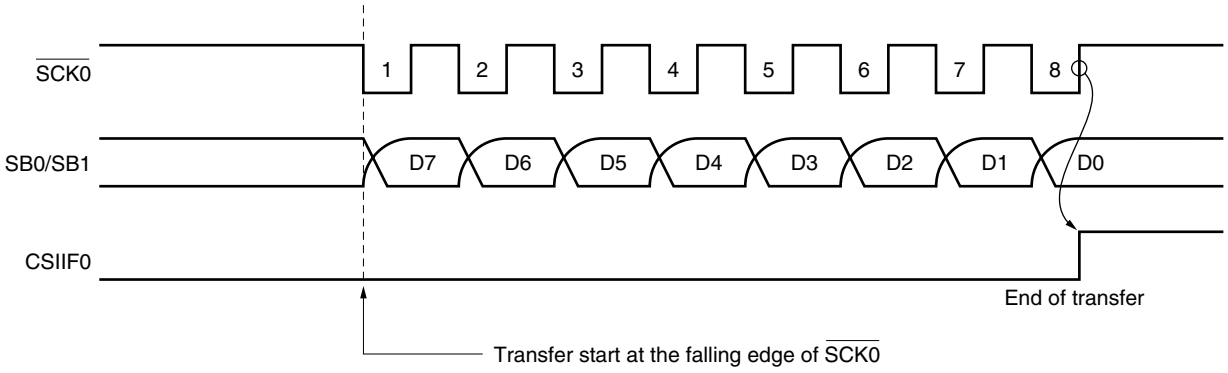
(2) Communication operation

The 2-wire serial I/O mode is used for data transmission/reception in 8-bit units. Data transmission/reception is carried out bit by bit in synchronization with the serial clock.

Shift operations of serial I/O shift register 0 (SIO0) are carried out in synchronization with the falling edge of the serial clock ($\overline{SCK0}$). The transmit data is held in the SO0 latch and is output from the SB0/P25 (or SB1/P26) pin with the MSB set as the start bit. The receive data input from the SB0 (or SB1) pin is latched into SIO0 at the rising edge of $\overline{SCK0}$.

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

Figure 13-31. 2-Wire Serial I/O Mode Timing



The SB0 (SB1) pin specified for the serial data bus is an N-ch open-drain I/O and thus it must be externally pulled up. Because the N-ch open-drain output must be high impedance for data reception, write FFH to SIO0 in advance.

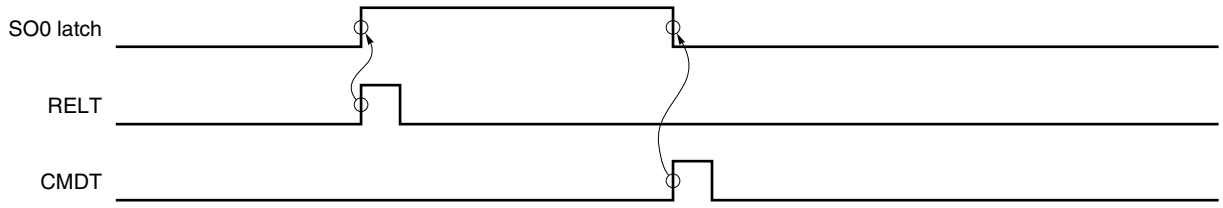
The SB0 (or SB1) pin generates the SO0 latch status and thus the SB0 (or SB1) pin output status can be manipulated by setting bit 0 (RELT) and bit 1 (CMDT) of the serial bus interface control register (SBIC). However, do not carry out this manipulation during serial transfer.

Control the $\overline{SCK0}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **13.4.5 $\overline{SCK0}$/P27 pin output manipulation**).

(3) Signals

Figure 13-32 shows the RELT and CMDT operations.

Figure 13-32. RELT and CMDT Operations



(4) Transfer start

Serial transfer is started by setting transfer data to 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 for data reception, write FFH to SIO0 in advance.

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

(5) Error detection

In the 2-wire serial I/O mode, the serial bus SB0 (SB1) status being transmitted is fetched into serial I/O shift register 0 (SIO0) of the transmitting device. Thus, transmit errors can be detected in the following two ways.

(a) Comparison of SIO0 data before transmission to that after transmission

In this case, if the two data differ, a transmit error is judged to have occurred.

(b) Use of the slave address register (SVA)

Transmit data is set to both SIO0 and SVA and is transmitted. After termination of transmission, the COI bit (match signal coming from the address comparator) of 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.

13.4.5 $\overline{\text{SCK0}}$/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.

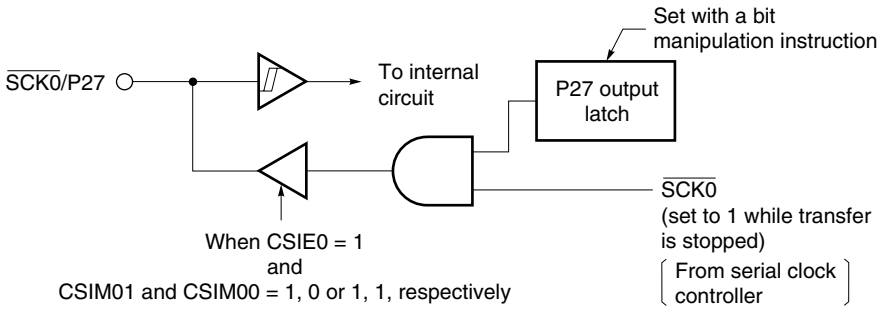
P27 output latch manipulation enables any value of $\overline{\text{SCK0}}$ to be set by software (SI0/SB0 and SO0/SB1 pins to be controlled with bit 0 (RELT) and bit 1 (CMDT) of the serial bus interface control register (SBIC)).

The $\overline{\text{SCK0}}$/P27 pin output manipulation procedure is described below.

[1] Set 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 13-33. $\overline{\text{SCK0}}$/P27 Pin Configuration



CHAPTER 14 SERIAL INTERFACE CHANNEL 1

14.1 Functions of Serial Interface Channel 1

Serial interface channel 1 has the following three modes.

Table 14-1. Modes of Serial Interface Channel 1

Operation Mode	Pins Used	Features	Usage
Operation stop mode	–	• Used when serial transfer is not carried out. • Power consumption can be reduced.	–
3-wire serial I/O mode (MSB-/LSB-first switchable)	$\overline{SCK1}$ (serial clock), SO1 (serial output), SI1 (serial input)	• Input and output lines are independent and they can transfer/receive at the same time, so the data transfer processing time is short. • The start bit of 8-bit data to undergo serial transfer is switchable between MSB and LSB.	These modes are used for connection of peripheral ICs and display controllers that incorporate a clocked serial interface.
3-wire serial I/O mode with automatic transmit/ receive function (MSB-/LSB-first switchable)	$\overline{SCK1}$ (serial clock), SO1 (serial output), SI1 (serial input)	• Mode with same function as 3-wire serial I/O mode above plus automatic transmit/receive function. • Can transmit/receive data with a maximum of 64 bytes. Therefore, this function enables the hardware to transmit/receive data to/from the OSD (On Screen Display) device and device with on-chip display controller/driver independently of the CPU thus the software load can be reduced.	

14.2 Configuration of Serial Interface Channel 1

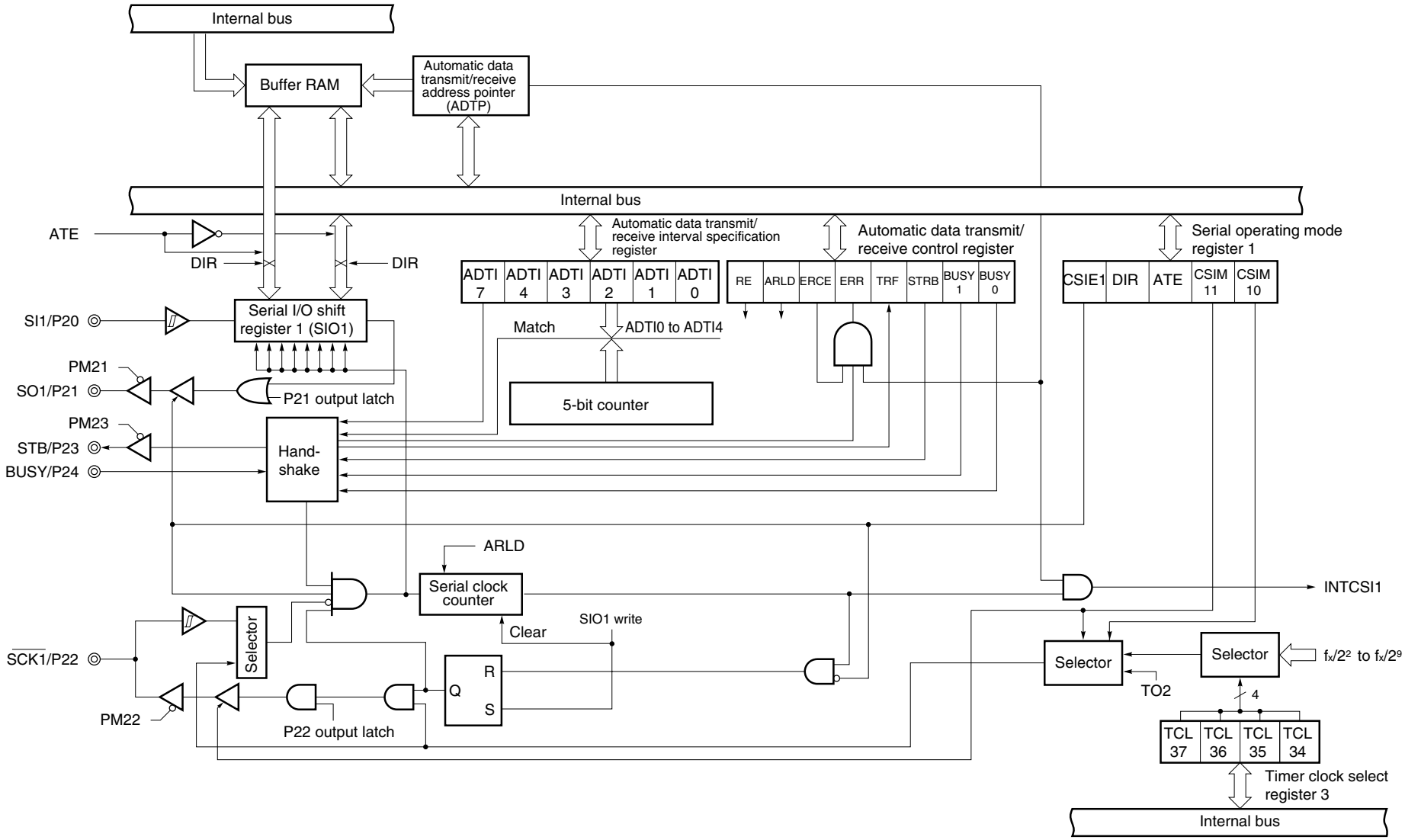
Serial interface channel 1 consists of the following hardware.

Table 14-2. Configuration of Serial Interface Channel 1

Item	Configuration
Registers	Serial I/O shift register 1 (SIO1) Automatic data transmit/receive address pointer (ADTP)
Control registers	Timer clock select register 3 (TCL3) Serial operating mode register 1 (CSIM1) Automatic data transmit/receive control register (ADTC) Automatic data transmit/receive interval specification register (ADTI) Port mode register 2 (PM2)[Note]

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

Figure 14-1. Block Diagram of Serial Interface Channel 1



(1) Serial I/O shift register 1 (SIO1)

This is an 8-bit register used to carry out parallel/serial conversion and serial transmission/reception (shift operations) in synchronization with the serial clock.

SIO1 is set with an 8-bit memory manipulation instruction.

When 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 of (the number of transmit data bytes – 1) while the automatic transmit/receive function is activated. It is decremented automatically with data transmission/reception.

ADTP is set with an 8-bit memory manipulation instruction. The higher 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 and checks whether 8-bit data has been transmitted/received.

14.3 Control Registers of Serial Interface Channel 1

The following four registers are used to control serial interface channel 1.

- Timer clock select register 3 (TCL3)
- Serial operating mode register 1 (CSIM1)
- Automatic data transmit/receive control register (ADTC)
- Automatic data transmit/receive interval specification register (ADTI)

(1) Timer clock select register 3 (TCL3)

This register sets the serial clock of serial interface channel 1.

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

$\overline{\text{RESET}}$ 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 14-2. Format of Timer Clock Select Register 3

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33	TCL32	TCL31	TCL30	Serial interface channel 0 serial clock selection
0	1	1	0	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_X/2^9$ (9.8 kHz)
Other than above				Setting prohibited

TCL37	TCL36	TCL35	TCL34	Serial interface channel 1 serial clock selection
0	1	1	0	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_X/2^9$ (9.8 kHz)
Other than above				Setting prohibited

Caution If TCL3 is to be rewritten with data that is not identical, stop the serial transfer first.

Remarks 1. f_X: Main system clock oscillation frequency

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

(2) Serial operating mode register 1 (CSIM1)

This register sets the serial interface channel 1 serial clock, operating mode, operation enable/stop, and automatic transmit/receive operation enable/stop.

CSIM1 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM1 to 00H.

Figure 14-3. Format of Serial Operating Mode Register 1

Symbol	<7>	6	<5>	4	3	2	1	0	Address	After reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM 11	CSIM 10	FF68H	00H	R/W

CSIM 11	CSIM 10	Serial interface channel 1 clock selection
0	×	Clock externally input to $\overline{\text{SCK1}}$ pin[Note 1]
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 4 to 7 of timer clock select register 3 (TCL3)

ATE	Serial interface channel 1 operating mode selection
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

DIR	Start bit	SI1 pin function	SO1 pin function
0	MSB	SI1/P20 (input)	SO1 (CMOS output)
1	LSB		

CSIE1	CSIM 11	PM20	P20	PM21	P21	PM22	P22	Shift register 1 operation	Serial clock counter operation control	SI1/P20 pin function	SO1/P21 pin function	$\overline{\text{SCK1}}$/P22 pin function
0	×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Operation stop	Clear	P20 (CMOS I/O)	P21 (CMOS I/O)	P22 (CMOS I/O)
1	0	Note 3 1	Note 3 ×	0	0	1	×	Operation enable	Count operation	SI1[Note 3] (input)	SO1 (CMOS output)	$\overline{\text{SCK1}}$ (input)
	1					0	1					$\overline{\text{SCK1}}$ (CMOS output)

Notes 1. If 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 a port pin.

3. Can be used as P20 when used only for transmission (set bit 7 (RE) of ADTC to 0).

Remark ×: don't care

PM××: Port mode register

P××: Port output latch

(3) Automatic data transmit/receive control register (ADTC)

This register sets automatic receive enable/disable, the operating mode, strobe output enable/disable, busy input enable/disable, error check enable/disable, and displays automatic transmit/receive execution and error detection.

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

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

Figure 14-4. Format of Automatic Data Transmit/Receive Control Register

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
ADTC	RE	ARLD	ERCE	ERR	TRF	STRB	BUSY1	BUSY0	FF69H	00H	R/W[Note 1]

R/W

BUSY1	BUSY0	Busy input control
0	×	Not using busy input
1	0	Busy input enabled (active high)
1	1	Busy input enabled (active low)

R/W

STRB	Strobe output control
0	Strobe output disabled
1	Strobe output enabled

R

TRF	Status of automatic transmit/receive function[Note 2]
0	Detection of termination of automatic transmission/ reception (This bit is set to 0 upon suspension of automatic transmission/reception or when ARLD = 0.)
1	During automatic transmission/reception (This bit is set to 1 when data is written to SIO1.)

R

ERR	Error detection of automatic transmit/receive function
0	No error (This bit is set to 0 when data is written to SIO1.)
1	Error occurred

R/W

ERCE	Error check control of automatic transmit/ receive function
0	Error check disabled
1	Error check enabled (only when BUSY1 = 1)

R/W

ARLD	Operating mode selection of automatic transmit/ receive function
0	Single operating mode
1	Repetitive operating mode

R/W

RE	Receive control of automatic transmit/receive function
0	Receive disabled
1	Receive enabled

Notes 1. Bits 3 and 4 (TRF and ERR) are read-only bits.

2. The termination of automatic transmission/reception should be judged by using TRF, not CSIIF1 (interrupt request flag).

Caution When external clock input is selected with bit 1 (CSIM11) of serial operating mode register 1 (CSIM1) set to 0, set STRB and BUSY1 of ADTC to 0, 0.

Remark ×: don't care

(4) Automatic data transmit/receive interval specification register (ADTI)

This register sets the automatic 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.

Figure 14-5. Format of Automatic Data Transmit/Receive Interval Specification Register (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI7	Data transfer interval control
0	No control of interval by ADTI[Note 1]
1	Control of interval by ADTI (ADTI0 to ADTI4)

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data transfer interval specification (f_X = 5.0 MHz operation)	
					Minimum[Note 2]	Maximum[Note 2]
0	0	0	0	0	36.8 μs + 0.5/f_{SCK}	40.0 μs + 1.5/f_{SCK}
0	0	0	0	1	62.4 μs + 0.5/f_{SCK}	65.6 μs + 1.5/f_{SCK}
0	0	0	1	0	88.0 μs + 0.5/f_{SCK}	91.2 μs + 1.5/f_{SCK}
0	0	0	1	1	113.6 μs + 0.5/f_{SCK}	116.8 μs + 1.5/f_{SCK}
0	0	1	0	0	139.2 μs + 0.5/f_{SCK}	142.4 μs + 1.5/f_{SCK}
0	0	1	0	1	164.8 μs + 0.5/f_{SCK}	168.0 μs + 1.5/f_{SCK}
0	0	1	1	0	190.4 μs + 0.5/f_{SCK}	193.6 μs + 1.5/f_{SCK}
0	0	1	1	1	216.0 μs + 0.5/f_{SCK}	219.2 μs + 1.5/f_{SCK}
0	1	0	0	0	241.6 μs + 0.5/f_{SCK}	244.8 μs + 1.5/f_{SCK}
0	1	0	0	1	267.2 μs + 0.5/f_{SCK}	270.4 μs + 1.5/f_{SCK}
0	1	0	1	0	292.8 μs + 0.5/f_{SCK}	296.0 μs + 1.5/f_{SCK}
0	1	0	1	1	318.4 μs + 0.5/f_{SCK}	321.6 μs + 1.5/f_{SCK}
0	1	1	0	0	344.0 μs + 0.5/f_{SCK}	347.2 μs + 1.5/f_{SCK}
0	1	1	0	1	369.6 μs + 0.5/f_{SCK}	372.8 μs + 1.5/f_{SCK}
0	1	1	1	0	395.2 μs + 0.5/f_{SCK}	398.4 μs + 1.5/f_{SCK}
0	1	1	1	1	420.8 μs + 0.5/f_{SCK}	424.0 μs + 1.5/f_{SCK}

Notes 1. The interval is dependent only on CPU processing.

2. The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if the minimum calculated by the following expression is smaller than 2/f_{SCK}, the minimum interval time is 2/f_{SCK}.

$$\text{Minimum} = (n + 1) \times \frac{2^7}{fx} + \frac{56}{fx} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^7}{fx} + \frac{72}{fx} + \frac{1.5}{f_{SCK}}$$

Cautions 1. ADTI should not be written to during operation of the automatic transmit/receive function.

2. Bits 5 and 6 must be set to 0.

3. When ADTI is used to control the interval time of data transfer by automatic transmit/receive function, busy control (refer to 14.4.3 (4) (a) Busy control option) is invalid.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{SCK}: Serial clock frequency

Figure 14-5. Format of Automatic Data Transmit/Receive Interval Specification Register (2/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data transfer interval specification (f_X = 5.0 MHz operation)	
					Minimum[Note]	Maximum[Note]
1	0	0	0	0	446.4 μs + 0.5/f_{SCK}	449.6 μs + 1.5/f_{SCK}
1	0	0	0	1	472.0 μs + 0.5/f_{SCK}	475.2 μs + 1.5/f_{SCK}
1	0	0	1	0	497.6 μs + 0.5/f_{SCK}	500.8 μs + 1.5/f_{SCK}
1	0	0	1	1	523.2 μs + 0.5/f_{SCK}	526.4 μs + 1.5/f_{SCK}
1	0	1	0	0	548.8 μs + 0.5/f_{SCK}	552.0 μs + 1.5/f_{SCK}
1	0	1	0	1	574.4 μs + 0.5/f_{SCK}	577.6 μs + 1.5/f_{SCK}
1	0	1	1	0	600.0 μs + 0.5/f_{SCK}	603.2 μs + 1.5/f_{SCK}
1	0	1	1	1	625.6 μs + 0.5/f_{SCK}	628.8 μs + 1.5/f_{SCK}
1	1	0	0	0	651.2 μs + 0.5/f_{SCK}	654.4 μs + 1.5/f_{SCK}
1	1	0	0	1	676.8 μs + 0.5/f_{SCK}	680.0 μs + 1.5/f_{SCK}
1	1	0	1	0	702.4 μs + 0.5/f_{SCK}	705.6 μs + 1.5/f_{SCK}
1	1	0	1	1	728.0 μs + 0.5/f_{SCK}	731.2 μs + 1.5/f_{SCK}
1	1	1	0	0	753.6 μs + 0.5/f_{SCK}	756.8 μs + 1.5/f_{SCK}
1	1	1	0	1	779.2 μs + 0.5/f_{SCK}	782.4 μs + 1.5/f_{SCK}
1	1	1	1	0	804.8 μs + 0.5/f_{SCK}	808.0 μs + 1.5/f_{SCK}
1	1	1	1	1	830.4 μs + 0.5/f_{SCK}	833.6 μs + 1.5/f_{SCK}

Note The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if the minimum calculated by the following expression is smaller than 2/f_{SCK}, the minimum interval time is 2/f_{SCK}.

$$\text{Minimum} = (n + 1) \times \frac{2^7}{f_X} + \frac{56}{f_X} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^7}{f_X} + \frac{72}{f_X} + \frac{1.5}{f_{SCK}}$$

Cautions 1. ADTI should not be written to during operation of the automatic transmit/receive function.

2. Bits 5 and 6 must be set to 0.

3. When ADTI is used to control the interval time of data transfer by automatic transmit/receive function, busy control (refer to 14.4.3 (4) (a) Busy control option) is invalid.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{SCK}: Serial clock frequency

14.4 Operations of Serial Interface Channel 1

The following three operating modes are available for serial interface channel 1.

- Operation stop mode
- 3-wire serial I/O mode
- 3-wire serial I/O mode with automatic transmit/receive function

14.4.1 Operation stop mode

Serial transfer is not carried out in the operation stop mode. Thus, power consumption can be reduced. Serial I/O shift register 1 (SIO1) does not carry out shift operations and can be used as an ordinary 8-bit register.

In the operation stop mode, the P20/SI1, P21/SO1, P22/$\overline{SCK1}$, P23/STB, and P24/BUSY pins can be used as ordinary I/O ports.

(1) Register setting

The operation stop mode is set by 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	CSIM 11	CSIM 10	FF68H	00H	R/W

CSIE1	CSIM 11	PM20	P20	PM21	P21	PM22	P22	Shift register 1 operation	Serial clock counter operation control	SI1/P20 pin function	SO1/P21 pin function	$\overline{SCK1}$/P22 pin function
0	×	Note 1 ×	Note 1 ×	Note 1 ×	Note 1 ×	Note 1 ×	Note 1 ×	Operation stop	Clear	P20 (CMOS I/O)	P21 (CMOS I/O)	P22 (CMOS I/O)
1	0	Note 2 1	Note 2 ×	0	0	1	×	Operation enable	Count operation	SI1 Note 2 (input)	SO1 (CMOS output)	$\overline{SCK1}$ (input)
	1					0	1					$\overline{SCK1}$ (CMOS output)

Notes 1. Can be used freely as a port pin.

2. Can be used as P20 (CMOS I/O) when used only for transmission (set bit 7 (RE) of the automatic data transmit/receive control register (ADTC) to 0).

Remark ×: don't care

PM××: Port mode register

P××: Port output latch

14.4.2 3-wire serial I/O mode operation

The 3-wire serial I/O mode is used for connection of peripheral ICs and display controllers that incorporate a clocked serial interface.

Communication is carried out using three lines: a serial clock ($\overline{SCK1}$), serial output (SO1), and serial input (SI1).

(1) Register setting

The 3-wire serial I/O mode is set by 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	CSIM 11	CSIM 10	FF68H	00H	R/W

CSIM 11	CSIM 10	Serial interface channel 1 clock selection
0	x	Clock externally input to $\overline{SCK1}$ pin[Note 1]
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 4 to 7 of timer clock select register 3 (TCL3)

ATE	Serial interface channel 1 operating mode selection
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

DIR	Start bit	SI1 pin function	SO1 pin function
0	MSB	SI1/P20 (input)	SO1 (CMOS output)
1	LSB		

CSIE1	CSIM 11	PM20	P20	PM21	P21	PM22	P22	Shift register 1 operation	Serial clock counter operation control	SI1/P20 pin function	SO1/P21 pin function	$\overline{SCK1}$/P22 pin function
0	×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Operation stop	Clear	P20 (CMOS I/O)	P21 (CMOS I/O)	P22 (CMOS I/O)
1	0	Note 3 1	Note 3 ×	0	0	1	×	Operation enable	Count operation	SI1[Note 3] (input)	SO1 (CMOS output)	$\overline{SCK1}$ (input)
	1					0	1					$\overline{SCK1}$ (CMOS output)

Notes 1. If 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 a port pin.

3. Can be used as P20 when used only for transmission (set bit 7 (RE) of ADTC to 0).

Remark ×: don't care
PM××: Port mode register
P××: Port output latch

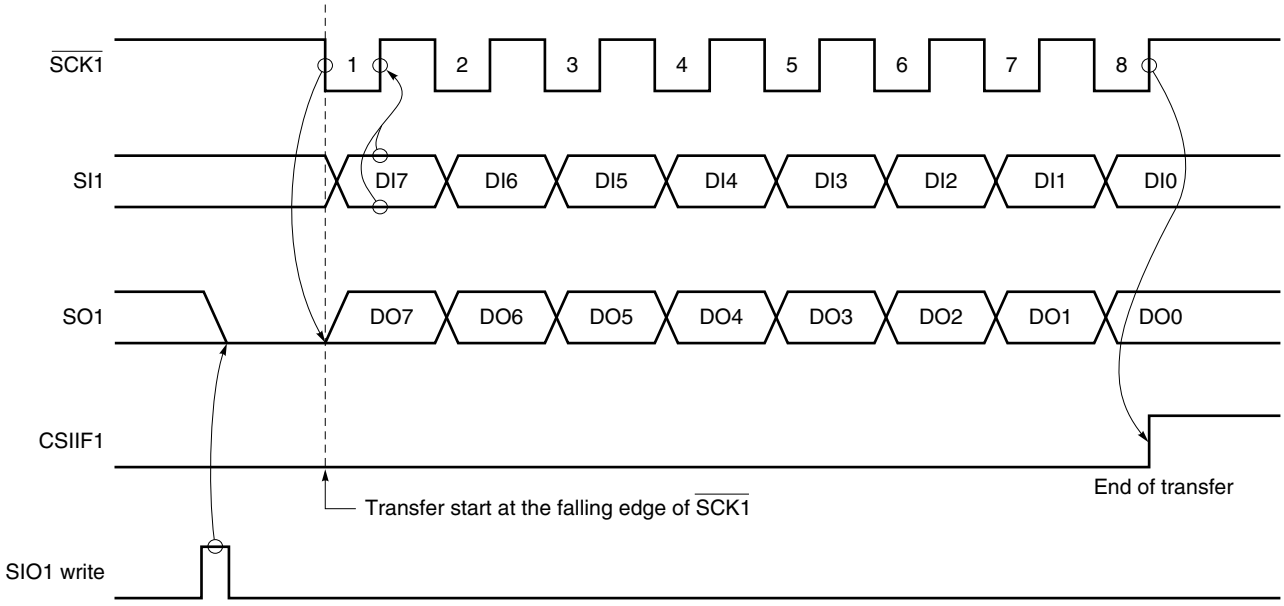
(2) Communication operation

The 3-wire serial I/O mode is used for data transmission/reception in 8-bit units. Data transmission/reception is carried out bit by bit in synchronization with the serial clock.

Shift operations of serial I/O shift register 1 (SIO1) are carried out at the falling edge of the serial clock ($\overline{SCK1}$). The transmit data is held in the SO1 latch and is output from the SO1 pin. The receive data input to the SI1 pin is latched into SIO1 at the rising edge of $\overline{SCK1}$.

Upon termination of 8-bit transfer, the SIO1 operation stops automatically and the interrupt request flag (CSIIF1) is set.

Figure 14-6. 3-Wire Serial I/O Mode Timing



Caution The SO1 pin becomes low level by SIO1 write.

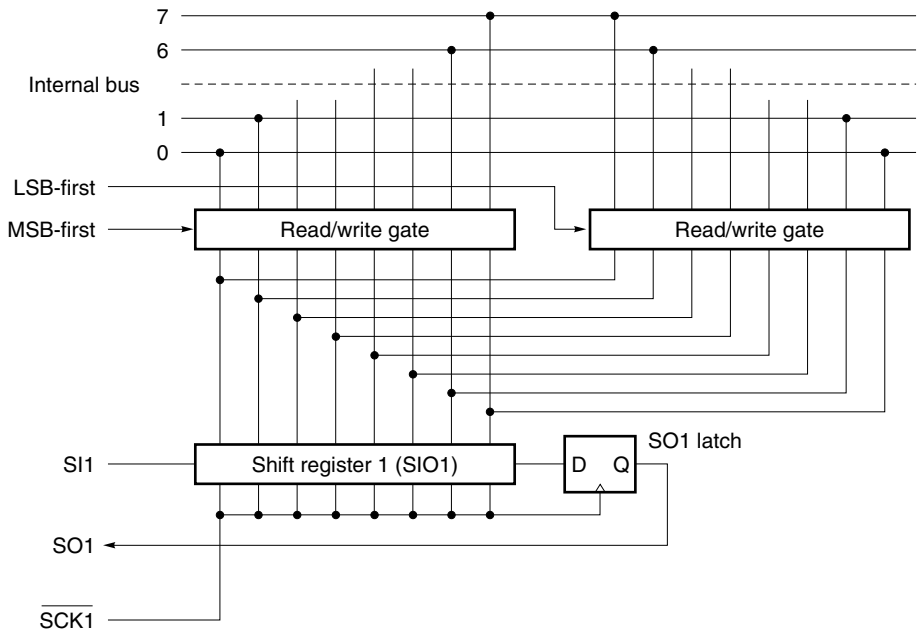
(3) MSB/LSB switching as the start bit

In the 3-wire serial I/O mode, transfer can be selected to start from the MSB or LSB.

Figure 14-7 shows the configuration of serial I/O shift register 1 (SIO1) and the internal bus. As shown in the figure, the MSB/LSB can be read/written in reverse form.

MSB/LSB switching as the start bit can be specified using bit 6 (DIR) of serial operating mode register 1 (CSIM1).

Figure 14-7. Circuit for Switching Transfer Bit Order



Start bit switching is realized by switching the bit order for data write to SIO1. The SIO1 shift order remains unchanged.

Thus, switching between MSB-first and LSB-first must be performed before writing data to the shift register.

(4) Transfer start

Serial transfer is started by setting transfer data to 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 at high level after 8-bit serial transfer.

Caution If CSIE1 is set to "1" after data write to SIO1, transfer does not start.

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (CSIIF1) is set.

14.4.3 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 up to 64-byte data without using software. Once transfer is started, the set number of bytes of the data prestored in the RAM can be transmitted, and the set number of bytes of data can be received and stored in the RAM.

Handshake signals (STB and BUSY) are supported by hardware to transmit/receive data continuously. An OSD (On Screen Display) LSI and peripheral LSI including an 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 by serial operating mode register 1 (CSIM1), the automatic data transmit/receive control register (ADTC), and the automatic data transmit/receive interval specification register (ADTI).

(a) Serial operating mode register 1 (CSIM1)

CSIM1 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ 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	CSIM 11	CSIM 10	FF68H	00H	R/W

CSIM 11	CSIM 10	Serial interface channel 1 clock selection
0	×	Clock externally input to $\overline{\text{SCK1}}$ pin[Note 1]
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 4 to 7 of timer clock select register 3 (TCL3)

ATE	Serial interface channel 1 operating mode selection
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

DIR	Start bit	SI1 pin function	SO1 pin function
0	MSB	SI1/P20 (input)	SO1 (CMOS output)
1	LSB		

CSIE1	CSIM 11	PM20	P20	PM21	P21	PM22	P22	Shift register 1 operation	Serial clock counter operation control	SI1/P20 pin function	SO1/P21 pin function	$\overline{\text{SCK1}}$/P22 pin function
0	×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Operation stop	Clear	P20 (CMOS I/O)	P21 (CMOS I/O)	P22 (CMOS I/O)
1	0	Note 3 1	Note 3 ×	0	0	1	×	Operation enable	Count operation	SI1[Note 3] (input)	SO1 (CMOS output)	$\overline{\text{SCK1}}$ (input)
	1					0	1					$\overline{\text{SCK1}}$ (CMOS output)

Notes 1. If 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 a port pin.

3. Can be used as P20 when used only for transmission (set bit 7 (RE) of ADTC to 0).

Remark ×: don't care

PM××: Port mode register

P××: Port output latch

(b) Automatic data transmit/receive control register (ADTC)

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

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

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
ADTC	RE	ARLD	ERCE	ERR	TRF	STRB	BUSY1	BUSY0	FF69H	00H	R/W[Note 1]

R/W

BUSY1	BUSY0	Busy input control
0	x	Not using busy input
1	0	Busy input enabled (active high)
1	1	Busy input enabled (active low)

R/W

STRB	Strobe output control
0	Strobe output disabled
1	Strobe output enabled

R

TRF	Status of automatic transmit/receive function[Note 2]
0	Detection of termination of automatic transmission/ reception (This bit is set to 0 upon suspension of automatic transmission/reception or when ARLD = 0.)
1	During automatic transmission/reception (This bit is set to 1 when data is written to SIO1.)

R

ERR	Error detection of automatic transmit/receive function
0	No error (This bit is set to 0 when data is written to SIO1.)
1	Error occurred

R/W

ERCE	Error check control of automatic transmit/ receive function
0	Error check disabled
1	Error check enabled (only when BUSY1 = 1)

R/W

ARLD	Operating mode selection of automatic transmit/ receive function
0	Single operating mode
1	Repetitive operating mode

R/W

RE	Receive control of automatic transmit/receive function
0	Receive disabled
1	Receive enabled

Notes 1. Bits 3 and 4 (TRF and ERR) are read-only bits.

2. The termination of automatic transmission/reception should be judged by using TRF, not CSIIF1 (interrupt request flag).

Caution When external clock input is selected with bit 1 (CSIM11) of serial operating mode register 1 (CSIM1) set to 0, set STRB and BUSY1 of ADTC to 0, 0 (handshake control cannot be executed when an external clock is input).

Remark ×: don't care

(c) Automatic data transmit/receive interval specification register (ADTI)

This register sets the data transfer interval of the automatic transmit/receive function.

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_X = 5.0 MHz operation)	
					Minimum[Note 2]	Maximum[Note 2]
0	0	0	0	0	36.8 μs + 0.5/f_{SCK}	40.0 μs + 1.5/f_{SCK}
0	0	0	0	1	62.4 μs + 0.5/f_{SCK}	65.6 μs + 1.5/f_{SCK}
0	0	0	1	0	88.0 μs + 0.5/f_{SCK}	91.2 μs + 1.5/f_{SCK}
0	0	0	1	1	113.6 μs + 0.5/f_{SCK}	116.8 μs + 1.5/f_{SCK}
0	0	1	0	0	139.2 μs + 0.5/f_{SCK}	142.4 μs + 1.5/f_{SCK}
0	0	1	0	1	164.8 μs + 0.5/f_{SCK}	168.0 μs + 1.5/f_{SCK}
0	0	1	1	0	190.4 μs + 0.5/f_{SCK}	193.6 μs + 1.5/f_{SCK}
0	0	1	1	1	216.0 μs + 0.5/f_{SCK}	219.2 μs + 1.5/f_{SCK}
0	1	0	0	0	241.6 μs + 0.5/f_{SCK}	244.8 μs + 1.5/f_{SCK}
0	1	0	0	1	267.2 μs + 0.5/f_{SCK}	270.4 μs + 1.5/f_{SCK}
0	1	0	1	0	292.8 μs + 0.5/f_{SCK}	296.0 μs + 1.5/f_{SCK}
0	1	0	1	1	318.4 μs + 0.5/f_{SCK}	321.6 μs + 1.5/f_{SCK}
0	1	1	0	0	344.0 μs + 0.5/f_{SCK}	347.2 μs + 1.5/f_{SCK}
0	1	1	0	1	369.6 μs + 0.5/f_{SCK}	372.8 μs + 1.5/f_{SCK}
0	1	1	1	0	395.2 μs + 0.5/f_{SCK}	398.4 μs + 1.5/f_{SCK}
0	1	1	1	1	420.8 μs + 0.5/f_{SCK}	424.0 μs + 1.5/f_{SCK}

Notes 1. The interval is dependent only on CPU processing.

2. The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if the minimum calculated by the following expression is smaller than $2/f_{SCK}$, the minimum interval time is $2/f_{SCK}$.

$$\text{Minimum} = (n + 1) \times \frac{2^7}{fx} + \frac{56}{fx} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^7}{fx} + \frac{72}{fx} + \frac{1.5}{f_{SCK}}$$

Cautions 1. ADTI should not be written to during operation of the automatic transmit/receive function.

2. Bits 5 and 6 must be set to 0.

3. When ADTI is used to control the interval time of data transfer by automatic transmit/receive function, busy control (refer to 14.4.3 (4) (a) Busy control option) is invalid.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{SCK}: Serial clock frequency

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data transfer interval specification (f_X = 5.0 MHz operation)	
					Minimum[Note]	Maximum[Note]
1	0	0	0	0	446.4 μs + 0.5/f_{SCK}	449.6 μs + 1.5/f_{SCK}
1	0	0	0	1	472.0 μs + 0.5/f_{SCK}	475.2 μs + 1.5/f_{SCK}
1	0	0	1	0	497.6 μs + 0.5/f_{SCK}	500.8 μs + 1.5/f_{SCK}
1	0	0	1	1	523.2 μs + 0.5/f_{SCK}	526.4 μs + 1.5/f_{SCK}
1	0	1	0	0	548.8 μs + 0.5/f_{SCK}	552.0 μs + 1.5/f_{SCK}
1	0	1	0	1	574.4 μs + 0.5/f_{SCK}	577.6 μs + 1.5/f_{SCK}
1	0	1	1	0	600.0 μs + 0.5/f_{SCK}	603.2 μs + 1.5/f_{SCK}
1	0	1	1	1	625.6 μs + 0.5/f_{SCK}	628.8 μs + 1.5/f_{SCK}
1	1	0	0	0	651.2 μs + 0.5/f_{SCK}	654.4 μs + 1.5/f_{SCK}
1	1	0	0	1	676.8 μs + 0.5/f_{SCK}	680.0 μs + 1.5/f_{SCK}
1	1	0	1	0	702.4 μs + 0.5/f_{SCK}	705.6 μs + 1.5/f_{SCK}
1	1	0	1	1	728.0 μs + 0.5/f_{SCK}	731.2 μs + 1.5/f_{SCK}
1	1	1	0	0	753.6 μs + 0.5/f_{SCK}	756.8 μs + 1.5/f_{SCK}
1	1	1	0	1	779.2 μs + 0.5/f_{SCK}	782.4 μs + 1.5/f_{SCK}
1	1	1	1	0	804.8 μs + 0.5/f_{SCK}	808.0 μs + 1.5/f_{SCK}
1	1	1	1	1	830.4 μs + 0.5/f_{SCK}	833.6 μs + 1.5/f_{SCK}

Note The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if the minimum calculated by the following expression is smaller than 2/f_{SCK}, the minimum interval time is 2/f_{SCK}.

$$\text{Minimum} = (n + 1) \times \frac{2^7}{f_X} + \frac{56}{f_X} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^7}{f_X} + \frac{72}{f_X} + \frac{1.5}{f_{SCK}}$$

Cautions 1. ADTI should not be written to during operation of the automatic transmit/receive function.

2. Bits 5 and 6 must be set to 0.

3. When ADTI is used to control the interval time of data transfer by automatic transmit/receive function, busy control (refer to 14.4.3 (4) (a) Busy control option) is invalid.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{SCK}: Serial clock frequency

(2) Automatic transmit/receive data setting

(a) Transmit data setting

[1] Write transmit data from the least significant address of buffer RAM, FAC0H (up to FAFFH). The transmit data should be in the order from higher address to lower address.
[2] Set the automatic data transmit/receive address pointer (ADTP) to the value obtained by subtracting 1 from the number of transmit data bytes.

(b) Automatic transmit/receive mode setting

[1] Set bit 7 (CSIE1) and bit 5 (ATE) of serial operating mode register 1 (CSIM1) to 1.
[2] Set RE of the automatic data transmit/receive control register (ADTC) to 1.
[3] Set the data transmit/receive transfer interval in the automatic data transmit/receive interval specification register (ADTI).
[4] Write any value to serial I/O shift register 1 (SIO1) (transfer start trigger).

Caution Writing any value to SIO1 orders the start of an automatic transmit/receive operation; 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 by 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 by 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 the data of address FAC0H is output (end of automatic transmission/reception).
- When automatic transmission/reception is terminated, bit 3 (TRF) of ADTC is cleared to 0.

(3) Communication operation

(a) Basic transmission/reception mode

This transmission/reception mode is the same as the 3-wire serial I/O mode in which the specified number of data are transmitted/received in 8-bit units.

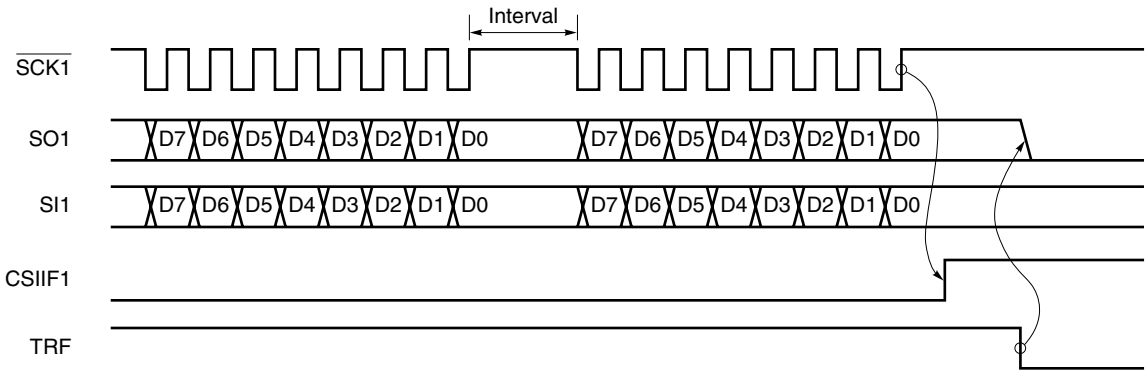
Serial transfer is started when any data is written to serial I/O shift register 1 (SIO1) while bit 7 (CSIE1) of serial operating mode register 1 (CSIM1) is set to 1.

Upon completion of transmission of the last byte, the interrupt request flag (CSIIF1) is set. However, the termination of automatic transmission/reception should be judged by bit 3 (TRF) of the automatic data transmit/receive control register (ADTC), not CSIIF1.

If busy control and strobe control are not executed, the P23/STB and P24/BUSY pins can be used as normal I/O ports.

Figure 14-8 shows the basic transmission/reception mode operation timing, and Figure 14-9 shows the operation flowchart. The operation of the buffer RAM to transmit/receive 6-byte data is shown in Figure 14-10.

Figure 14-8. Basic Transmission/Reception Mode Operation Timing



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 until the next transmission/reception.

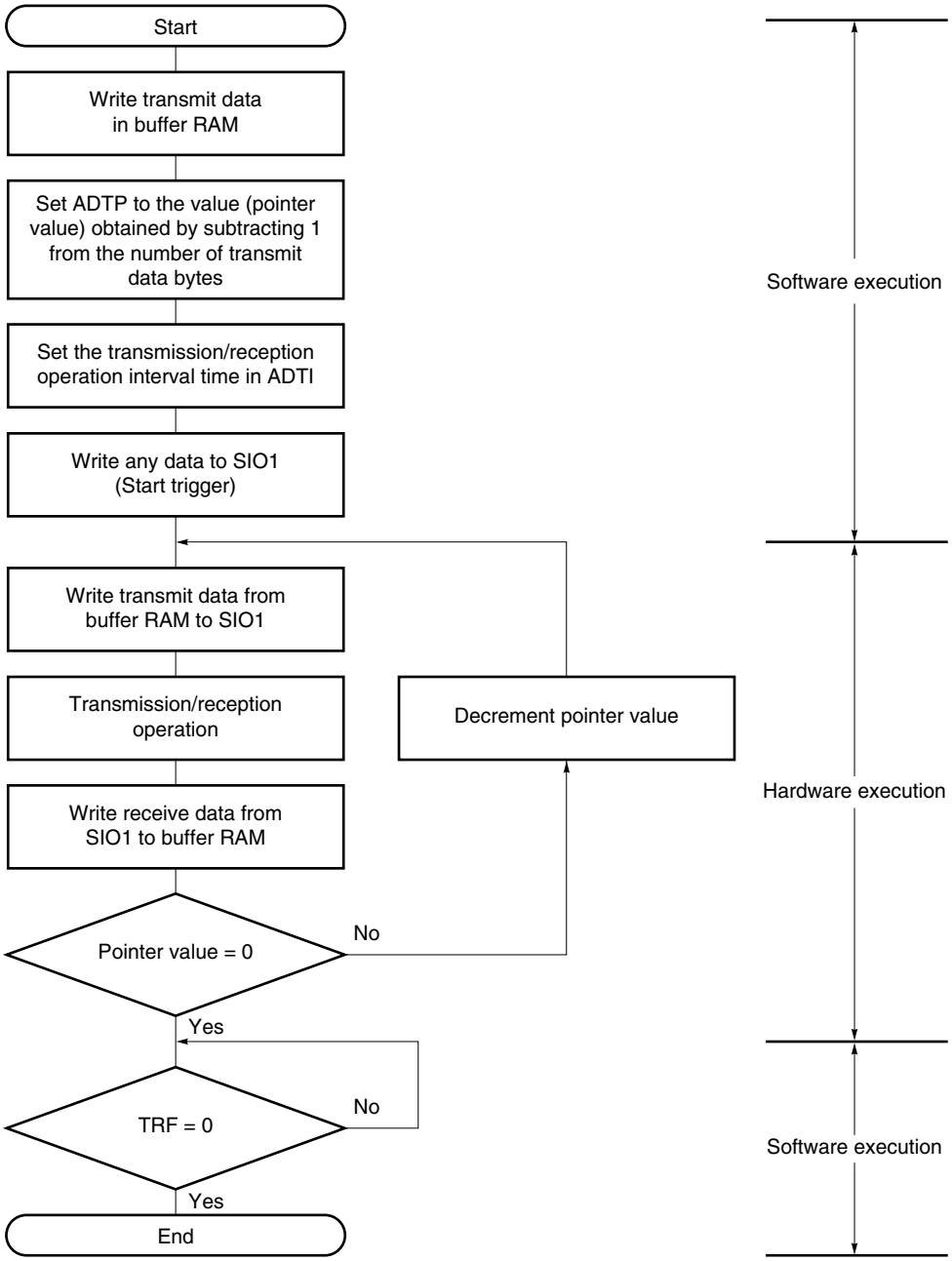
As the buffer RAM write/read is performed at the same time as CPU processing, the maximum interval is dependent upon CPU processing and the value of the automatic data transmit/receive interval specification register (ADTI) (see (5) Automatic transmit/receive interval).

2. When TRF is cleared, the SO1 pin becomes low level.

Remark CSIIF1: Interrupt request flag

TRF: Bit 3 of the automatic data transmit/receive control register (ADTC)

Figure 14-9. Basic Transmission/Reception Mode Flowchart



ADTP: Automatic data transmit/receive address pointer
ADTI: Automatic data transmit/receive interval specification register
SIO1: Serial I/O shift register 1
TRF: Bit 3 of the automatic data transmit/receive control register (ADTC)

In 6-byte transmission/reception (ARLD = 0, RE = 1) in basic transmission/reception mode, buffer RAM operates as follows.

(i) Before transmission/reception (refer to **Figure 14-10 (a)**)

After arbitrary data has been written to 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, receive data 1 (R1) is transferred from SIO1 to the buffer RAM, and the automatic data transmit/receive address pointer (ADTP) is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1.

(ii) 4th byte transmission/reception point (refer to **Figure 14-10 (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, receive data 4 (R4) is transferred from SIO1 to the buffer RAM, and ADTP is decremented.

(iii) Completion of transmission/reception (refer to **Figure 14-10 (c)**)

When transmission of the sixth byte is completed, receive data 6 (R6) is transferred from SIO1 to the buffer RAM, and the interrupt request flag (CSIIF1) is set (INTCSI1 generation).

Figure 14-10. Buffer RAM Operation in 6-Byte Transmission/Reception (in Basic Transmission/Reception Mode) (1/2)

(a) Before transmission/reception

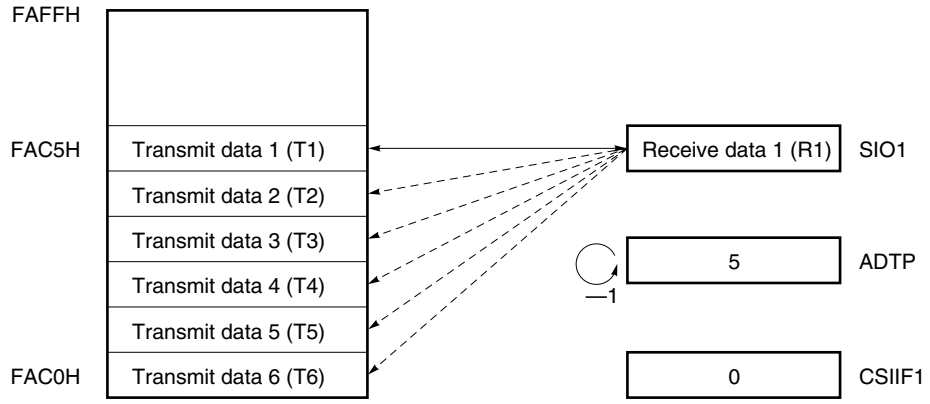
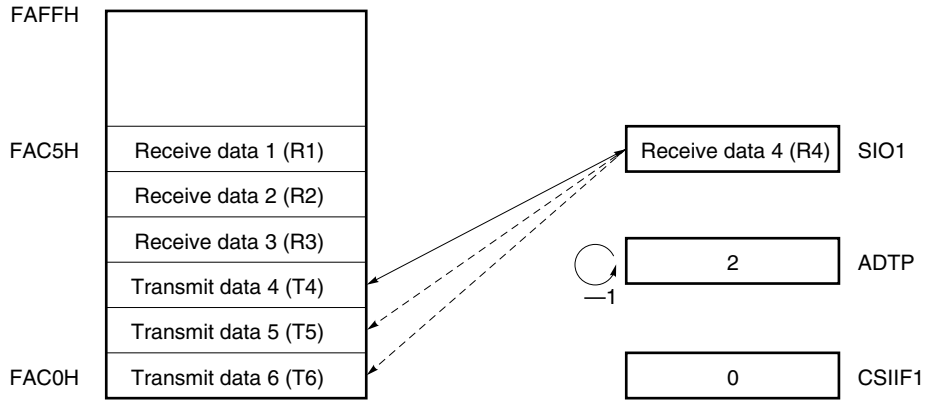
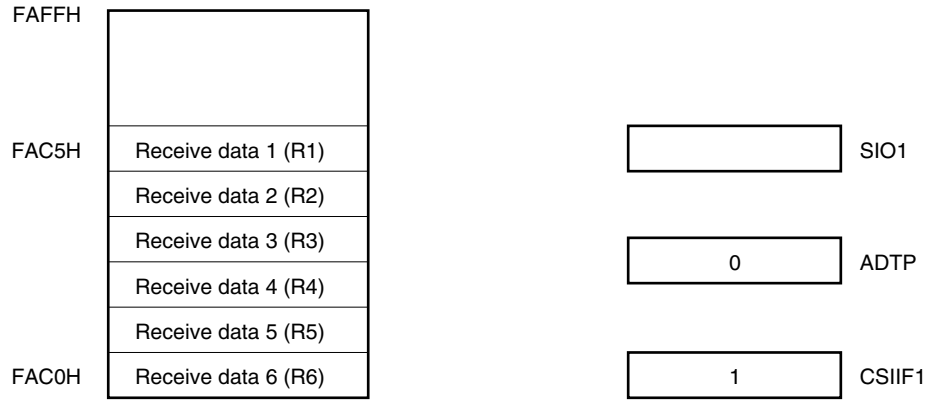


Figure 14-10. Buffer RAM Operation in 6-Byte Transmission/Reception (in Basic Transmission/Reception Mode) (2/2)

(b) 4th byte transmission/reception point



(c) Completion of transmission/reception



(b) Basic transmission mode

In this mode, the specified number of 8-bit unit data are transmitted.

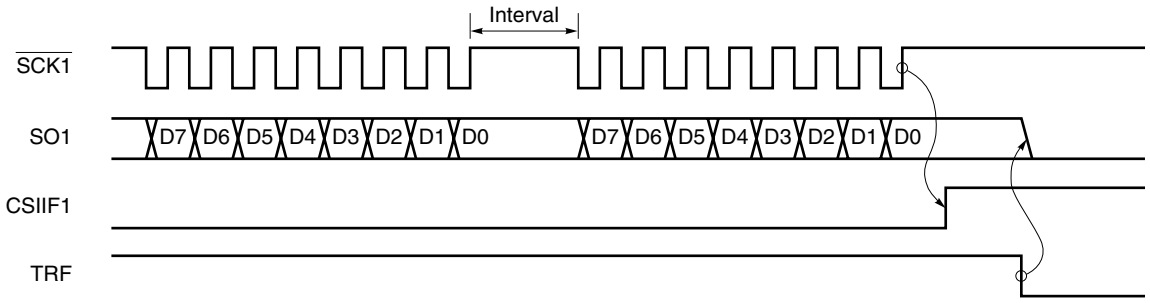
Serial transfer is started when any data is written to serial I/O shift register 1 (SIO1) while bit 7 (CSIE1) of serial operating mode register 1 (CSIM1) is set to 1.

Upon completion of transmission of the last byte, the interrupt request flag (CSIIF1) is set. However, the termination of automatic transmission/reception should be judged by bit 3 (TRF) of the automatic data transmit/receive control register (ADTC), not CSIIF1.

If a receive operation, busy control, and strobe control are not executed, the P20/SI1, P23/STB, and P24/BUSY pins can be used as normal I/O ports.

Figure 14-11 shows the basic transmission mode operation timing, and Figure 14-12 shows the operation flowchart. The operation of the buffer RAM to transmit 6-byte data in transmission mode is shown in Figure 14-13.

Figure 14-11. Basic Transmission Mode Operation Timing



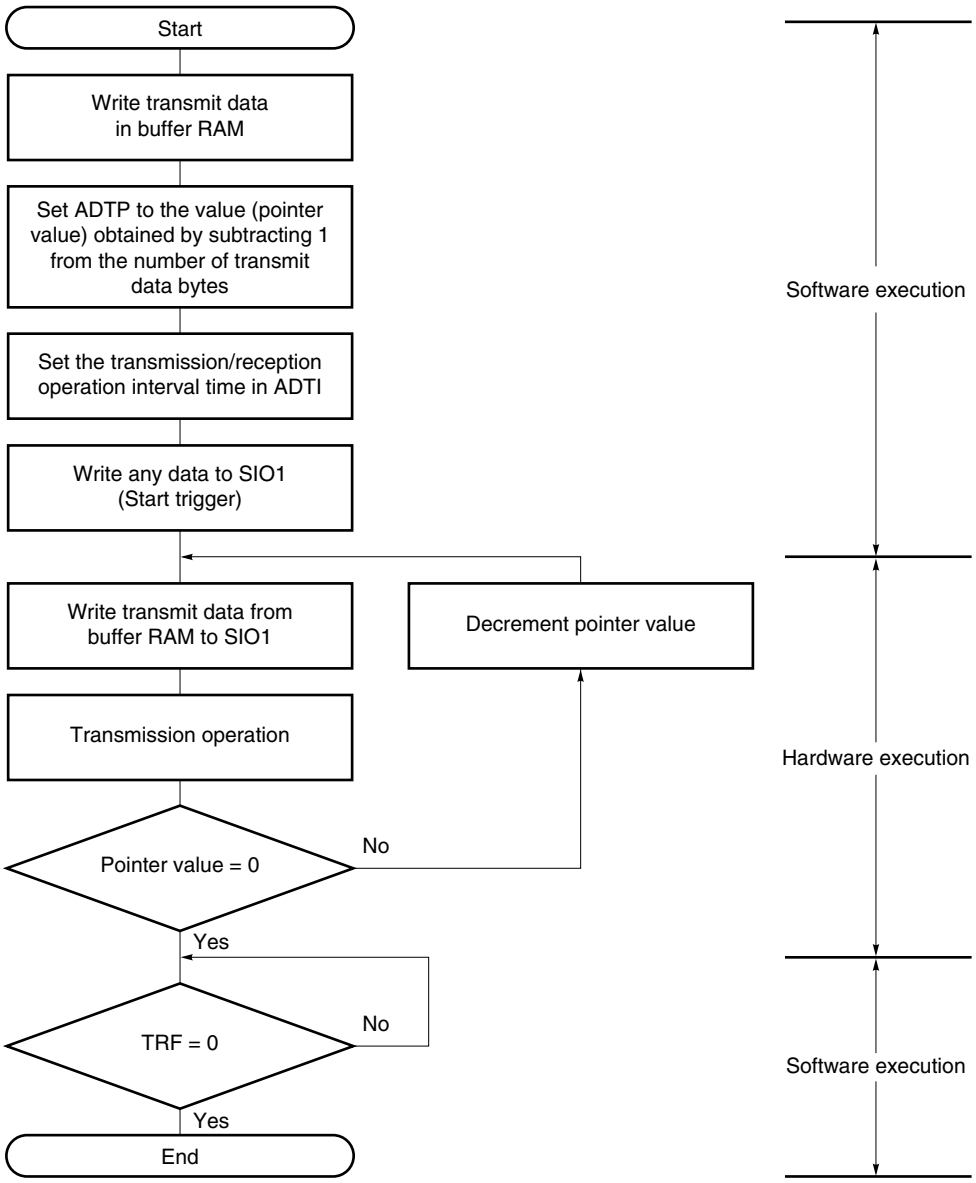
Cautions 1. Because, in the basic transmission mode, the automatic transmit/receive function reads data from the buffer RAM after 1-byte transmission, an interval is inserted until the next transmission. As the buffer RAM read is performed at the same time as CPU processing, the maximum interval is dependent upon CPU processing and the value of the automatic data transmit/receive interval specification register (ADTI) (see (5) Automatic transmit/receive interval).

2. When TRF is cleared, the SO1 pin becomes low level.

Remark CSIIF1: Interrupt request flag

TRF: Bit 3 of the automatic data transmit/receive control register (ADTC)

Figure 14-12. Basic Transmission Mode Flowchart



ADTP: Automatic data transmit/receive address pointer
ADTI: Automatic data transmit/receive interval specification register
SIO1: Serial I/O shift register 1
TRF: Bit 3 of the automatic data transmit/receive control register (ADTC)

In 6-byte transmission (ARLD = 0, RE = 0) in basic transmission mode, buffer RAM operates as follows.

(i) Before transmission (refer to **Figure 14-13 (a)**)

After arbitrary data has been written to 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 14-13 (b)**)

Transmission of the third byte is completed, and transmit data 4 (T4) is transferred from the buffer RAM to SIO1. When transmission of the fourth byte is completed, ADTP is decremented.

(iii) Completion of transmission (refer to **Figure 14-13 (c)**)

When transmission of the sixth byte is completed, the interrupt request flag (CSIIF1) is set (INTCSI1 generation).

Figure 14-13. Buffer RAM Operation in 6-Byte Transmission (in Basic Transmission Mode) (1/2)

(a) Before transmission

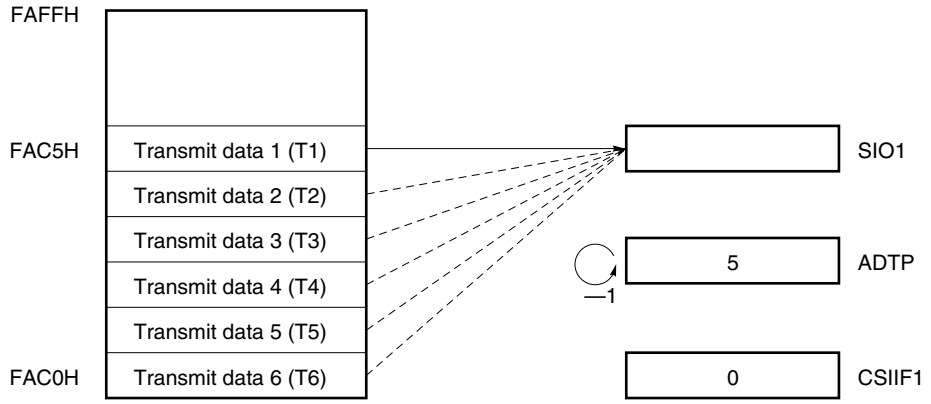
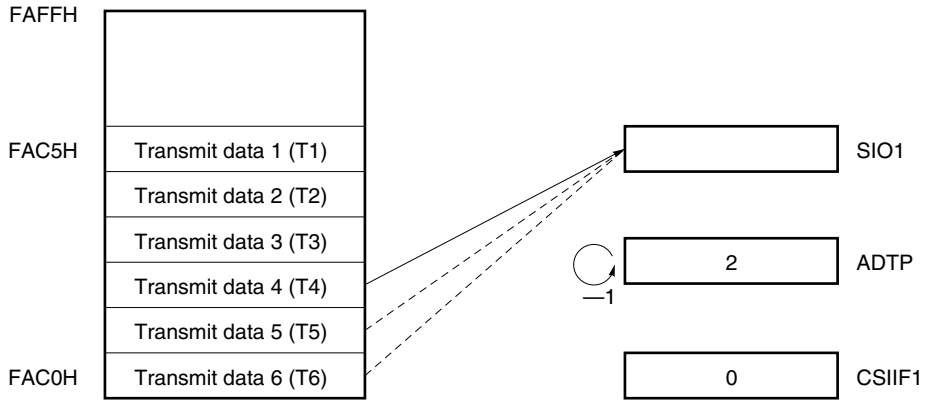
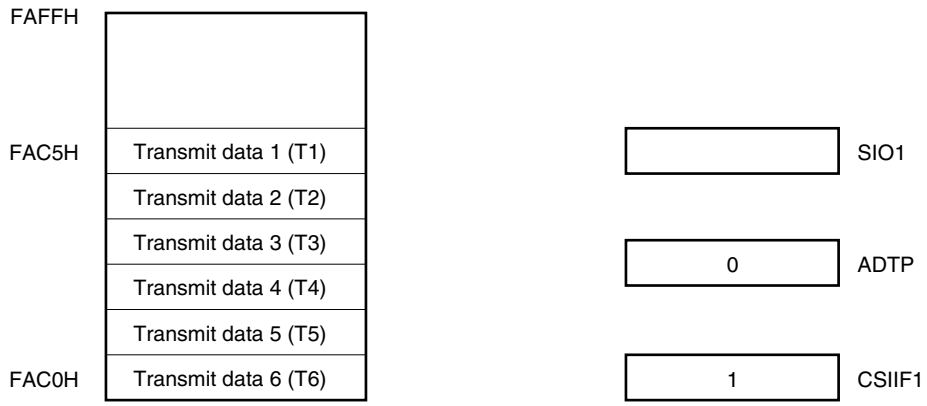


Figure 14-13. Buffer RAM Operation in 6-Byte Transmission (in Basic Transmission Mode) (2/2)

(b) 4th byte transmission point



(c) Completion of transmission



(c) Repeat transmission mode

In this mode, data stored in the buffer RAM is transmitted repeatedly.

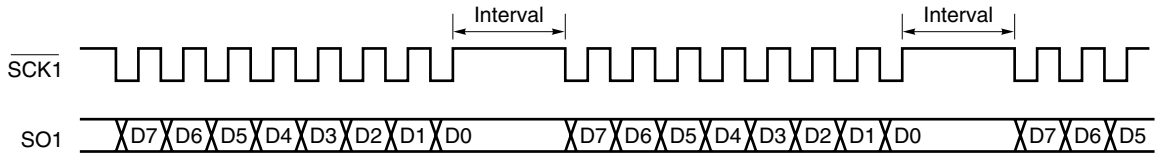
Serial transfer is started by writing any data to serial I/O shift register 1 (SIO1) while bit 7 (CSIE1) of serial operating mode register 1 (CSIM1) is set to 1.

Unlike the basic transmission mode, after the last byte (data in address FAC0H) has been transmitted, the interrupt request flag (CSIIF1) is not set, the value at the time 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 normal I/O ports.

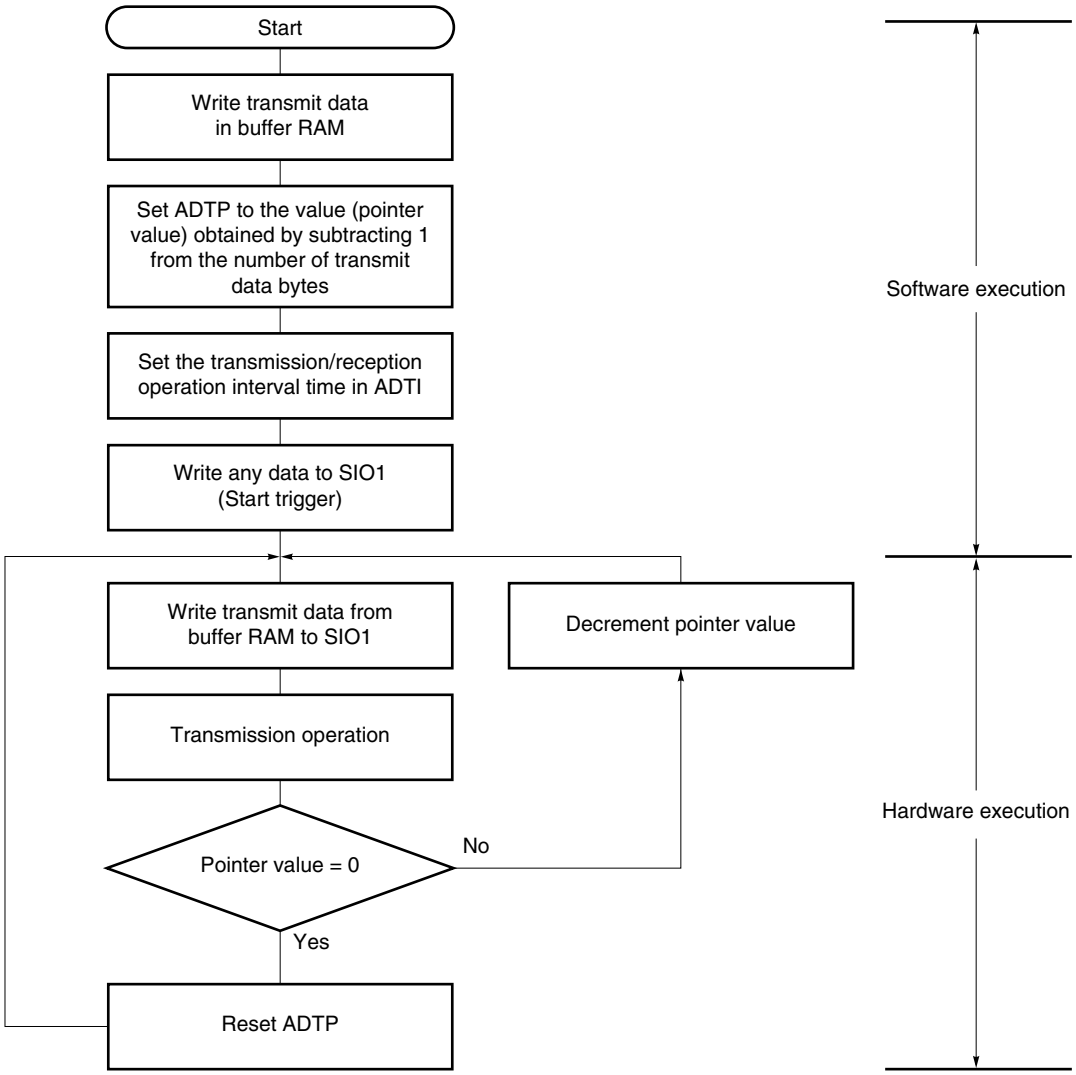
The repeat transmission mode operation timing is shown in Figure 14-14, and the operation flowchart in Figure 14-15. The operation of the buffer RAM to transmit 6-byte data in repeat transmission mode is shown in Figure 14-16.

Figure 14-14. Repeat Transmission Mode Operation Timing



Caution Since, in the repeat transmission mode, a read is performed on the buffer RAM after the transmission of one byte, the interval is included in the period up to the next transmission. As the buffer RAM read is performed at the same time as CPU processing, the maximum interval is dependent upon the CPU operation and the value of the automatic data transmit/receive interval specification register (ADTI) (see (5) Automatic transmit/receive interval).

Figure 14-15. Repeat Transmission Mode Flowchart



ADTP: Automatic data transmit/receive address pointer
ADTI: Automatic data transmit/receive interval specification register
SIO1: Serial I/O shift register 1

When data of 6 bytes are transmitted in repeat transmission mode (ARLD = 1, RE = 0), the buffer RAM operates as follows.

(i) Before transmission (refer to **Figure 14-16 (a)**)

After arbitrary data has been written to 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 14-16 (b)**)

When transmission of the sixth byte is completed, the interrupt request flag (CSIIF1) is not set. The first pointer value is set again to ADTP.

(iii) 7th byte transmission point (refer to **Figure 14-16 (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 14-16. Buffer RAM Operation in 6-Byte Transmission (in Repeat Transmission Mode) (1/2)

(a) Before transmission

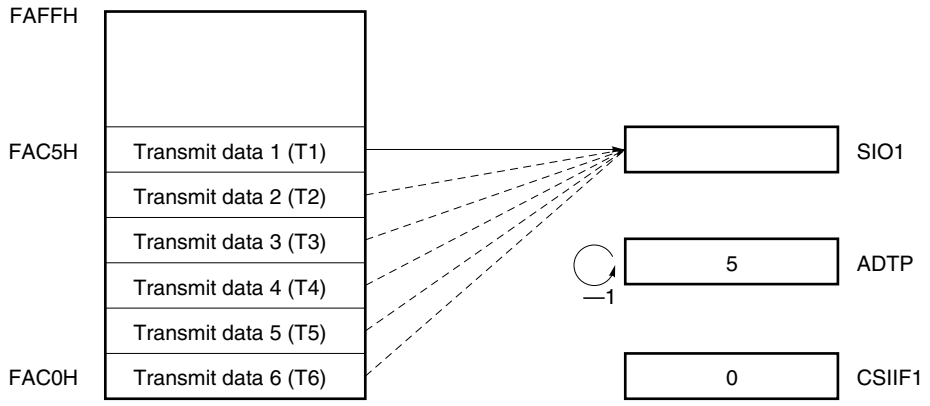
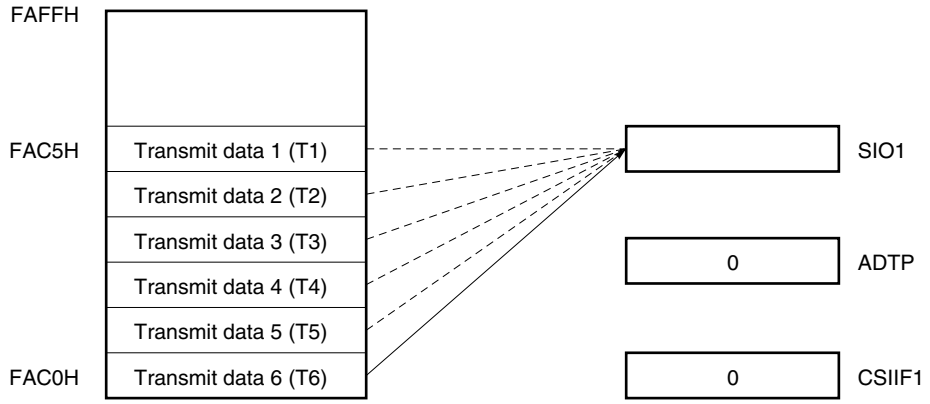
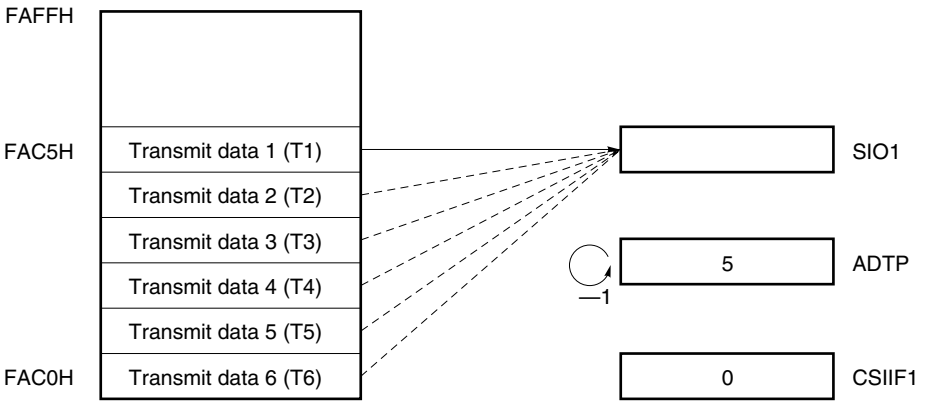


Figure 14-16. Buffer RAM Operation in 6-Byte Transmission (in Repeat Transmission Mode) (2/2)

(b) Upon completion of transmission of 6 bytes



(c) 7th byte transmission point



(d) Automatic transmission/reception suspension and restart

Automatic transmission/reception can be temporarily suspended by resetting bit 7 (CSIE1) of serial operating mode register 1 (CSIM1) to 0.

If 8-bit data transfer is in progress, the transmission/reception is not suspended if bit 7 (CSIE1) is reset to 0. It is suspended upon completion of 8-bit data transfer.

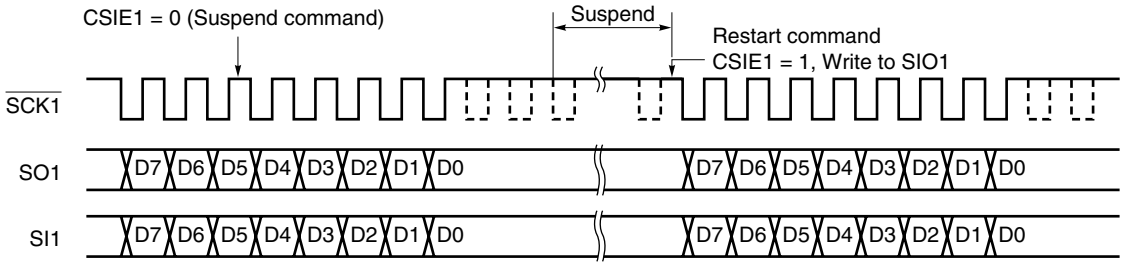
When suspended, bit 3 (TRF) of the automatic data transmit/receive control register (ADTC) is set to 0 after transfer of the 8th bit, and all the port pins used alternately as serial interface pins (P20/SI1, P21/SO1, P22/SCK1, P23/STB, and P24/BUSY) are set to the port mode.

To resume automatic transmission/reception, set CSIE1 to 1 and write any value to serial I/O shift register 1 (SIO1). This enables transmission of the remaining data.

Cautions 1. If the HALT instruction is executed during automatic transmission/reception, transfer is suspended and the HALT mode is set even during 8-bit data transfer. When the HALT mode is cleared, automatic transmission/reception is restarted at the suspended point.

2. When suspending automatic transmission/reception, do not change the operating mode to 3-wire serial I/O mode while TRF = 1.

Figure 14-17. Automatic Transmission/Reception Suspension and Restart



CSIE1: Bit 7 of serial operating mode register 1 (CSIM1)

(4) Synchronization control

Busy control and strobe control are used to synchronize transmission/reception data between the master device and slave device.

By using these functions, a bit slippage in data being transmitted/received can be detected.

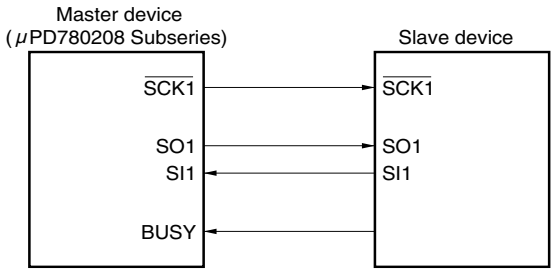
(a) Busy control option

Busy control is used to allow a slave device to output a busy signal to the master device, so that the master device puts serial transmission/reception into a wait state while the busy signal is active. To use the busy control option, the following conditions must be satisfied.

- Set bit 5 (ATE) of serial operating mode register 1 (CSIM1) to 1.
- Set bit 1 (BUSY1) of the automatic data transmit/receive control register (ADTC) to 1.

Figure 14-18 shows the system configuration of the master device and a slave device when the busy control option is used.

Figure 14-18. System Configuration with Busy Control Option



The master device inputs the busy signal output by the slave device to the BUSY/P24 pin. It samples the input busy signal in synchronization with the fall of the serial clock. Even if the busy signal becomes active while 8-bit data is being transmitted or received, transmission/reception is not put into a wait state. If the busy signal is active at the rising edge of the serial clock two clocks after transmission or reception of 8-bit data has been completed, the busy signal becomes valid. After that, transmission or reception is put into a wait state while the busy signal is active.

The active level of the busy signal is specified by bit 0 (BUSY0) of ADTC, as follows.

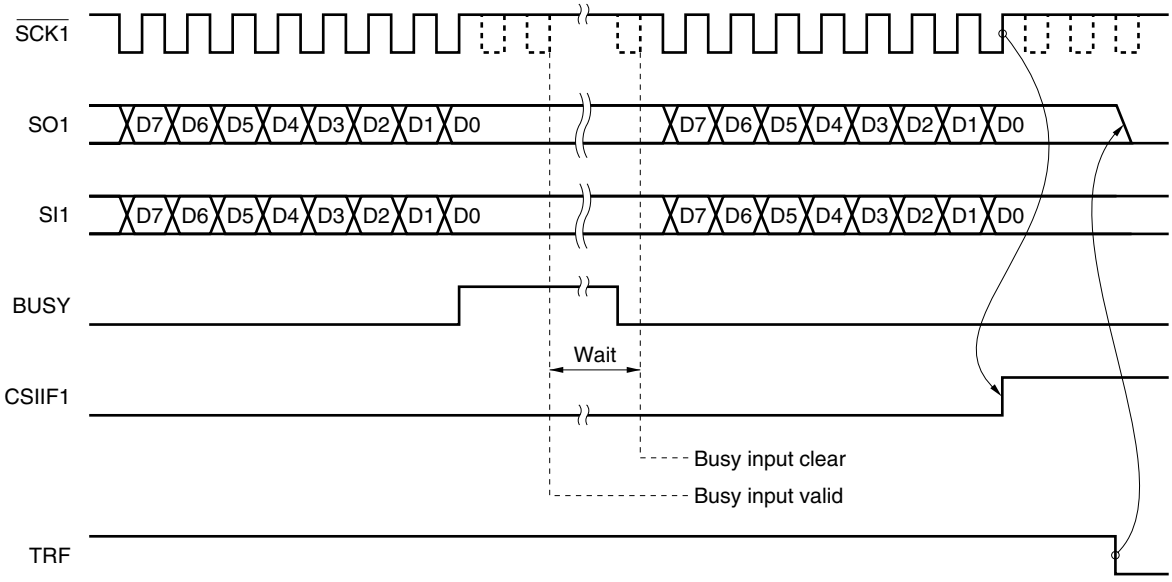
BUSY0 = 0: Active high
BUSY0 = 1: Active low

When using the busy control option, select the internal clock as the serial clock. Busy control cannot be executed with an external clock.

Figure 14-19 shows the operation timing when using the busy control option.

Caution Busy control cannot be executed when the interval time is controlled by using the automatic data transmit/receive interval specification register (ADTI). If an attempt is made to execute both control operations at the same time, busy control is invalid.

Figure 14-19. Operation Timing When Using Busy Control Option (BUSY0 = 0)

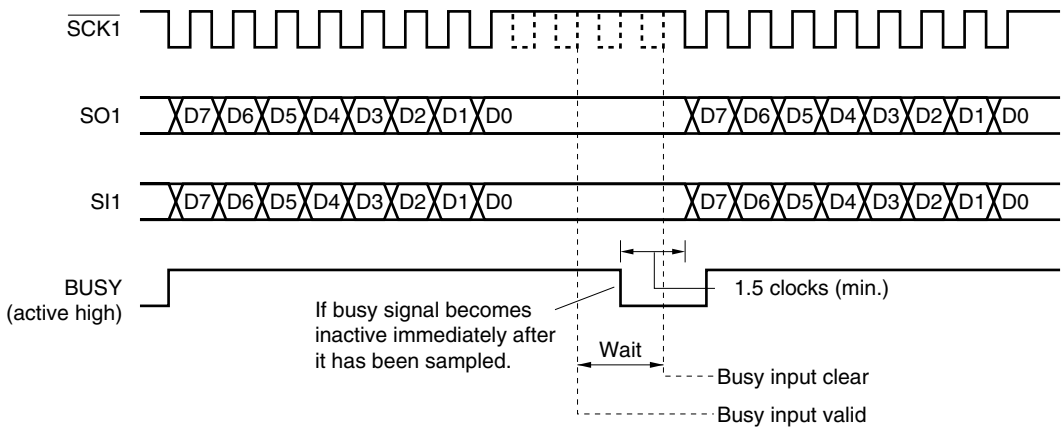


Caution When TRF is cleared, the SO1 pin becomes low level.

Remark CSIIF1: Interrupt request flag
TRF: Bit 3 of the automatic data transmit/receive control register (ADTC)

When the busy signal becomes inactive, the wait is cleared. If the sampled busy signal is inactive, transmission/reception of the next 8-bit data is started at the falling edge of the next serial clock. Note that, because the busy signal is asynchronous to the serial clock, it takes the master device up to 1 clock to sample the busy signal even if the slave device has made the busy signal inactive. In addition, it takes 0.5 clock until data transfer is started after the signal has been sampled.
To clear the wait, therefore, it is necessary for the slave device to keep the busy signal inactive for at least 1.5 clocks.
Figure 14-20 shows the timing of the busy signal and wait clearance. In Figure 14-20, the busy signal becomes active as soon as transmission/reception has started.

Figure 14-20. Busy Signal and Clearing Wait (BUSY0 = 0)



(b) Busy & strobe control option

Strobe control is used to synchronize data transmission/reception between the master device and a slave device. The master device outputs a strobe signal from the STB/P23 pin on completion of transmission/reception of 8-bit data. This strobe signal informs the slave device of the data transmission/reception completion timing of the master device. Therefore, synchronization can be established even if bit slippage occurs due to noise carried on the serial clock, keeping bit slippage from affecting transmission of the next byte.

To use the strobe control option, the following conditions must be satisfied.

- Set bit 5 (ATE) of serial operating mode register 1 (CSIM1) to 1.
- Set bit 2 (STRB) of the automatic data transmit/receive control register (ADTC) to 1.

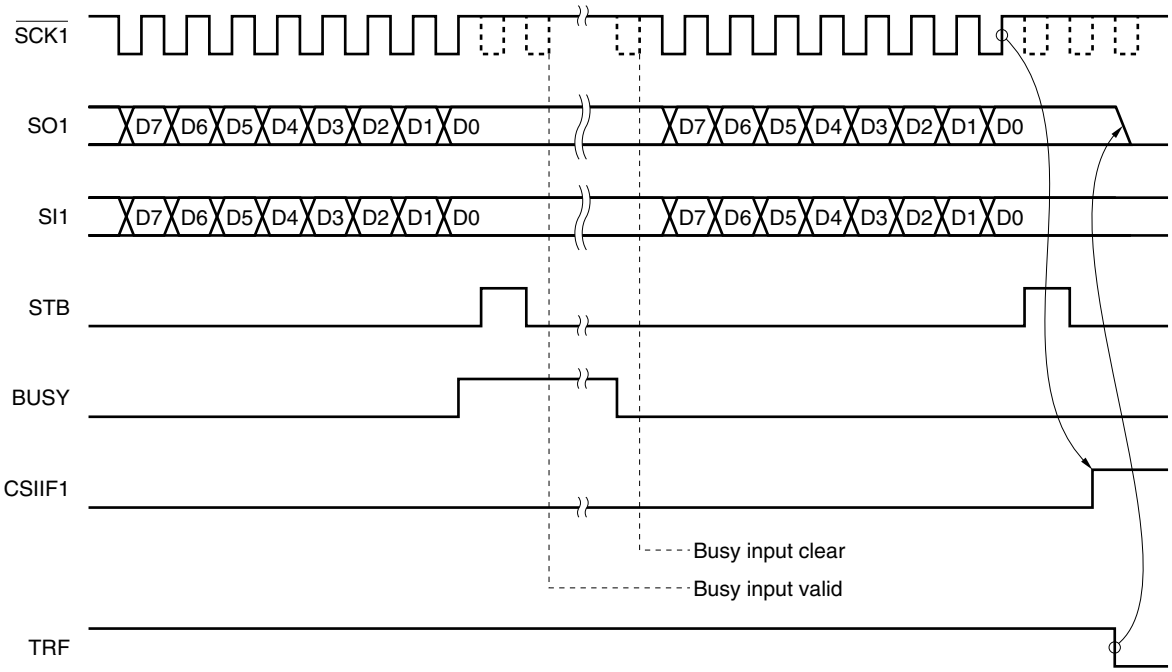
Usually, the busy control and strobe control options are simultaneously used for handshaking. In this case, the strobe signal is output from the STB/P23 pin and the BUSY/P24 pin is sampled. While a busy signal is being input to the pin, transmission/reception can be put into a wait state.

If strobe control is not executed, the P23/STB pin can be used as a normal I/O port pin.

Figure 14-21 shows the operation timing when using the busy & strobe control option.

When the strobe control option is used, the interrupt request flag (CSIIF1) that is set on completion of transmission/reception is set after the strobe signal has been output.

Figure 14-21. Operation Timing When Using Busy & Strobe Control Option (BUSY0 = 0)



Caution When TRF is cleared, the SO1 pin becomes low level.

Remark CSIIF1: Interrupt request flag
TRF: Bit 3 of the automatic data transmit/receive control register (ADTC)

(c) Bit slippage detection function with busy signal

During automatic transmission/reception, bit slippage may take place in the serial clock of the slave device due to the noise carried on the serial clock signal output by the master device. Unless the strobe control option is used at this time, the bit slippage affects transmission of the next byte. In such a case, the master device can detect the bit slippage by using the busy control option and checking the busy signal during transmission.

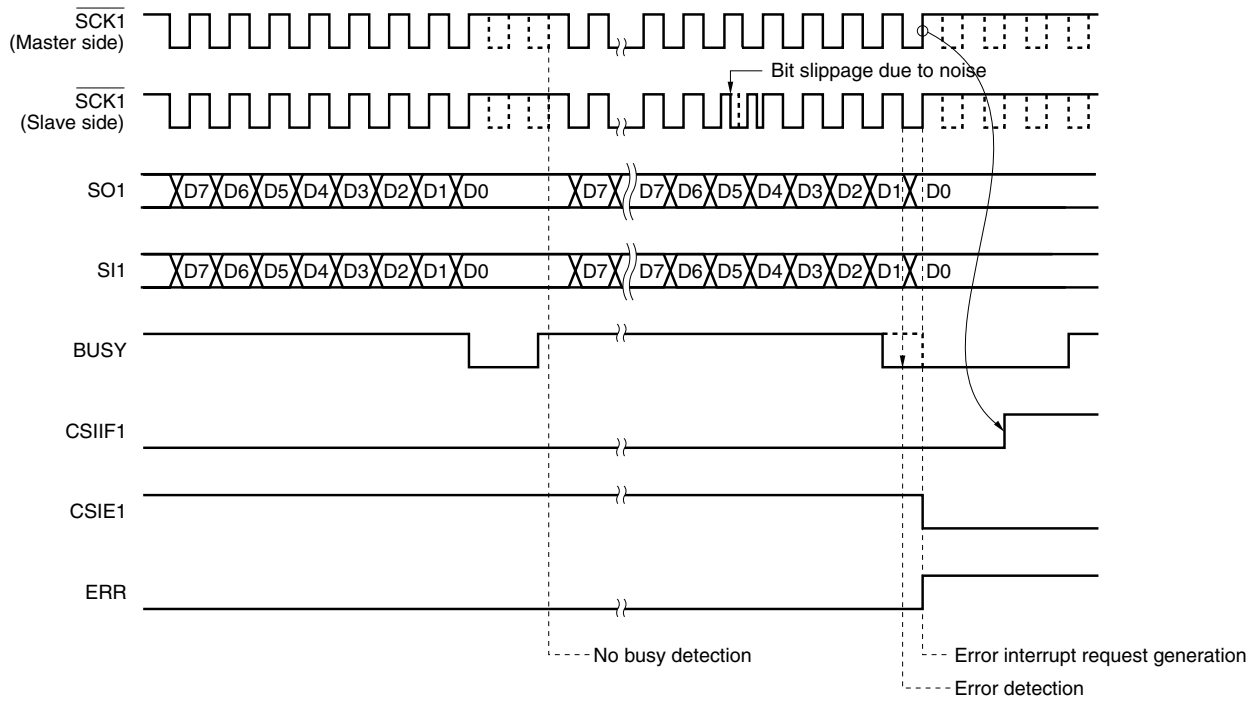
The bit slippage is detected by using the busy signal as follows.

The slave outputs a busy signal after the 8th rise of the serial clock during data transmission/reception (at this time, make the busy signal inactive within two clocks to stop the master device putting transmission/reception into a wait state).

The master samples the busy signal in synchronization with the fall of the serial clock. If bit slippage does not occur, the busy signal is found to be inactive after it has been sampled eight times. If the busy signal is found to be active when it has been sampled, it is assumed that bit slippage has occurred, and error processing is performed (by setting bit 4 (ERR) of the automatic data transmit/receive control register (ADTC) to 1).

Figure 14-22 shows the operation timing of the bit slippage detection function using the busy signal.

Figure 14-22. Operation Timing of Bit Slippage Detection Function Using Busy Signal (BUSY0 = 1)



CSIIF1: 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

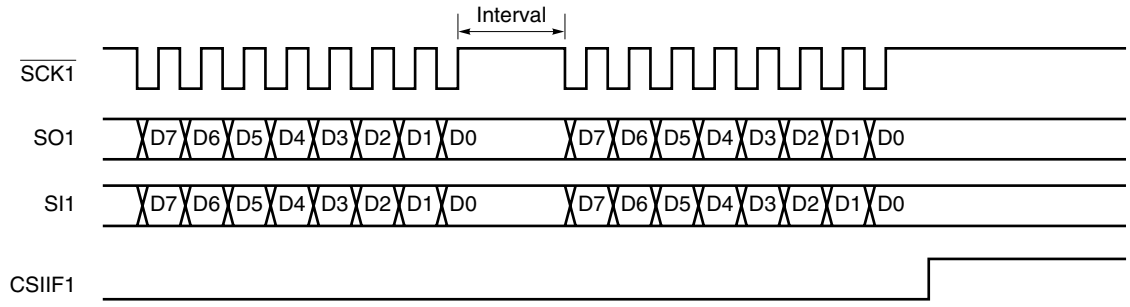
When the automatic transmit/receive function is used, one byte is transmitted/received and then the buffer RAM is read/written; therefore an interval is inserted before the next transmission/reception.

When the automatic transmit/receive function is performed using an internal clock, since the read/write operations from/to the buffer RAM are done in parallel with CPU processing, the interval depends on the CPU processing at the timing of the serial clock's eighth rising-edge and the value set in the automatic data transmit/receive interval specification register (ADTI). Whether or not the interval depends on ADTI can be selected by setting bit 7 (ADTI7) of ADTI.

When ADTI7 is set to 0, the interval depends only on the CPU processing. When ADTI7 is set to 1, the interval is the value determined by the contents of ADTI or other value determined by CPU processing, whichever is greater.

When the automatic transmit/receive function is performed using an external clock, the clock must be selected so that the interval is longer than the value shown in (b).

Figure 14-23. Automatic Transmit/Receive Interval



CSIIF1: Interrupt request flag

(a) When automatic transmit/receive function is performed using an internal clock

The internal clock operation is selected when bit 1 (CSIM11) of serial operating mode register 1 (CSIM1) is set to 1.

In this case, the interval is determined as follows by CPU processing.

When bit 7 (ADTI7) of the automatic data transmit/receive interval specification register (ADTI) is set to 0, the interval is determined by CPU processing. When ADTI7 is set to 1, the interval is determined by the contents of ADTI or by CPU processing, whichever is greater.

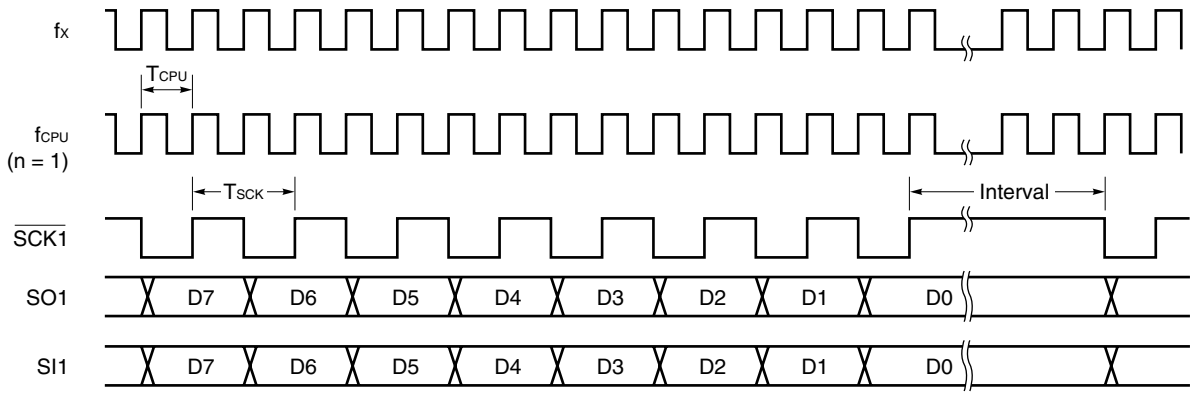
For the interval determined by ADTI, see **Figure 14-5 Format of Automatic Data Transmit/Receive Interval Specification Register**.

Table 14-3. Interval Determined by CPU Processing (with Internal Clock Operation)

CPU Processing	Interval
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))
MAX. (a, b): a or b, whichever is greater

Figure 14-24. Operation Timing When Automatic Transmit/Receive Function Is Operating with Internal Clock



fx: Main system clock oscillation frequency
f_{CPU}: CPU clock (set by bits 0 to 2 (PCC0 to PCC2) of the processor clock control register (PCC))
T_{CPU}: $1/f_{CPU}$
T_{SCK}: $1/f_{SCK}$
f_{SCK}: Serial clock frequency

(b) When automatic transmit/receive function is performed using an external clock

The external clock operation is selected when bit 1 (CSIM11) of serial operating mode register 1 (CSIM1) is cleared to 0.

When the automatic transmit/receive function is performed using an external clock, the clock must be selected so that the interval is longer than the values shown below.

Table 14-4. Interval Determined by CPU Processing (with External Clock Operation)

CPU Processing	Interval
When using multiplication instruction	$13T_{CPU}$ or more
When using division instruction	$20T_{CPU}$ or more
External access 1-wait mode	$9T_{CPU}$ or more
Other than above	$7T_{CPU}$ or more

T_{CPU}: $1/f_{CPU}$

f_{CPU}: CPU clock (set by bits 0 to 2 (PCC0 to PCC2) of the processor clock control register (PCC))

CHAPTER 15 VFD CONTROLLER/DRIVER

15.1 VFD Controller/Driver Functions

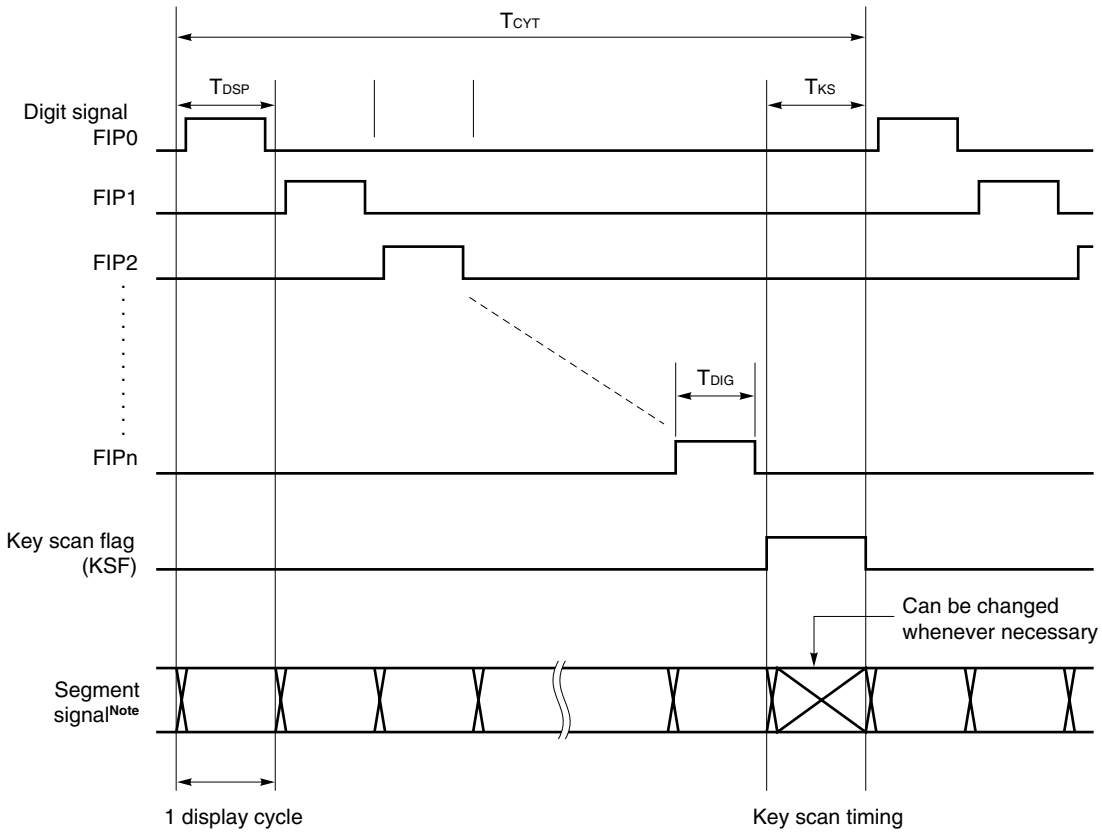
The functions of the VFD controller/driver incorporated in the μPD780208 Subseries are as follows.

(1) Automatically outputs the segment signals (DMA operation) and digit signals by automatically reading data displayed.
(2) Controls 9- to 40-segment and 2- to 16-digit VFDs (vacuum fluorescent display) using display mode registers 0, 1, and 2 (DSPM0 to DSPM2).
(3) Digit signal output timing can be specified freely by selecting display mode 2 using display mode register 0 (DSPM0).
(4) Pins not used for VFD display can be used as output and I/O ports (but FIP0 to FIP12 are display output-only pins).
(5) Luminance can be adjusted in 8 levels using display mode register 1 (DSPM1).
(6) Incorporates hardware for key scan application.
    - Generates interrupt signal (INTKS) indicating key scan timing.
    - Outputs key scan signals from segment output pins by setting key scan data to port 8 through port 12.
    - Detects timing at which key scan data are output by the key scan flag (KSF).
(7) Incorporates a high-withstanding-voltage output buffer that can directly drive the VFD.
(8) The display output pins can be connected to on-chip pull-down resistors by mask option.

Cautions 1. This cannot be used with the subsystem clock. To stop the main oscillation, stop the display in advance by setting bits 4 to 7 (DIGS0 to DIGS3) of display mode register 1 (DSPM1) to 0000.

2. Set ports 8 through 12 to 0 and output latches to 0 before doing the following:
- **Using the VFD controller/driver**
- **Stopping display**

Figure 15-1. VFD Controller Operation Timing in Display Mode 1 (DSPM05 = 0)



DSPM05: Bit 5 of display mode register 0 (DSPM0)

n: Displayed digits – 1 (Digits 2 to 16 can be selected using display mode register 1 (DSPM1))

T_{DSP}: 1 display cycle ($1024/f_x$ (204.8 μs: @ 5.0 MHz operation) or $2048/f_x$ (409.6 μs: @ 5.0 MHz operation))

T_{KS}: Key scan timing ($T_{KS} = T_{DSP}$)

T_{CYT}: Display cycle ($T_{CYT} = T_{DSP}$ x (Displayed digits + 1))

T_{DIG}: Pulse width of digit signal (Can be selected from 8 types using DSPM1)

Note The user can select the cut width of the segment signals by setting bits 1 to 3 (DIMS1 to DIMS3) of DSPM1. Therefore, actual output waveforms may be different from the above illustration and have the cut widths shown in Figure 15-6.

Remark If DSPM05 is set to 1, digit signals are output according to the values set in the display RAM.

There are 53 display output pins. Of these, 40 pins, FIP13 to FIP52, have alternate port functions. These pins are used as port pins when display stop is set using bits 4 to 7 (DIGS0 to DIGS3) of display mode register 1 (DSPM1).

Even when display is enabled, display output pins not used for outputting digit signals and segment signals can be used as port pins.

Table 15-1. Relationship Between Display Output Pins and Port Pins

Display Pin Name	Alternate Port Name	I/O
FIP13 to FIP20	P80 to P87	For output port
FIP21 to FIP28	P90 to P97	For output port
FIP29 to FIP36	P100 to P107	I/O port
FIP37 to FIP44	P110 to P117	I/O port
FIP45 to FIP52	P120 to P127	I/O port

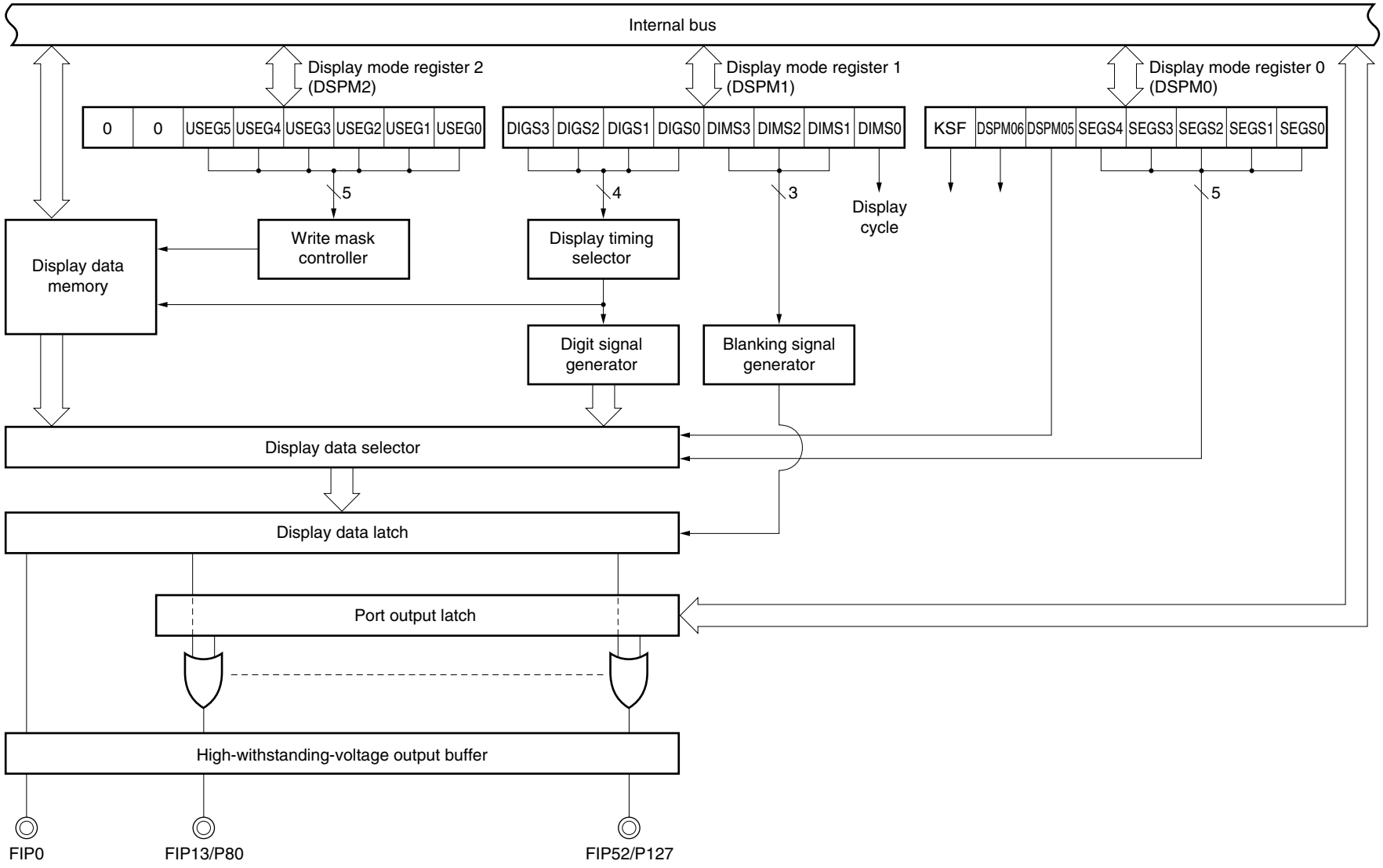
15.2 VFD Controller/Driver Configuration

The VFD controller/driver consists of the following hardware.

Table 15-2. VFD Controller/Driver Configuration

Item	Configuration
Display output	53 pins (segments: 9 to 40, digits: 2 to 16)
Control registers	Display mode register 0 (DSPM0) Display mode register 1 (DSPM1) Display mode register 2 (DSPM2)

Figure 15-2. VFD Controller/Driver Block Diagram



15.3 VFD Controller/Driver Control Registers

15.3.1 Control registers

There are three registers for controlling the VFD controller/driver.

- Display mode register 0 (DSPM0)
- Display mode register 1 (DSPM1)
- Display mode register 2 (DSPM2)

(1) Display mode register 0 (DSPM0) (see Figure 15-3)

This register sets the following and displays the display timing/key scan state.

- Display mode
- Display segment number/display output total number
- Mode for subsystem clock noise eliminator

DSPM0 is set with an 8-bit memory manipulation instruction. However, only bit 7 (KSF) can be read with a 1-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets DSPM0 to 00H.

(2) Display mode register 1 (DSPM1) (see Figure 15-4)

This register sets the following.

- Display digit number/display pattern number
- Cut width of the VFD output signal
- Display cycle (T_{DSP})

When bit 0 (DIMS0) is set to 1 and the display cycle to 2048/f_x (409.6 μs: @ 5.0 MHz operation), light leakage is reduced. As the display cycle approaches the commercial power supply frequency when the display digits are increased, the display will flicker. In this case, select 1024/f_x (204.8 μs: @ 5.0 MHz operation). If light leaks, adjust the cut width of the digit signal using bits 1 to 3 (DIMS1 to DIMS3).

DSPM1 is set with an 8-bit memory manipulation instruction.

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

(3) Display mode register 2 (DSPM2) (see Figure 15-5)

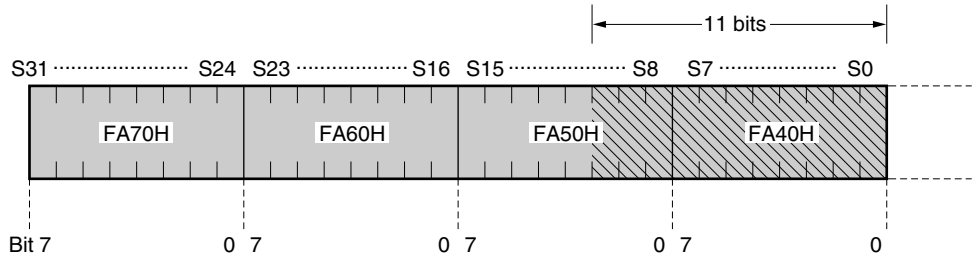
DSPM2 is the register that holds the number of mask bits in the display data storage area when display mode 2 (DSPM05 = 1) is selected by display mode register 0 (DSPM0). By using this register to mask the part of the display data that does not need to be rewritten, the software workload is reduced.

Mask bits are assigned from S0 (= the least significant bit of the lowest address in the display output area defined by bits 0 to 4 of DSPM0).

DSPM2 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets DSPM2 to 00H.

The following illustration shows the status of the display data memory when the number of segments is 32 and the number of mask bits is 11.



: The shaded part shows the area in which display data is rewritable during display

: The slashed part shows the area in which display data is not rewritable during display (display data are fixed)

Caution The number of mask bits specified must be below the total number of display outputs defined by display mode register 0 (DSPM0).

Figure 15-3. Format of Display Mode Register 0 (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
DSPM0	KSF	DSPM06	DSPM05	SEGS4	SEGS3	SEGS2	SEGS1	SEGS0	F F A 0 H	0 0 H	R/W

R/W	SEGS4	SEGS3	SEGS2	SEGS1	SEGS0	Display segment (display mode 1)	Display output total (display mode 2)
	0	0	0	0	0	9	9
	0	0	0	0	1	10	10
	0	0	0	1	0	11	11
	0	0	0	1	1	12	12
	0	0	1	0	0	13	13
	0	0	1	0	1	14	14
	0	0	1	1	0	15	15
	0	0	1	1	1	16	16
	0	1	0	0	0	17	17
	0	1	0	0	1	18	18
	0	1	0	1	0	19	19
	0	1	0	1	1	20	20
	0	1	1	0	0	21	21
	0	1	1	0	1	22	22
	0	1	1	1	0	23	23
	0	1	1	1	1	24	24
	1	0	0	0	0	25	25
	1	0	0	0	1	26	26
	1	0	0	1	0	27	27
	1	0	0	1	1	28	28
	1	0	1	0	0	29	29
	1	0	1	0	1	30	30
	1	0	1	1	0	31	31
	1	0	1	1	1	32	32
	1	1	0	0	0	33	33
	1	1	0	0	1	34	34
	1	1	0	1	0	35	35
	1	1	0	1	1	36	36
	1	1	1	0	0	37	37
	1	1	1	0	1	38Note	38
	1	1	1	1	0	39Note	39
	1	1	1	1	1	40Note	40

Note When the total number of digits and segments together exceeds 53, the digits have priority.

Figure 15-3. Format of Display Mode Register 0 (2/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
DSPM0	KSF	DSPM06	DSPM05	SEGS4	SEGS3	SEGS2	SEGS1	SEGS0	F F A 0 H	0 0 H	R/W[Note 1]

R/W	DSPM05	Display mode setting
	0	Display mode 1 (Segment/character type)
	1	Display mode 2 (Type in which a segment spans over two or more grids.)

R/W	DSPM06	Subsystem clock noise eliminator mode setting[Note 2]
	0	2.5 MHz < $f_X \leq$ 5.0 MHz
	1	1.25 MHz $\leq f_X \leq$ 2.5 MHz[Note 3]

R	KSF	Timing state
	0	Display timing
	1	Key scan timing

Notes 1. Bit 7 (KSF) is a read-only bit.

2. Set the values according to the main system clock oscillation frequency (f_X) used. The noise eliminator is enabled during VFD display operations.

3. If f_X selected is between 1.25 MHz and 2.5 MHz, set DSPM06 to 1 prior to VFD display.

Caution When a main system clock frequency below 1.25 MHz is selected and the VFD controller/driver is used, make sure to use the main system clock for watch timer counting by setting TCL24 to 0.

Figure 15-4. Format of Display Mode Register 1

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

DIMS0	Display mode cycle setting
0	1024/fx is 1 display cycle. (1 display cycle = 204.8 μs: when operated at 5.0 MHz)
1	2048/fx is 1 display cycle. (1 display cycle = 409.6 μs: when operated at 5.0 MHz)

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

DIGS3	DIGS2	DIGS1	DIGS0	Display digit (display mode 1) DSPM05 = 0	Display pattern (display mode 2) DSPM05 = 1
0	0	0	0	Display stopped (static display)[Note]	Display stopped (static display)[Note]
0	0	0	1	2 digits	2 patterns
0	0	1	0	3 digits	3 patterns
0	0	1	1	4 digits	4 patterns
0	1	0	0	5 digits	5 patterns
0	1	0	1	6 digits	6 patterns
0	1	1	0	7 digits	7 patterns
0	1	1	1	8 digits	8 patterns
1	0	0	0	9 digits	9 patterns
1	0	0	1	10 digits	10 patterns
1	0	1	0	11 digits	11 patterns
1	0	1	1	12 digits	12 patterns
1	1	0	0	13 digits	13 patterns
1	1	0	1	14 digits	14 patterns
1	1	1	0	15 digits	15 patterns
1	1	1	1	16 digits	16 patterns

Note Static display is possible when display stop is selected, by manipulating the port output latch.

Remark fx: Main system clock oscillation frequency
DSPM05: Bit 5 of display mode register 0 (DSPM0)

Figure 15-5. Format of Display Mode Register 2 (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
DSPM2	0	0	USEG5	USEG4	USEG3	USEG2	USEG1	USEG0	F F A 2 H	0 0 H	R/W

USEG5	USEG4	USEG3	USEG2	USEG1	USEG0	Number of mask bits to be written
0	0	0	0	0	0	None
0	0	0	0	0	1	1
0	0	0	0	1	0	2
0	0	0	0	1	1	3
0	0	0	1	0	0	4
0	0	0	1	0	1	5
0	0	0	1	1	0	6
0	0	0	1	1	1	7
0	0	1	0	0	0	8
0	0	1	0	0	1	9
0	0	1	0	1	0	10
0	0	1	0	1	1	11
0	0	1	1	0	0	12
0	0	1	1	0	1	13
0	0	1	1	1	0	14
0	0	1	1	1	1	15
0	1	0	0	0	0	16
0	1	0	0	0	1	17
0	1	0	0	1	0	18
0	1	0	0	1	1	19
0	1	0	1	0	0	20
0	1	0	1	0	1	21
0	1	0	1	1	0	22
0	1	0	1	1	1	23
0	1	1	0	0	0	24
0	1	1	0	0	1	25
0	1	1	0	1	0	26
0	1	1	0	1	1	27
0	1	1	1	0	0	28
0	1	1	1	0	1	29
0	1	1	1	1	0	30
0	1	1	1	1	1	31

Figure 15-5. Format of Display Mode Register 2 (2/2)

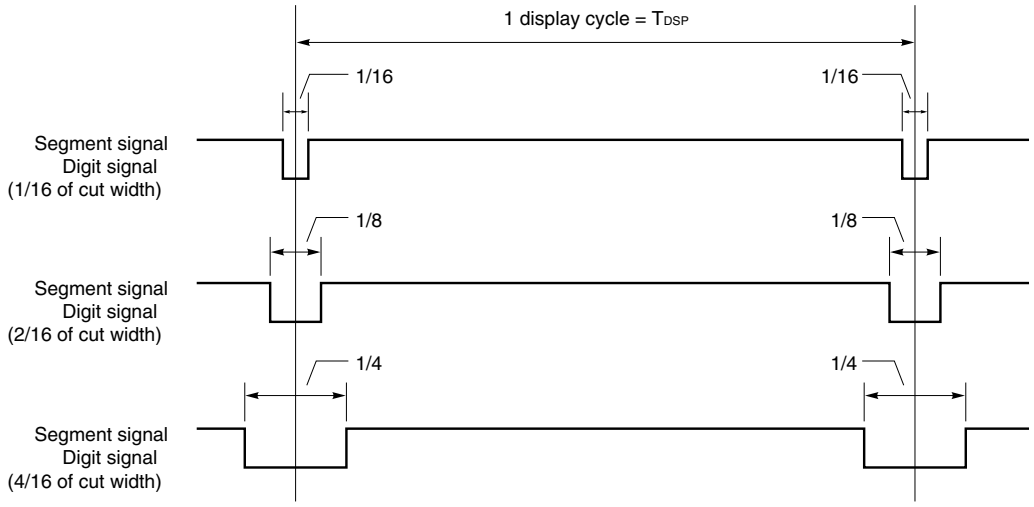
Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
DSPM2	0	0	USEG5	USEG4	USEG3	USEG2	USEG1	USEG0	F F A 2 H	0 0 H	R/W

USEG5	USEG4	USEG3	USEG2	USEG1	USEG0	Number of mask bits to be written
1	0	0	0	0	0	32
1	0	0	0	0	1	33
1	0	0	0	1	0	34
1	0	0	0	1	1	35
1	0	0	1	0	0	36
1	0	0	1	0	1	37
1	0	0	1	1	0	38
1	0	0	1	1	1	39
Other than the above						Setting prohibited

15.3.2 One-display period and cut width

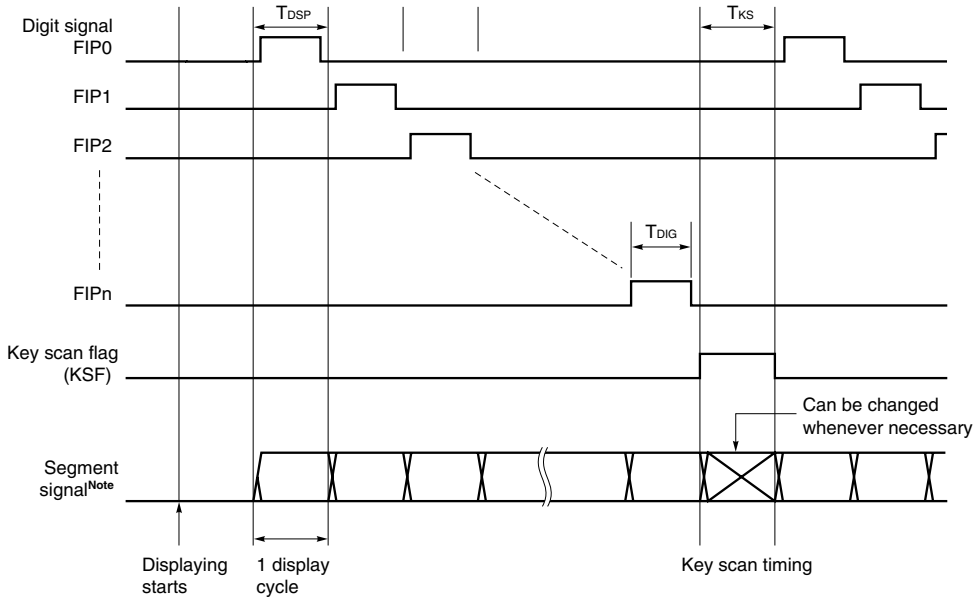
The digit signal is equally cut at the beginning and end of the display period by the cut width set by bits 1 to 3 (DIMS1 to DIMS3) of display mode register 1 (DSPM1).

Figure 15-6. Cut Width of Segment/Digit Signal



0 is output for the first one-display cycle when display is started from the display stop status.

Figure 15-7. VFD Controller Display Start Timing



n: Displayed digits – 1 (Digits 2 to 16 can be selected using display mode register 1 (DSPM1))

T_{DSP}: 1 display cycle (1024/f_x (204.8 μs: @ 5.0 MHz operation) or 2048/f_x (409.6 μs: @ 5.0 MHz operation))

T_{KS}: Key scan timing (T_{KS} = T_{DSP})

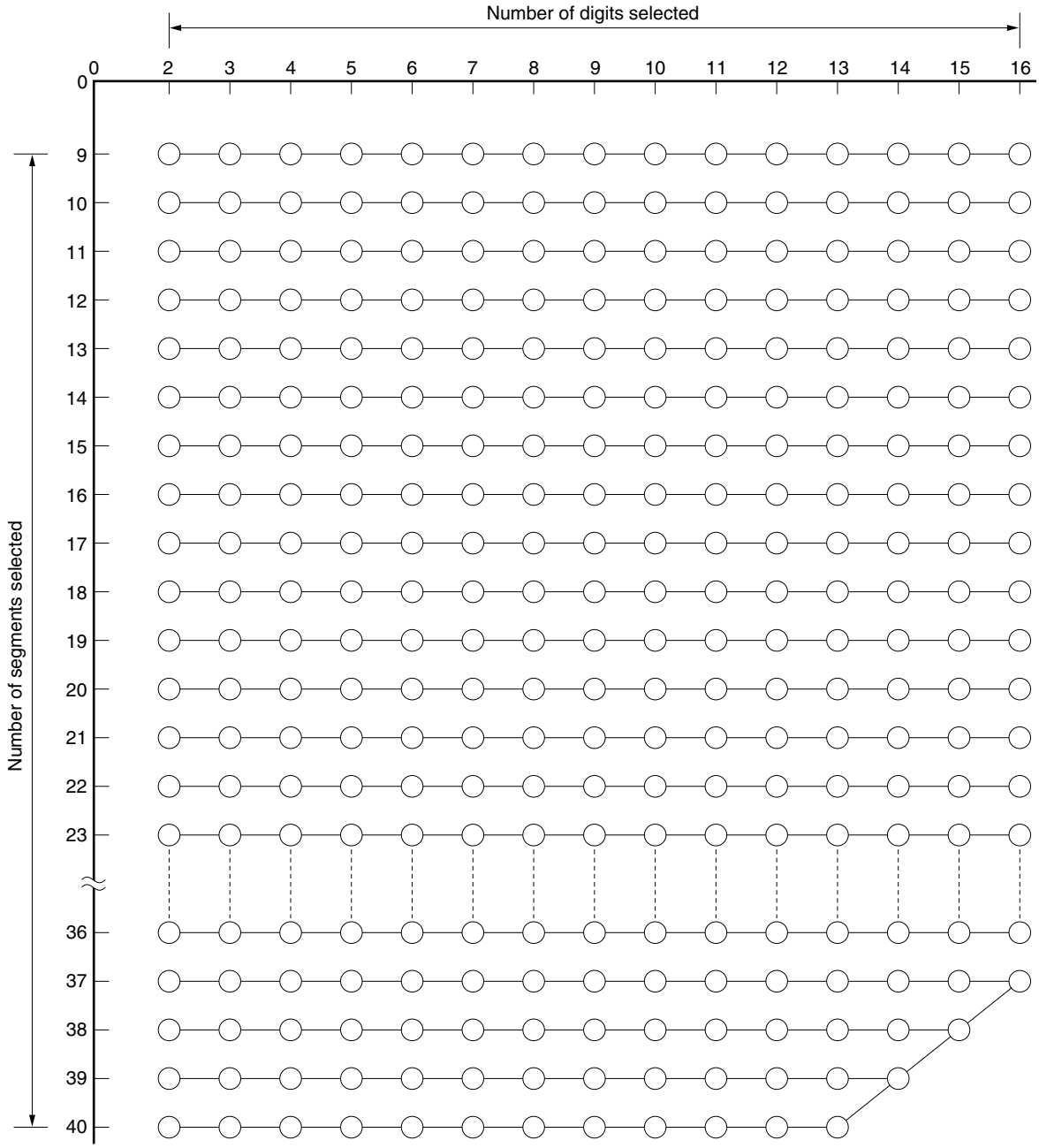
T_{DIG}: Pulse width of digit signal (Can be selected from 8 types using DSPM1)

Note The user can select the cut width of the segment signals by setting bits 1 to 3 (DIMS1 to DIMS3) of DSPM1. Therefore, actual output waveforms may be different from the above illustration and have the cut widths shown in Figure 15-6.

15.4 Selecting Display Mode

The number of segments and digits displayed by the VFD controller/driver depends on the display mode set.

Figure 15-8. Selection of Display Mode



Caution When the total number of digits and segments together exceeds 53, the digits have priority.

15.5 Display Mode and Display Output

The on-chip VFD controller/driver assigns pins FIP0 to FIP52/P127 to digit signals and segment signals (in this order). The number assigned is specified by display mode registers 0 and 1 (DSPM0 and DSPM1). The remaining pins are assigned as general-purpose ports.

The pin configuration for a 14-segment display is shown below as an example.

Figure 15-9. Pin Configuration for 14-Segment Display

Pin name	Number of display digits selected							
	Display stop	2	3	4	.....	14	15	16
FIP0	FIP0	T0	T0	T0		T0	T0	T0
FIP1	FIP1	T1	T1	T1		T1	T1	T1
FIP2	FIP2	S0	T2	T2		T2	T2	T2
FIP3	FIP3	S1	S0	T3		T3	T3	T3
FIP4	FIP4	S2	S1	S0		T4	T4	T4
FIP5	FIP5	S3	S2	S1		T5	T5	T5
FIP6	FIP6	S4	S3	S2		T6	T6	T6
FIP7	FIP7	S5	S4	S3		T7	T7	T7
FIP8	FIP8	S6	S5	S4		T8	T8	T8
FIP9	FIP9	S7	S6	S5		T9	T9	T9
FIP10	FIP10	S8	S7	S6		T10	T10	T10
FIP11	FIP11	S9	S8	S7		T11	T11	T11
FIP12	FIP12	S10	S9	S8		T12	T12	T12
FIP13/P80	P80	S11	S10	S9		T13	T13	T13
FIP14/P81	P81	S12	S11	S10	.....	S0	T14	T14
FIP15/P82	P82	S13	S12	S11		S1	S0	T15
FIP16/P83	P83	P83	S13	S12		S2	S1	S0
FIP17/P84	P84	P84	P84	S13		S3	S2	S1
FIP18/P85	P85	P85	P85	P85		S4	S3	S2
FIP19/P86	P86	P86	P86	P86		S5	S4	S3
FIP20/P87	P87	P87	P87	P87		S6	S5	S4
FIP21/P90	P90	P90	P90	P90		S7	S6	S5
FIP22/P91	P91	P91	P91	P91		S8	S7	S6
FIP23/P92	P92	P92	P92	P92		S9	S8	S7
FIP24/P93	P93	P93	P93	P93		S10	S9	S8
FIP25/P94	P94	P94	P94	P94		S11	S10	S9
FIP26/P95	P95	P95	P95	P95		S12	S11	S10
FIP27/P96	P96	P96	P96	P96		S13	S12	S11
FIP28/P97	P97	P97	P97	P97		P97	S13	S12
FIP29/P100	P100	P100	P100	P100		P100	P100	S13
FIP30/P101	P101	P101	P101	P101		P101	P101	P101
FIP31/P102	P102	P102	P102	P102		P102	P102	P102
⋮		⋮					⋮	
FIP51/P126	P126	P126	P126	P126		P126	P126	P126
FIP52/P127	P127	P127	P127	P127		P127	P127	P127

Remark T0 to T15: Display digit pins
S0 to S13: Segment pins

15.6 Display Data Memory

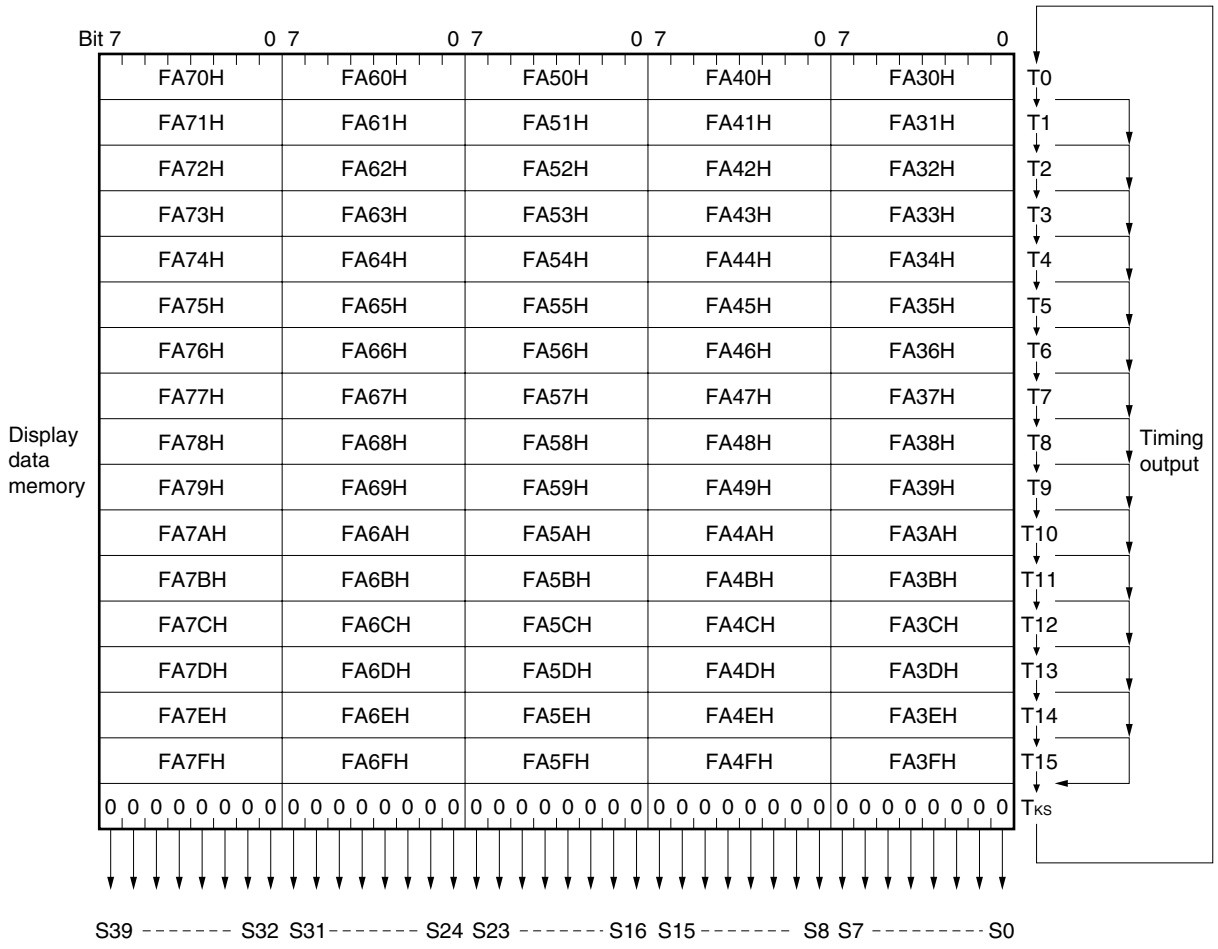
The display data memory is the area for storing the segment data to be displayed.

This memory is mapped at addresses FA30H to FA7FH. To display data on the VFD, the VFD controller reads the data stored in this memory regardless of the type of operations performed by the CPU (DMA operations).

The area not used for the display data can be used as normal RAM area.

At the key scan timing (T_{KS}), all segment outputs and digit outputs become “0” and the output latch data of ports 8, 9, 10, 11, and 12 are output to FIP37/P110 to FIP52/P127.

Figure 15-10. Relationship Between Display Data Memory Contents and Segment Output



15.7 Key Scan Flag and Key Scan Data

15.7.1 Key scan flag

The key scan flag (KSF) is set to 1 during the key scan timing and reset automatically to 0 during the display timing.

KSF is mapped at bit 7 of display mode register 0 (DSPM0) and can be tested one bit at a time. It cannot be written.

By testing the KSF, it can be determined if it is during the key scan timing and if the data input using keys is correct.

15.7.2 Key scan data

The data stored in ports 8, 9, 10, 11, and 12 are output from pins FIP13 to FIP52 at the key scan timing.

By changing the data output from ports 11 and 12 during the key scan timing, key scan can be performed using these pins FIP13 to FIP52.

Caution If, during the key scan timing, scanning is performed which causes both the segment and digit lines to turn ON at the same time, the display may be disturbed.

15.8 Light Leakage of VFD

Light may leak when a VFD is driven using the μPD780208 Subseries. Two possible causes are as follows.

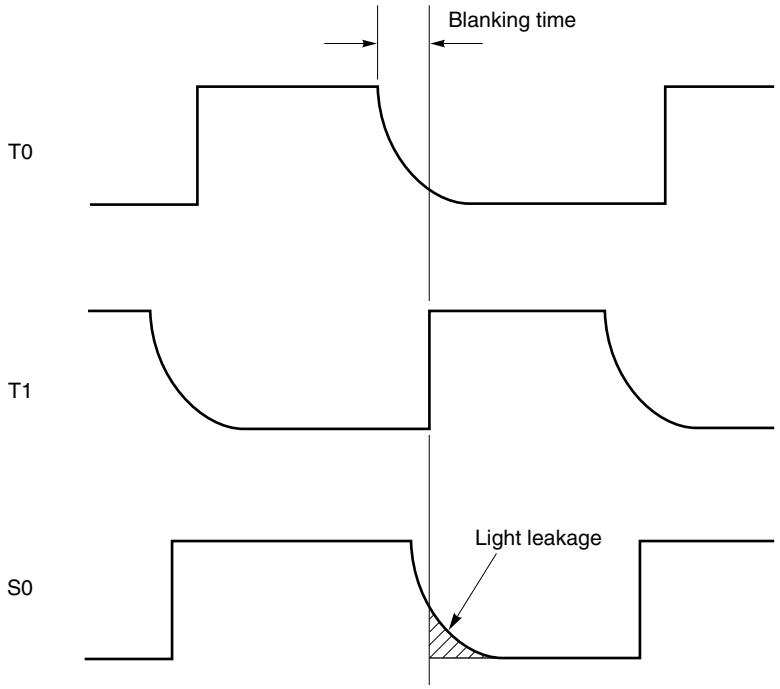
(1) Light leakage due to a short blanking time

Figure 15-11 shows the signal waveforms when only the first digit of two digits to be displayed is lit.

As shown in this figure, when the blanking time is short, the T1 signal rises before the segment signal disappears, resulting in light leakage.

Generally, as approximately 20 μs is required for the blanking time, consider the values set to display mode register 1 (DSPM1) carefully.

Figure 15-11. Light Leakage due to Short Blanking Time



(2) Light leakage due to capacitance between segment and grid of VFD

As shown in Figure 15-13, light may leak even if the blanking time is sufficient. As shown by C_{SG} in Figure 15-12, as there is capacitance between the grid and segment of the VFD, the timing signal pin voltage will be increased via C_{SG} when the segment signal turns on.

As shown in Figure 15-13, when this voltage exceeds the cut-off voltage (E_K), light will leak.

This spike noise voltage depends on the size of C_{SG} and the on-chip pull-down resistor (R_L). The greater the value of C_{SG} or the R_L value, the greater the voltage, making it easy for light to leak.

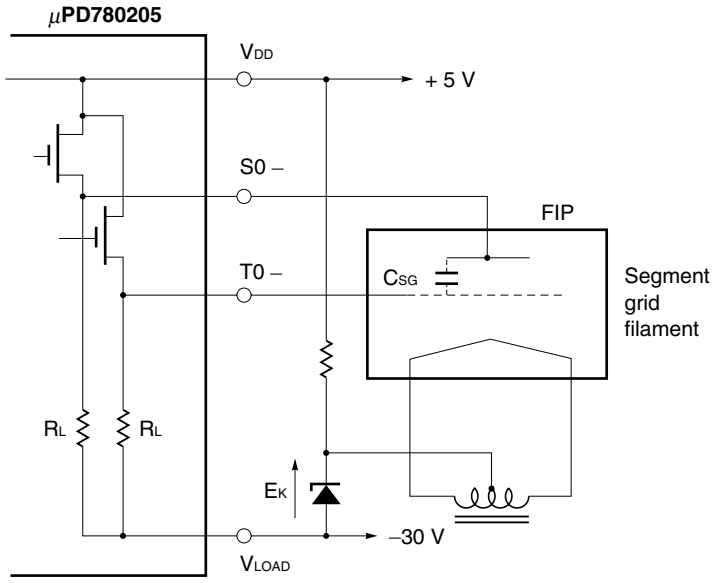
This C_{SG} value differs according to the area of the data displayed on the VFD. The greater the area, the greater C_{SG}. Consequently, pull-down resistor values with which light will not leak also depend on the size of the VFD.

As the value of the pull-down resistor incorporated by the mask option is comparatively great, in some cases, light leakage cannot be controlled with resistance only.

If the quality of the display is insufficient, increase the back bias (increase the E_K), place a filter over the VFD, or attach a 10 kΩ pull-down resistor externally to the timing signal pin.

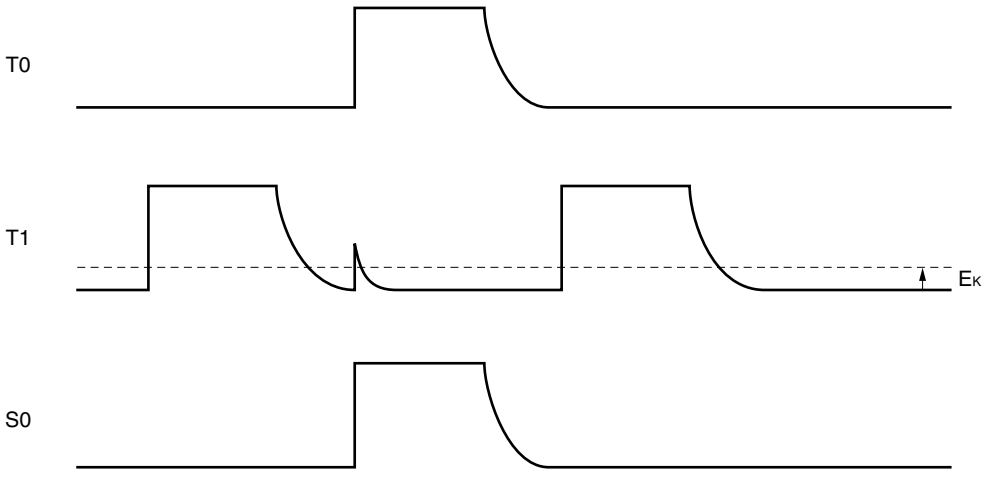
Depending on the duty cycle of the spike noise voltage for the whole display period, the ease with which light leaks due to C_{SG} varies. The fewer the number of digits displayed, the easier it is for light to leak. Lowering the luminance of the display is also effective.

Figure 15-12. Light Leakage due to C_{SG}



E_K: Cut-off voltage
R_L: On-chip pull-down resistor

Figure 15-13. Waveform of Light Leakage due to C_{SG}



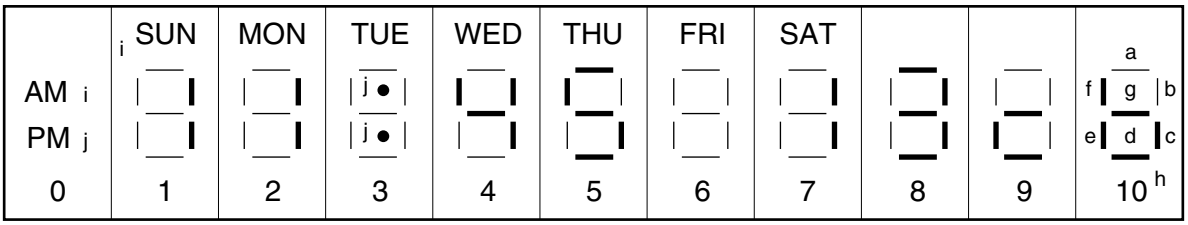
15.9 Display Examples

The μPD780208 Subseries has a VFD controller/driver that enables the following three types of VFD display. Display types can be switched by setting bit 5 (DSPM05) of display mode register 0 (DSPM0).

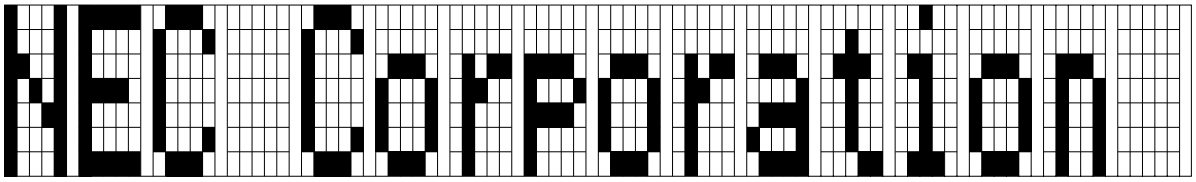
- Segment type (Display mode 1: DSPM05 = 0)
- Dot type (Display mode 1: DSPM05 = 0)
- Display type in which a segment spans two or more grids (Display mode 2: DSPM05 = 1)

The following figures show VFD display examples for each display type.

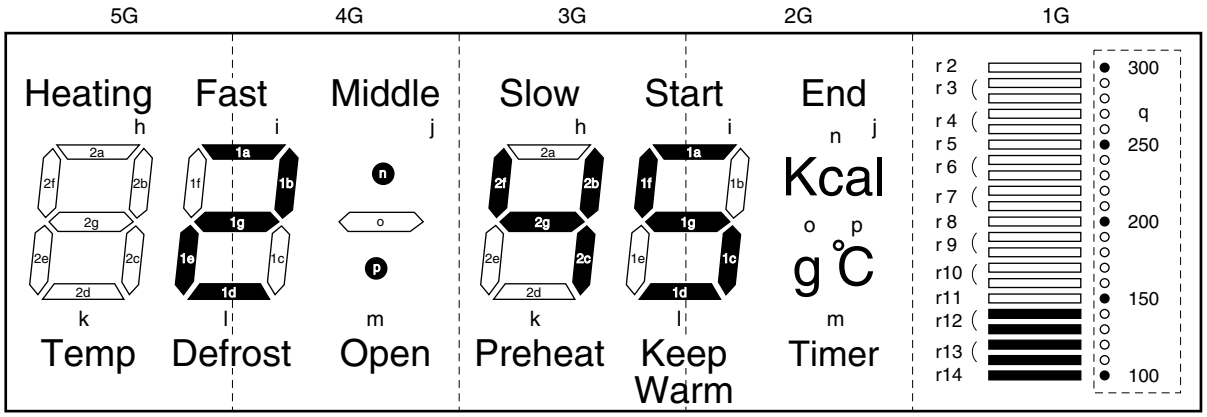
(1) Segment type: 10 segments x 11 digits



(2) Dot type: 35 segments x 16 digits



(3) Display type in which a segment spans two or more grids: 23 segments x 7 patterns

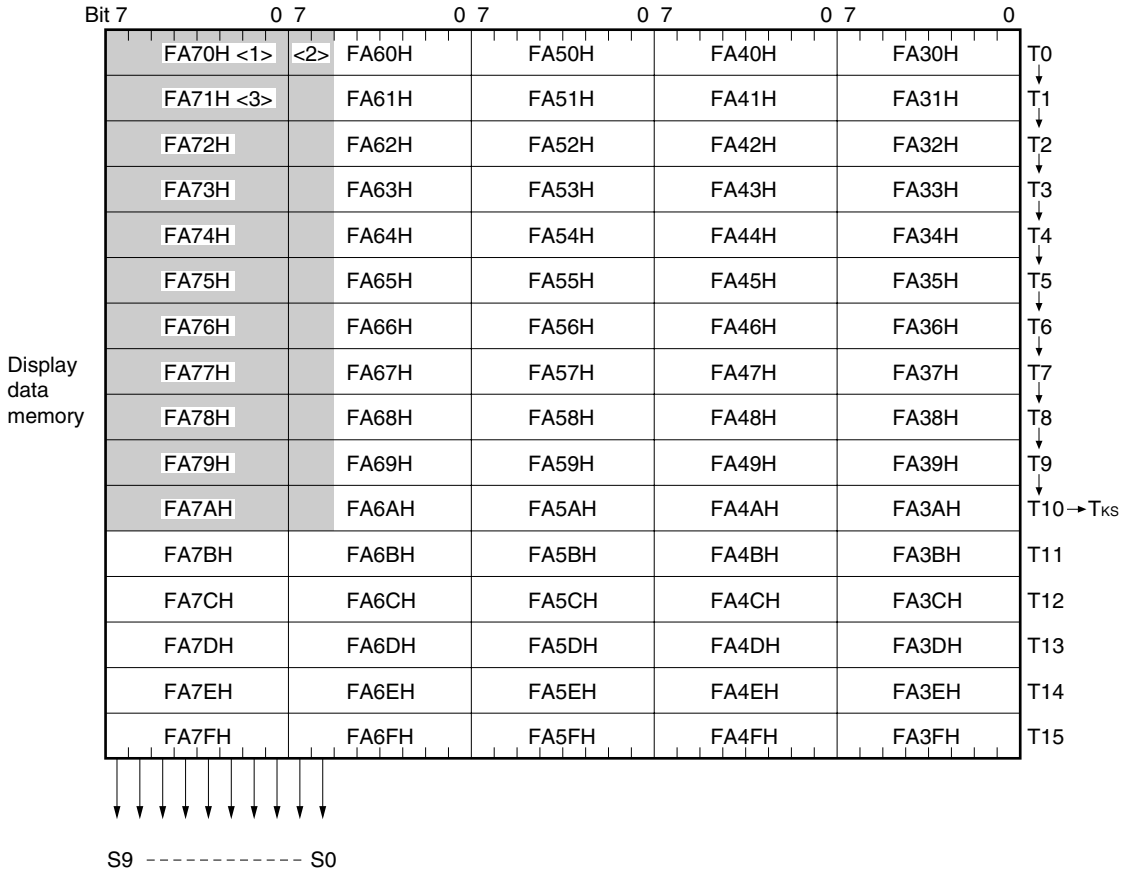


15.9.1 Segment type (display mode 1: DSPM05 = 0)

Figure 15-14 shows the display data memory configuration and data reading order when the device controls a 10-segment x 11-digit VFD display.

As “segment type” (display mode 1) is selected, the display data memory stores segment data.

Figure 15-14. Display Data Memory Configuration and Segment Data Reading Order (Segment Type)

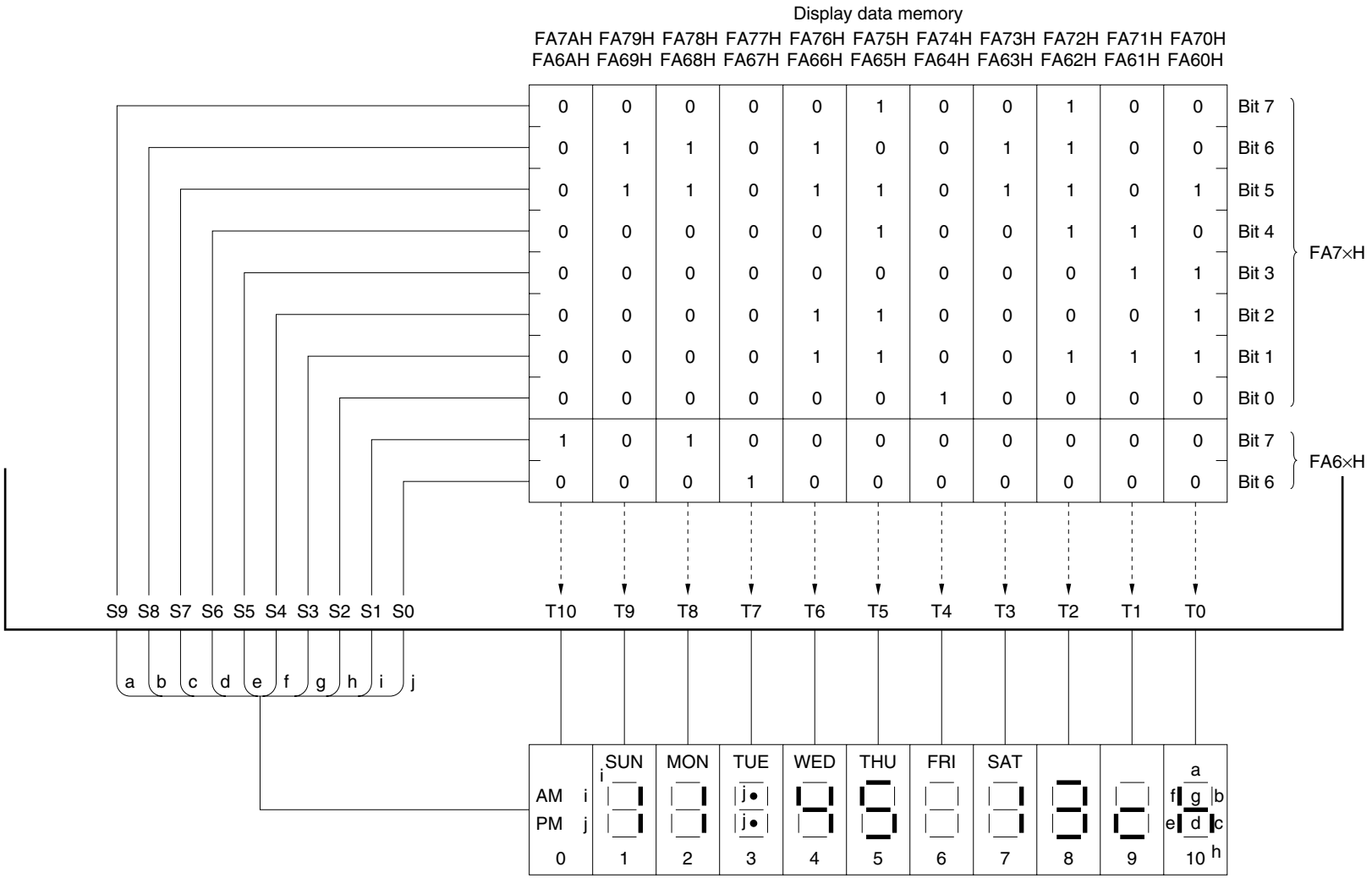


Remarks 1. <1> through <3> show the segment data reading order.

2. The shaded area shows the segment data storage area.

In the case of example **(1)** in **15.9**, the contents of the data memory indicated by shading are as shown in Figure 15-15.

Figure 15-15. Relationship Between Display Data Memory Contents and Segment Outputs in 10-Segment x 11-Digit Display Mode

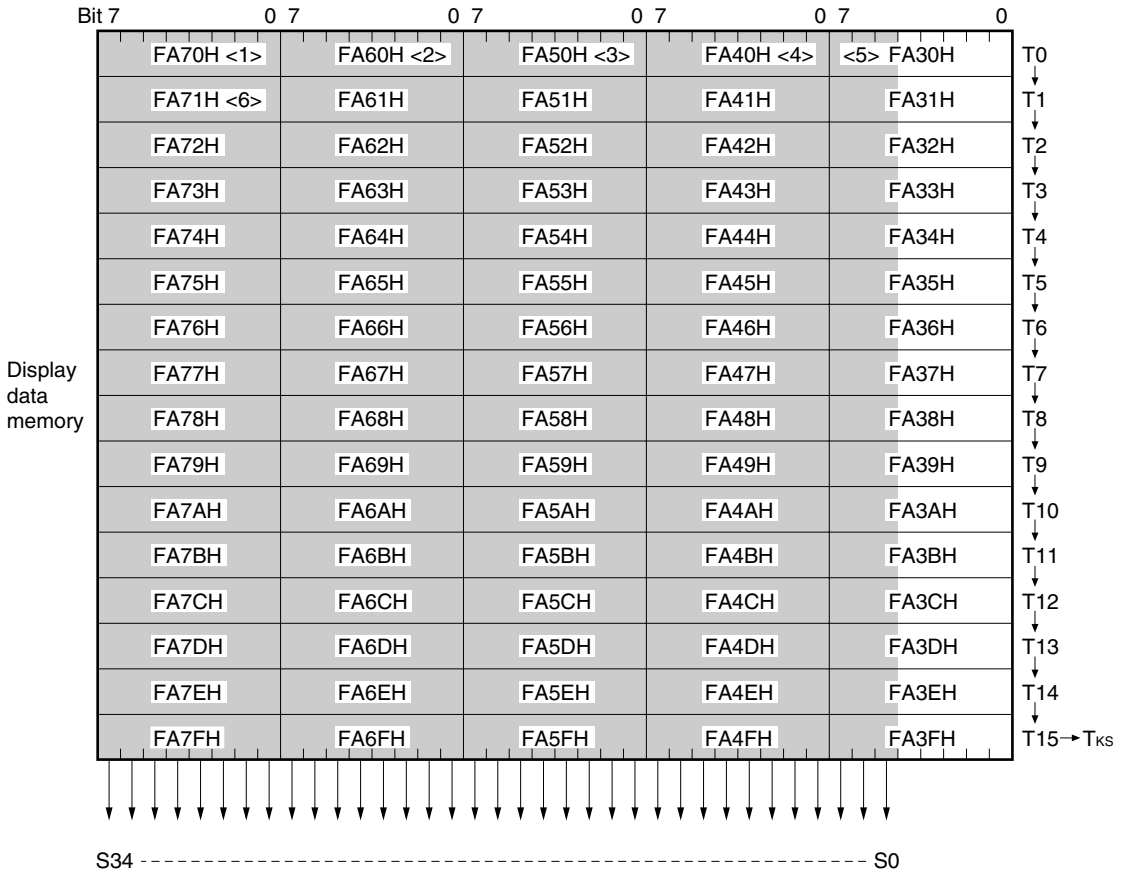


15.9.2 Dot type (display mode 1: DSPM05 = 0)

Figure 15-16 shows the display data memory configuration and data reading order when the device controls a 35-segment (5 x 7 dots) x 16-digit VFD display.

As “dot type” (display mode 1) is selected, the display data memory stores segment data.

Figure 15-16. Display Data Memory Configuration and Segment Data Reading Order (Dot Type)

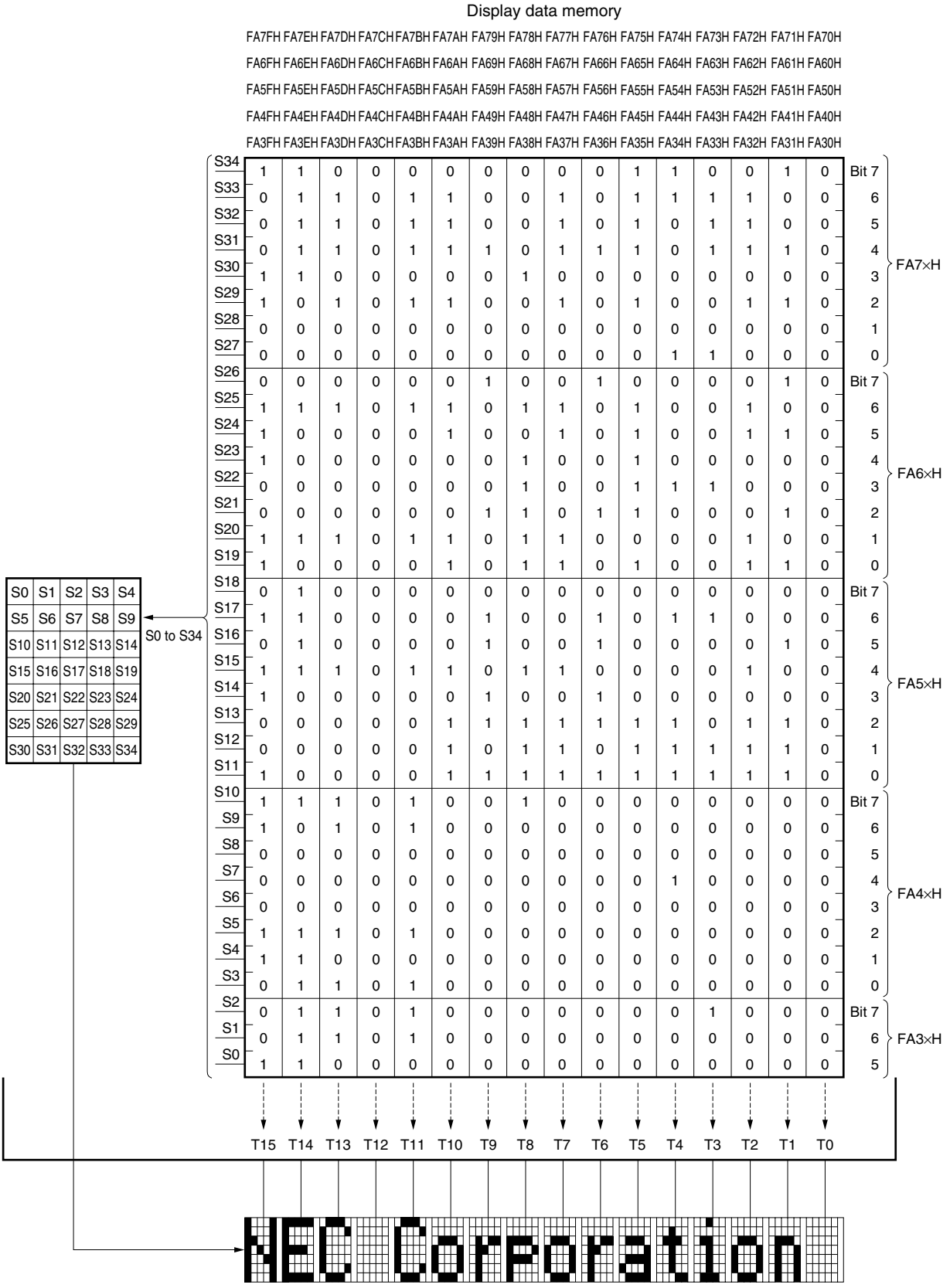


Remarks 1. <1> through <6> show the segment data reading order.

2. The shaded area shows the segment data storage area.

In the case of example **(2)** in **15.9**, the contents of the data memory indicated by shading are as shown in Figure 15-17.

Figure 15-17. Relationship Between Display Data Memory Contents and Segment Outputs in 35-Segment x 16-Digit Display Mode

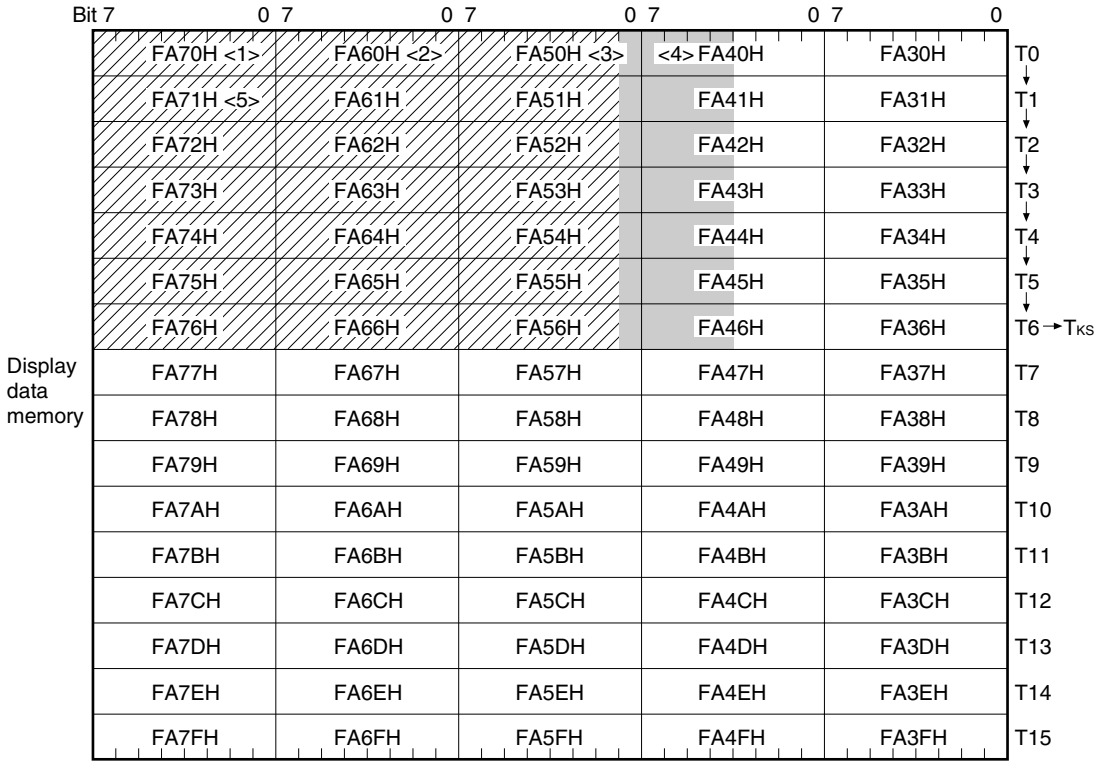


15.9.3 Display type in which a segment spans two or more grids (display mode 2: DSPM05 = 1)

In display mode 2, all of the display output data are stored in the display data memory.

Figure 15-18 shows the display data RAM configuration and data reading order in a 23-segment x 5-grid display.

Figure 15-18. Display Data Memory Configuration and Data Reading Order (Display Mode 2)



Remarks 1. <1> through <5> show the display output data reading order.

2. The slashed area shows the segment data storage area.

3. The shaded area shows the grid data storage area.

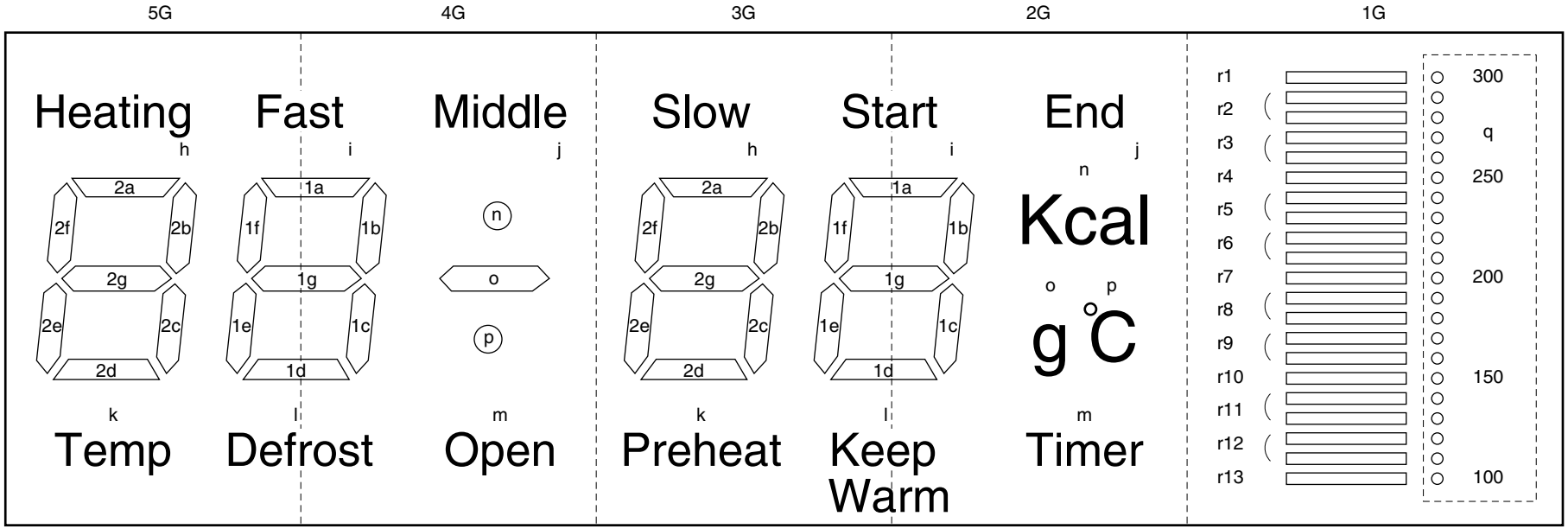
In the case of example **(3)** in **15.9**, the contents of the data memory areas indicated by shading and slashes are as shown in Figure 15-21.

T0 through T6 in display mode 2 are for display patterns. Therefore, designate bits 4 to 7 (DIGS0 to DIGS3) of display mode register 1 (DSPM1) as 7 patterns, and bits 0 to 4 (SEGS0 to SEGS4) of display mode register 0 (DSPM0) as 28 display outputs in total.

If there is some memory area where rewriting display output data is unnecessary, it should be masked by setting display mode register 2 (DSPM2).

Figure 15-19. Segment Connection Example

P23	P22	P21	P20	P19	P18	P17	P16	P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	
r1	r2	q	r8	r9	r10	–	–	–	–	–	r13	r12	r11	–	–	r7	r6	r5	r4	r3	–	–	1G
i	j	–	n	o	p	–	–	–	–	–	m	l	–	1d	–	1c	1g	–	1b	1a	–	–	2G
	–	h	–	–	–	2a	2b	2f	2g	2c	–		k		1e	–		1f	–		2d	2e	3G
	j	–	n	o	p	–	–	–	–	–	m		–		–	1c		–	1b		–	–	4G
	–	h	–	–	–	2a	2b	2f	2g	2c	–		k		1e	–		1f	–		2d	2e	5G



The light timing of each segment is discussed next. In display example **(3)** in **15.9**, a segment spans two grids (that is, 2G and 3G, 4G and 5G). Therefore, these segments will be lit at the timing from T0 to T6 as shown in Figure 15-20.

For example, when “Fast” is to be lit as shown in example **(3)** in **15.9**, the lighting timing must be T5 in Figure 15-20 because the “Fast” segment spans the 4G and 5G grids. In addition, it can be seen from Figure 15-3 that the “Fast” segment, that is, “i” segment which spans 4G and 5G, can be lit in the T5 cycle.

Figure 15-20. Grid Driving Timing

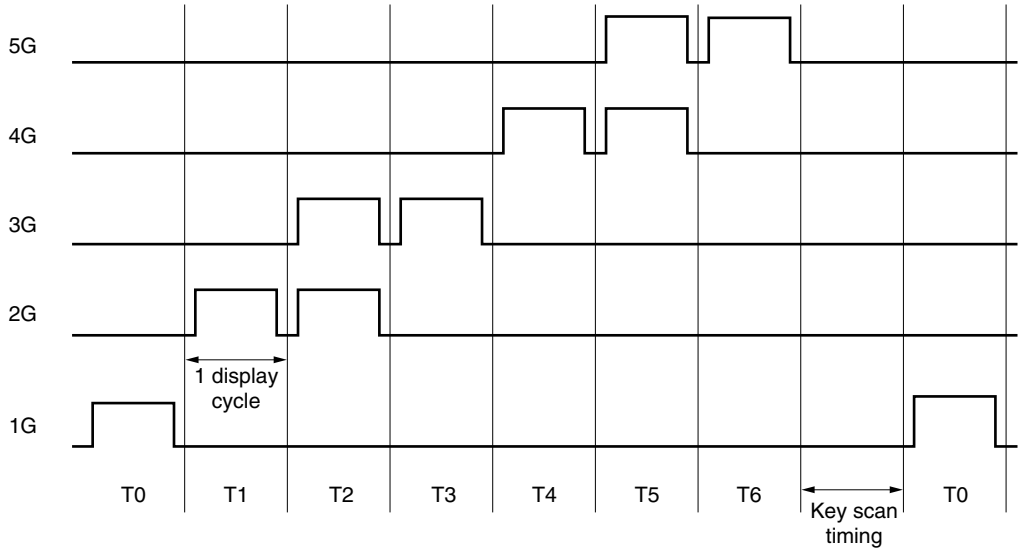
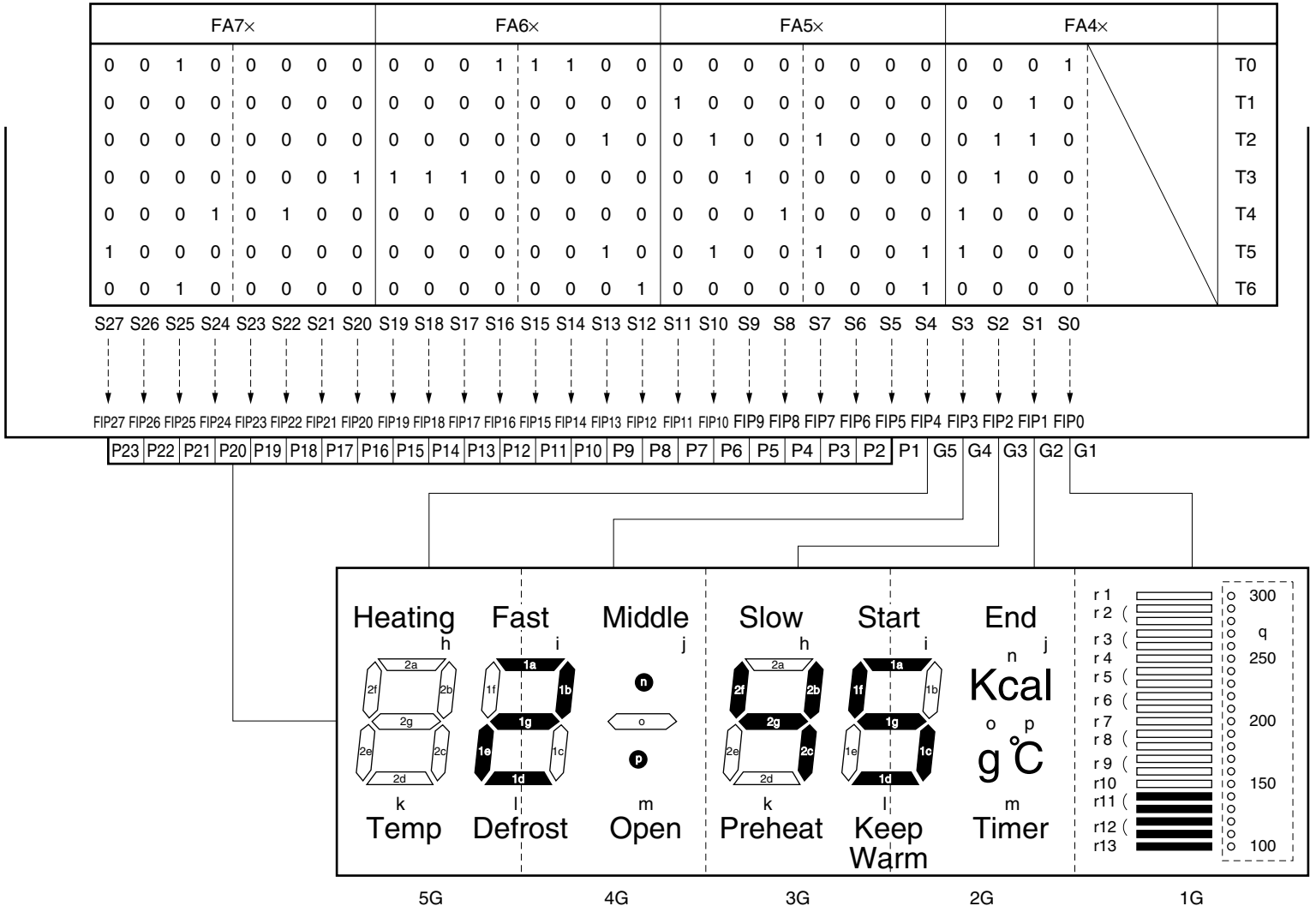


Table 15-3. Segment Lighting Timing

	Lighting Segment
T0	q, r1, r2, r3, r4, r5, r6, r7, r8, r9, r10, r11, r12, r13
T1	1b, 1c, j, m, n, o, p
T2	1a, 1g, 1d, i, l
T3	2a, 2b, 2c, 2d, 2e, 2f, 2g, h, k, 1e, 1f
T4	1b, 1c, j, m, n, o, p
T5	1a, 1g, 1d, i, l
T6	2a, 2b, 2c, 2d, 2e, 2f, 2g, h, k, 1e, 1f

Figure 15-21. Data Memory Status in 23-Segment x 5-Grid Display Mode



15.10 Calculating Total Power Dissipation

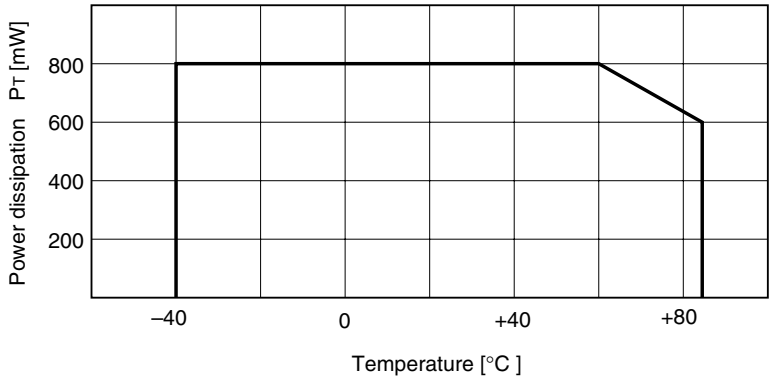
The total power dissipation of the μPD780208 Subseries is the sum of the values of the following three parts. Design your application set so that the sum is lower than the total power dissipation P_T stipulated in **Figure 15-22**. (The recommended operating condition is 80% or lower of the rated value.)

<1> CPU: The power consumed by the CPU and calculated with V_{DD} (max.) x I_{DD} (max.)

<2> Output pins: The power dissipation when the maximum current flows through the display output pins

<3> Pull-down resistors: The power consumed at the on-chip pull-down resistors connected to the display output pins

Figure 15-22. Allowable Total Power Dissipation P_T (T_A = –40 to +85°C)



The following example assumes the case where the display examples shown in **15.9** are displayed.

15.10.1 Segment type (display mode 1: DSPM05 = 0)

The calculation method for the total power dissipation in the case of the display example in Figure 15-23 is described below.

Example Assume the following conditions:

V_{DD} = 5 V ±10%, 5.0 MHz oscillation

Supply current (I_{DD}) = 21.6 mA

Display output: 11 grids × 10 segments (cut width = 1/16: when DIMS1 to DIMS3 = 000B)

Maximum current at the grid pin is 15 mA.

Maximum current at the segment pin is 3 mA.

At the key scan timing, display output pin is OFF.

Display output voltage: grid $V_{OD} = V_{DD} – 2$ V (voltage drop of 2 V)

segments $V_{OD} = V_{DD} – 0.4$ V (voltage drop of 0.4 V)

Fluorescent display control voltage (V_{LOAD}) = –35 V

Mask option pull-down resistor = 25 kΩ

By placing the above conditions in calculations <1> to <3>, the total dissipation can be worked out.

<1> CPU power dissipation: 5.5 V × 21.6 mA = 118.8 mW

<2> Output pin power dissipation:

$$\text{Grid} \quad (V_{DD} - V_{OD}) \times \frac{\text{Total current value of each grid}}{\text{No. of grids} + 1} \times \text{Digit width} \, (1 - \text{Cut width}) =$$

$$2\ \text{V} \times \frac{15\ \text{mA} \times 11\ \text{grids}}{11\ \text{grids} + 1} \times \left(1 - \frac{1}{16}\right) = 25.8\ \text{mW}$$

$$\text{Segment} \quad (V_{DD} - V_{OD}) \times \frac{\text{Total segment current value of illuminated dots}}{\text{No. of grids} + 1} \times \text{Digit width} \, (1 - \text{Cut width}) =$$

$$0.4\ \text{V} \times \frac{3\ \text{mA} \times 31\ \text{dots}}{11\ \text{grids} + 1} \times \left(1 - \frac{1}{16}\right) = 2.9\ \text{mW}$$

<3> Pull-down resistor power dissipation:

$$\text{Grid} \quad \frac{(V_{OD} - V_{LOAD})^2}{\text{Pull-down resistor value}} \times \frac{\text{No. of grids}}{\text{No. of grids} + 1} \times \text{Digit width} \, (1 - \text{Cut width}) =$$

$$\frac{(5.5\ \text{V} - 2\ \text{V} - (-35\ \text{V}))^2}{25\ \text{k}\Omega} \times \frac{11\ \text{grids}}{11\ \text{grids} + 1} \times \left(1 - \frac{1}{16}\right) = 50.9\ \text{mW}$$

$$\text{Segment} \quad \frac{(V_{OD} - V_{LOAD})^2}{\text{Pull-down resistor value}} \times \frac{\text{No. of illuminated dots}}{\text{No. of grids} + 1} \times \text{Digit width} \, (1 - \text{Cut width}) =$$

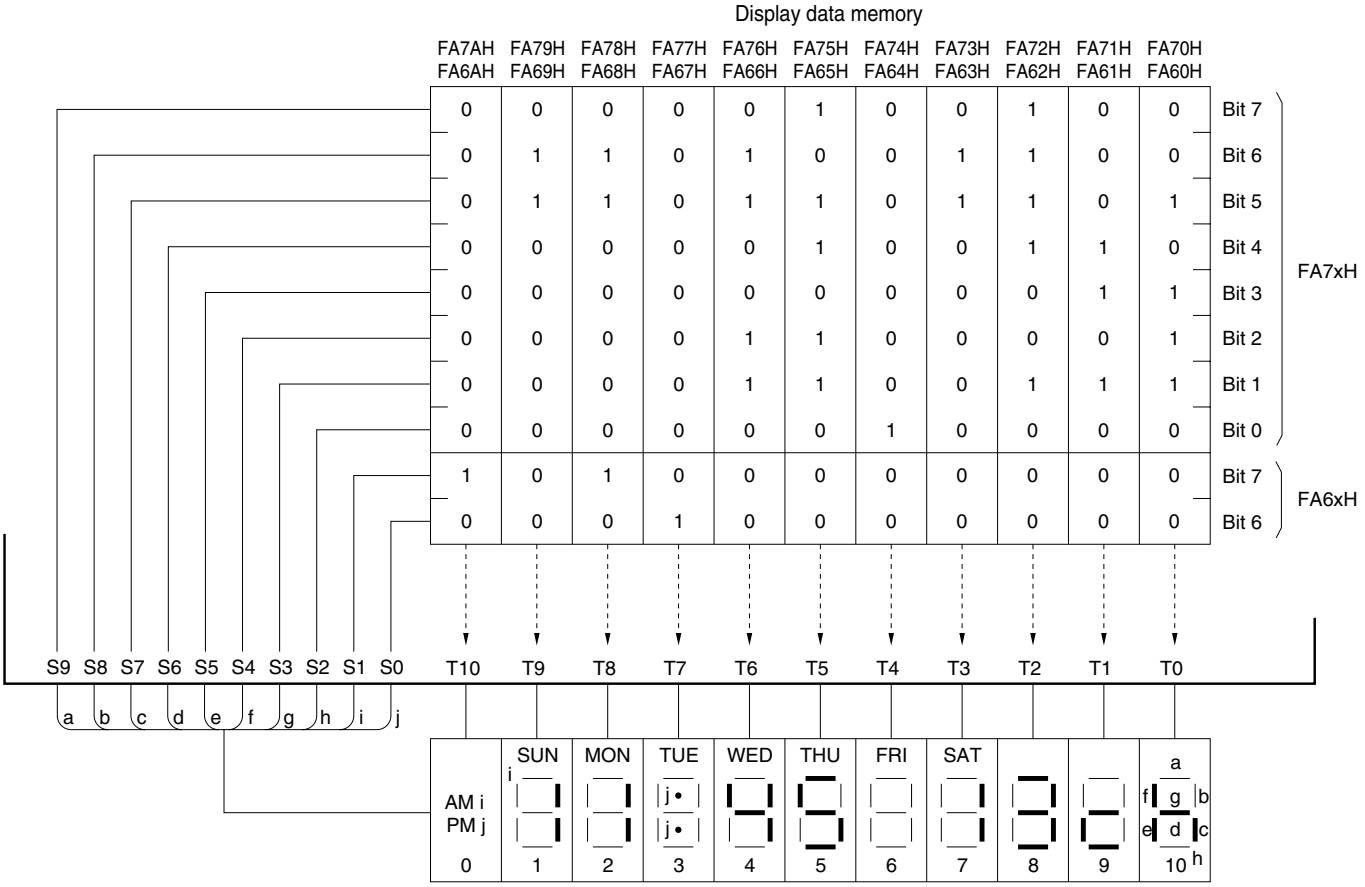
$$\frac{(5.5\ \text{V} - 0.4\ \text{V} - (-35\ \text{V}))^2}{25\ \text{k}\Omega} \times \frac{31\ \text{dots}}{11\ \text{grids} + 1} \times \left(1 - \frac{1}{16}\right) = 155.8\ \text{mW}$$

Total power dissipation = <1> + <2> + <3> = 118.8 + 25.8 + 2.9 + 50.9 + 155.8 = 354.2 mW

In this example, the power dissipation problem is cleared because the total power dissipation does not exceed the allowable total power dissipation rating shown in Figure 15-22.

Figure 15-23 shows a display example and display data for “segment type”.

Figure 15-23. Relationship Between Display Data Memory Contents and Segment Outputs in 10-Segment x 11-Digit Display Mode



15.10.2 Dot type (display mode 1: DSPM05 = 0)

The calculation method for the total power dissipation in the case of the display example in Figure 15-24 is described below.

Example Assume the following conditions:

V_{DD} = 5 V ±10%, 5.0 MHz oscillation

Supply current (I_{DD}) = 21.6 mA

Display output: 16 grids × 35 segments (cut width = 1/16: when DIMS1 to DIMS3 = 000B)

Maximum current at the grid pin is 15 mA.

Maximum current at the segment pin is 3 mA.

At the key scan timing, display output pin is OFF.

Display output voltage: grid $V_{OD} = V_{DD} - 2$ V (voltage drop of 2 V)

segments $V_{OD} = V_{DD} - 0.4$ V (voltage drop of 0.4 V)

Fluorescent display control voltage (V_{LOAD}) = –35 V

Mask option pull-down resistor = 25 kΩ

By placing the above conditions in calculation <1> to <3>, the total dissipation can be worked out.

<1> CPU power dissipation: 5.5 V × 21.6 mA = 118.8 mW

<2> Output pin power dissipation:

Grid $(V_{DD} - V_{OD}) \times \dfrac{\text{Total current value of each grid}}{\text{No. of grids + 1}} \times \text{Digit width } (1 - \text{Cut width}) =$

$$2\text{ V} \times \frac{15\text{ mA} \times 16\text{ grids}}{16\text{ grids} + 1} \times \left(1 - \frac{1}{16}\right) = 26.5\text{ mW}$$

Segment $(V_{DD} - V_{OD}) \times \dfrac{\text{Total segment current value of illuminated dots}}{\text{No. of grids + 1}} \times \text{Digit width } (1 - \text{Cut width}) =$

$$0.4\text{ V} \times \frac{3\text{ mA} \times 168\text{ dots}}{16\text{ grids} + 1} \times \left(1 - \frac{1}{16}\right) = 11.1\text{ mW}$$

<3> Pull-down resistor power dissipation:

Grid $\dfrac{(V_{OD} - V_{LOAD})^2}{\text{Pull-down resistor value}} \times \dfrac{\text{No. of grids}}{\text{No. of grids + 1}} \times \text{Digit width } (1 - \text{Cut width}) =$

$$\frac{(5.5\text{ V} - 2\text{ V} - (-35\text{ V}))^2}{25\text{ k}\Omega} \times \frac{16\text{ grids}}{16\text{ grids} + 1} \times \left(1 - \frac{1}{16}\right) = 52.3\text{ mW}$$

Segment $\dfrac{(V_{OD} - V_{LOAD})^2}{\text{Pull-down resistor value}} \times \dfrac{\text{No. of illuminated dots}}{\text{No. of grids + 1}} \times \text{Digit width } (1 - \text{Cut width}) =$

$$\frac{(5.5\text{ V} - 0.4\text{ V} - (-35\text{ V}))^2}{25\text{ k}\Omega} \times \frac{168\text{ dots}}{16\text{ grids} + 1} \times \left(1 - \frac{1}{16}\right) = 595.9\text{ mW}$$

Total power dissipation = <1> + <2> + <3> = 118.8 + 26.5 + 11.1 + 52.3 + 595.9 = 804.6 mW

In this example, the total power dissipation exceeds the allowable total power dissipation rating shown in Figure 15-22. In this case, the power dissipation can be lowered by reducing the number of enabled on-chip pull-down resistors.

Next, calculation expressions are shown for the display example where on-chip pull-down resistors are enabled for S0 through S24 only.

<3> Pull-down resistor power dissipation:

$$\text{Grid}\quad \frac{(V_{OD} - V_{LOAD})^2}{\text{Pull-down resistor value}} \times \frac{\text{No. of grids}}{\text{No. of grids} + 1} \times \text{Digit width} \times (1 - \text{Cut width}) =$$

$$\frac{(5.5\text{ V} - 2\text{ V} - (-35\text{ V}))^2}{25\text{ k}\Omega} \times \frac{16\text{ grids}}{16\text{ grids} + 1} \times \left(1 - \frac{1}{16}\right) = 52.3\text{ mW}$$

$$\text{Segment}\quad \frac{(V_{OD} - V_{LOAD})^2}{\text{Pull-down resistor value}} \times \frac{\text{No. of illuminated dots}}{\text{No. of grids} + 1} \times \text{Digit width} \times (1 - \text{Cut width}) =$$

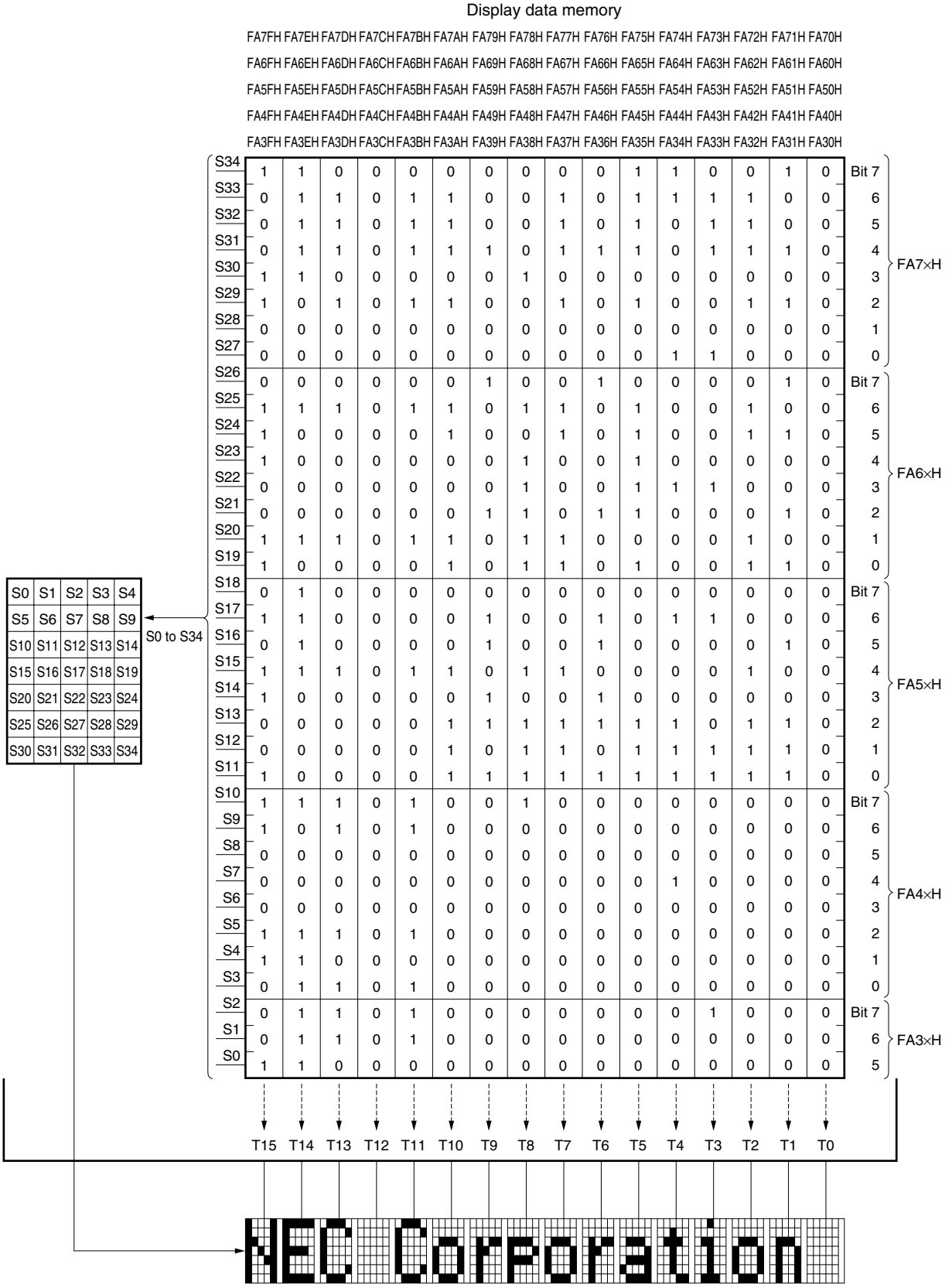
$$\frac{(5.5\text{ V} - 0.4\text{ V} - (-35\text{ V}))^2}{25\text{ k}\Omega} \times \frac{110\text{ dots}}{16\text{ grids} + 1} \times \left(1 - \frac{1}{16}\right) = 390.2\text{ mW}$$

Total power dissipation = <1> + <2> + <3> = 118.8 + 26.5 + 11.1 + 52.3 + 390.2 = 598.9 mW

In this manner, design the system so that the power dissipation does not exceed the allowable total power dissipation rating.

Figure 15-24 shows a display example and display data for “dot type”.

Figure 15-24. Relationship Between Display Data Memory Contents and Segment Outputs in 35-Segment x 16-Digit Display Mode



15.10.3 Display type in which a segment spans two or more grids (display mode 2: DSPM05 = 1)

The calculation method for the total power dissipation in the case of the display example in Figure 15-26 is described below.

Example Assume the following conditions:

V_{DD} = 5 V ±10%, 5.0 MHz oscillation
Supply current (I_{DD}) = 21.6 mA
Display output: 23 segments × 7 patterns (cut width = 1/16: when DIMS1 to DIMS3 = 000B)
Maximum current at the display output pin is 15 mA.
Display output voltage (V_{OD}) = V_{DD} – 2 V (voltage drop of 2 V)
Fluorescent display control voltage (V_{LOAD}) = –35 V
Mask option pull-down resistor = 25 kΩ

By placing the above conditions in calculation <1> to <3>, the total dissipation can be worked out.

<1> CPU power dissipation: 5.5 V × 21.6 mA = 118.8 mW

<2> Output pin power dissipation:

$$(V_{DD} - V_{OD}) \times \frac{\text{Total current value of each grid}}{\text{No. of grids} + 1} \times \text{Digit width} \ (1 - \text{Cut width}) =$$

$$2\ \text{V} \times \frac{15\ \text{mA} \times 9\ \text{grids}}{7\ \text{grids} + 1} \times \left(1 - \frac{1}{16}\right) = 31.6\ \text{mW}$$

<3> Pull-down resistor power dissipation:

Grid

$$\frac{(V_{OD} - V_{LOAD})^2}{\text{Pull-down resistor value}} \times \frac{\text{No. of grids}}{\text{No. of grids} + 1} \times \text{Digit width} \ (1 - \text{Cut width}) =$$

$$\frac{(5.5\ \text{V} - 2\ \text{V} - (-35\ \text{V}))^2}{25\ \text{k}\Omega} \times \frac{9\ \text{grids}}{7\ \text{grids} + 1} \times \left(1 - \frac{1}{16}\right) = 62.5\ \text{mW}$$

Segment

$$\frac{(V_{OD} - V_{LOAD})^2}{\text{Pull-down resistor value}} \times \frac{\text{No. of illuminated dots}}{\text{No. of grids} + 1} \times \text{Digit width} \ (1 - \text{Cut width}) =$$

$$\frac{(5.5\ \text{V} - 0.5\ \text{V} - (-35\ \text{V}))^2}{25\ \text{k}\Omega} \times \frac{22\ \text{dots}}{7\ \text{grids} + 1} \times \left(1 - \frac{1}{16}\right) = 152.8\ \text{mW}$$

Total power dissipation = <1> + <2> + <3> = 118.8 + 31.6 + 62.5 + 152.8 = 365.7 mW

In this example, the power dissipation problem is cleared because the total power dissipation does not exceed the allowable total power dissipation rating shown in Figure 15-22.

Figure 15-25 shows the grid driving timing, and Figure 15-26 shows a display example and display data of a display type in which a segment spans two or more grids.

Figure 15-25. Grid Driving Timing

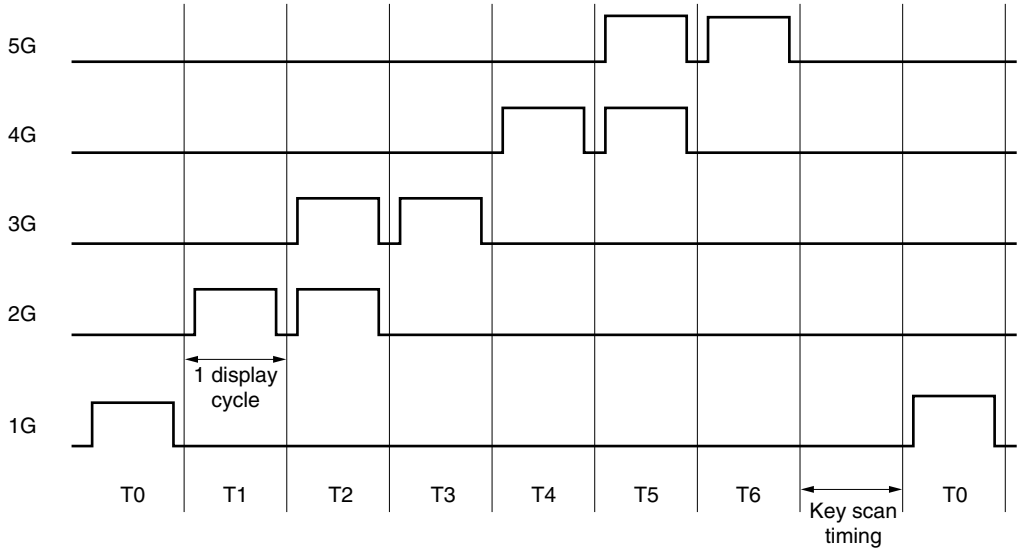
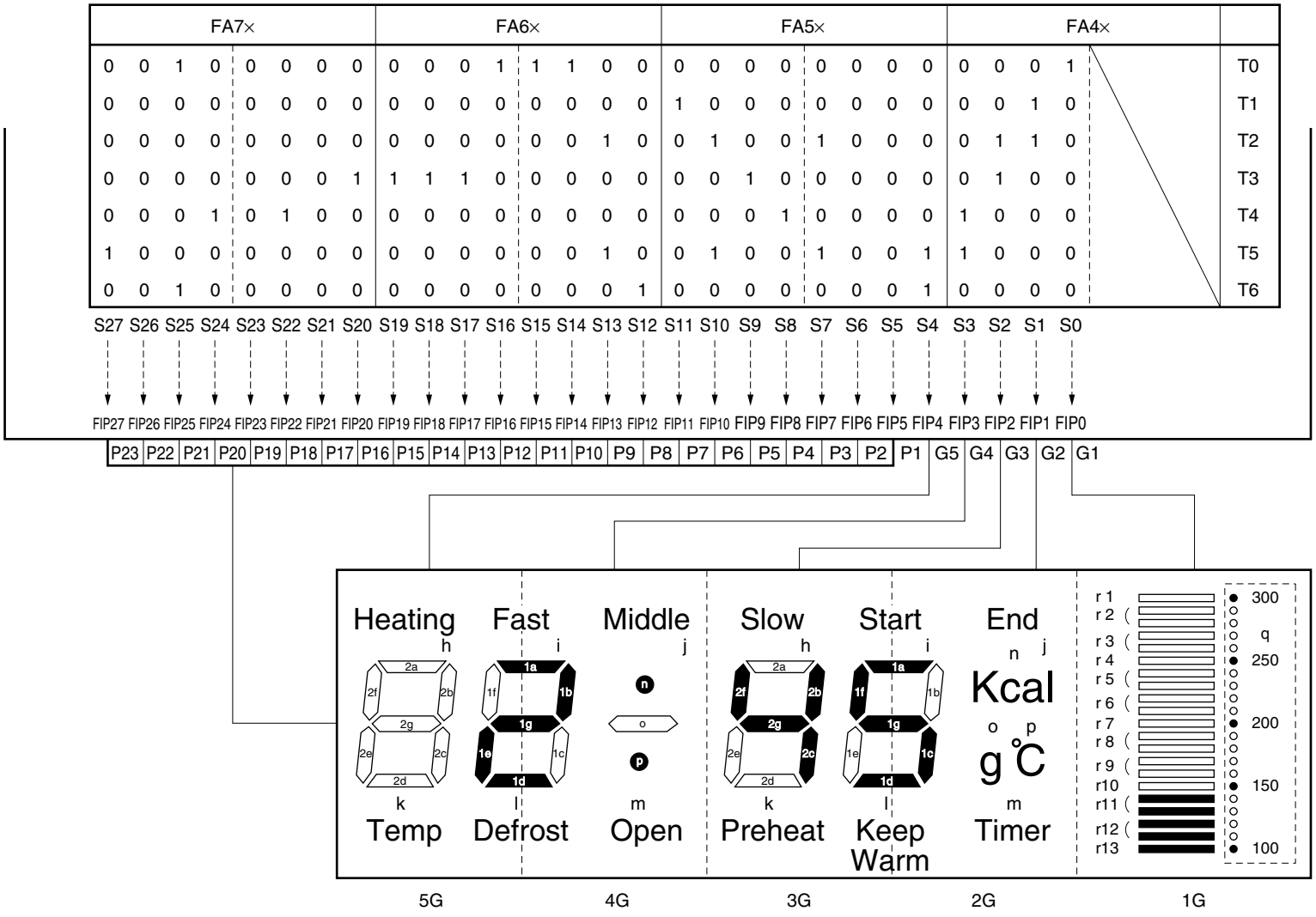


Figure 15-26. Data Memory Status in 23-Segment x 5-Grid Display Mode



CHAPTER 16 INTERRUPT AND TEST FUNCTIONS

16.1 Interrupt Function Types

The following three types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally (that is, even in the interrupt disabled state). It does not undergo interrupt priority control and is given top priority over all other interrupt requests.

A standby release signal is generated.

One interrupt request from the watchdog timer is provided as a non-maskable interrupt.

(2) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into a high interrupt priority group and a low interrupt priority group by setting the priority specification flag register (PR0L and PR0H).

Multiple interrupt servicing of high-priority interrupts can be applied to low-priority interrupts. If two or more interrupts with the same priority are simultaneously generated, each interrupt has a predetermined priority (see **Table 16-1**).

A standby release signal is generated.

Four external interrupt requests and 9 internal interrupt requests are provided as maskable interrupts.

(3) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even in the interrupt disabled state. The software interrupt does not undergo interrupt priority control.

16.2 Interrupt Sources and Configuration

A total of 15 interrupt sources are provided including non-maskable, maskable, and software interrupts (see **Table 16-1**).

Table 16-1. Interrupt Source List

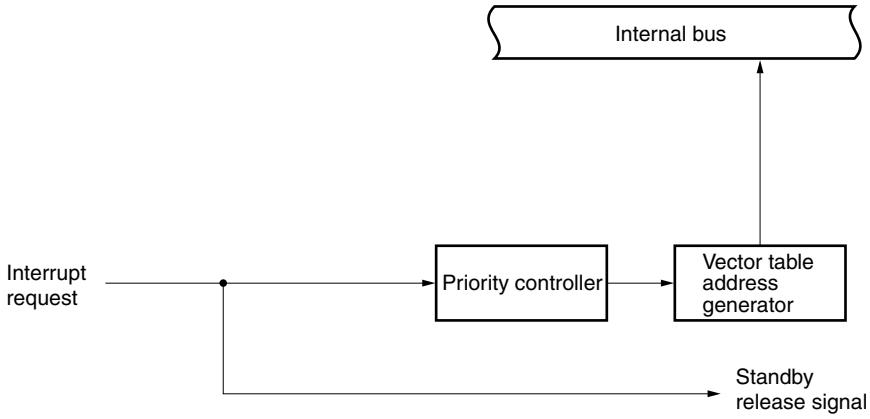
Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Non-maskable	—	INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (with interval timer mode selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	(D)
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTCSI0	End of serial interface channel 0 transfer	Internal	000EH	(B)
	6	INTCSI1	End of serial interface channel 1 transfer		0010H	
	7	INTTM3	Reference time interval signal from watch timer		0012H	
	8	INTTM0	Generation of 16-bit timer/event counter match signal		0014H	
	9	INTTM1	Generation of 8-bit timer/event counter 1 match signal		0016H	
	10	INTTM2	Generation of 8 bit timer/event counter 2 match signal		0018H	
	11	INTAD	End of A/D converter conversion		001AH	
	12	INTKS	Key scan timing from VFD controller/driver		001CH	
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 12 is the lowest priority.

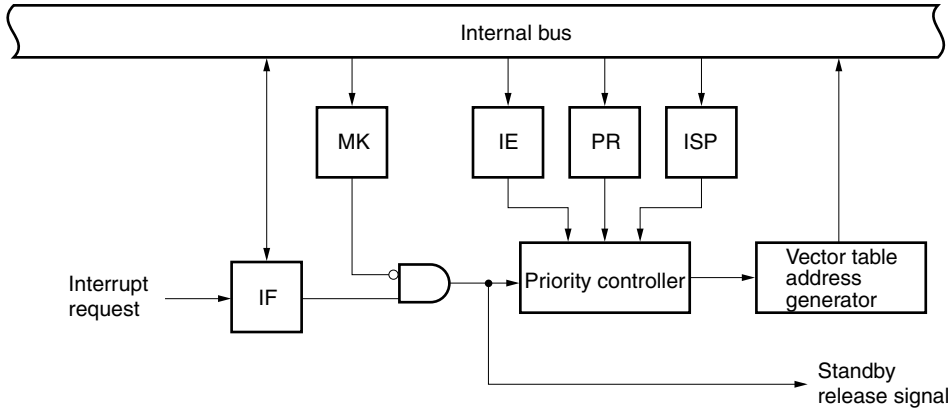
2. Basic configuration types (A) through (E) correspond to (A) through (E) on the following pages.

Figure 16-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt (INTP0)

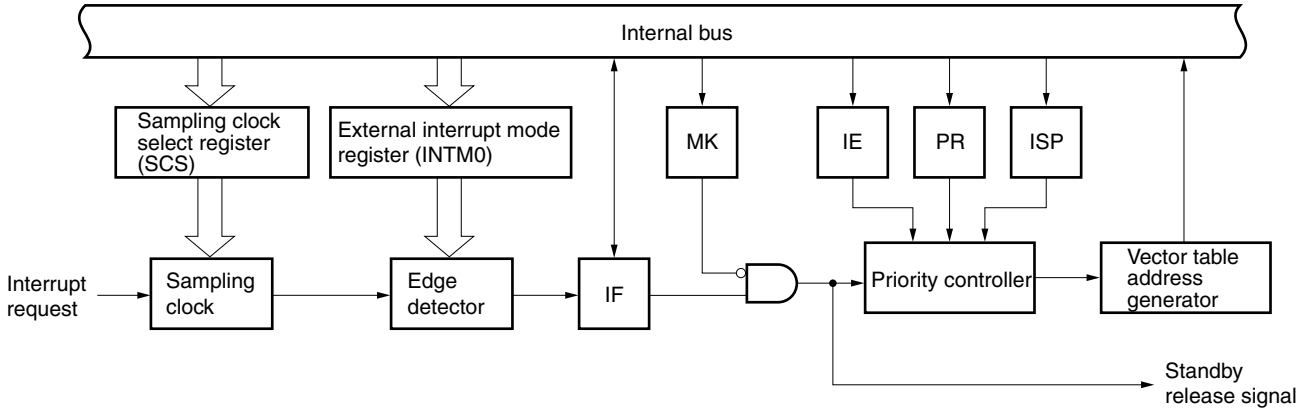
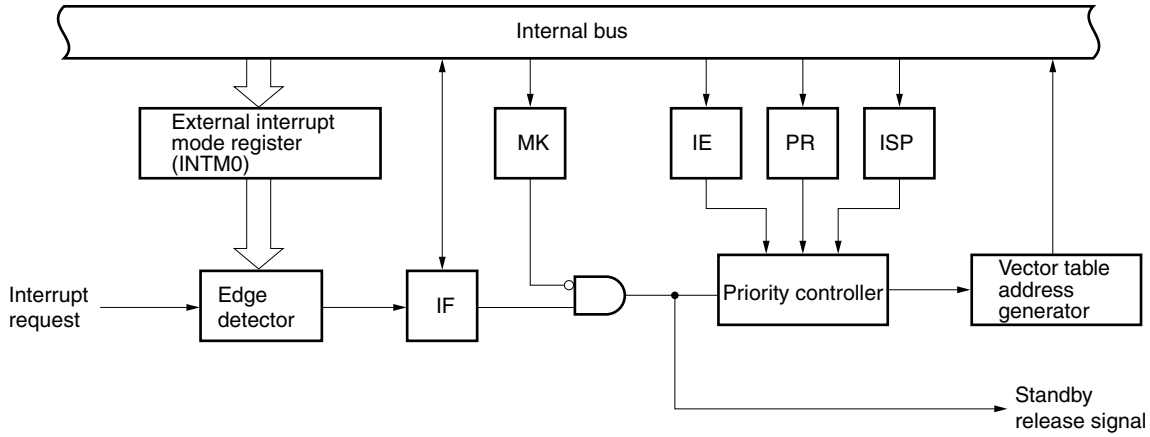
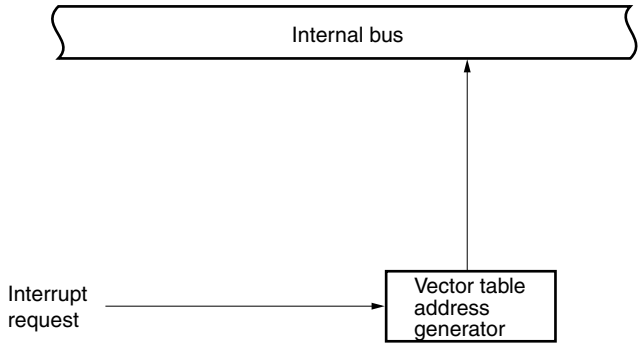


Figure 16-1. Basic Configuration of Interrupt Function (2/2)

(D) External maskable interrupt (except INTP0)



(E) Software interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
ISP: In-service priority flag
MK: Interrupt mask flag
PR: Priority specification flag

16.3 Interrupt Function Control Registers

The following six types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H)
- Interrupt mask flag register (MK0L, MK0H)
- Priority specification flag register (PR0L, PR0H)
- External interrupt mode register (INTM0)
- Sampling clock select register (SCS)
- Program status word (PSW)

Table 16-2 gives a listing of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 16-2. Various Flags Corresponding to Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTWDT	TMIF4	IF0L	TMMK4	MK0L	TMPR4	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTCSI0	CSIIF0		CSIMK0		CSIPR0	
INTCSI1	CSIIF1		CSIMK1		CSIPR1	
INTTM3	TMIF3		TMMK3		TMPR3	
INTTM0	TMIF0	IF0H	TMMK0	MK0H	TMPR0	PR0H
INTTM1	TMIF1		TMMK1		TMPR1	
INTTM2	TMIF2		TMMK2		TMPR2	
INTAD	ADIF		ADMK		ADPR	
INTKS	KSIF		KSMK		KSPR	

(1) Interrupt request flag registers (IF0L, IF0H)

The interrupt request flag is set to 1 when the corresponding interrupt request is generated or an instruction is executed. It is cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon application of $\overline{\text{RESET}}$.

IF0L and IF0H are set with a 1-bit or 8-bit memory manipulation instruction. If IF0L and IF0H are used as a 16-bit register IF0, use a 16-bit memory manipulation instruction for setting.

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

Figure 16-2. Format of Interrupt Request Flag Register

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
IF0L	TMIF3	CSIIF1	CSIIF0	PIF3	PIF2	PIF1	PIF0	TMIF4	FFE0H	00H	R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
IF0H	0	0	WTIF[Note]	KSIF	ADIF	TMIF2	TMIF1	TMIF0	FFE1H	00H	R/W

xxIF	Interrupt request flag
0	No interrupt request signal
1	Interrupt request signal is generated: Interrupt request state

Note WTIF is the test input flag. A vectored interrupt request is not generated.

Cautions 1. The TMIF4 flag is R/W enabled only when the watchdog timer is used as an interval timer. If the watchdog timer is used in watchdog timer mode 1, set the TMIF4 flag to 0.

2. Always set bits 6 and 7 of IF0H to 0.

★ **3. When an interrupt is acknowledged, the interrupt request flag is automatically cleared, and then servicing of the interrupt routine is started.**

★ **4. When the interrupt request flag register is manipulated (including by a 1-bit memory manipulation instruction), if an interrupt request corresponding to another flag in the same register is generated, the flag corresponding to that interrupt request may not be set to 1.**

(2) Interrupt mask flag registers (MK0L, MK0H)

The interrupt mask flag is used to enable/disable the corresponding maskable interrupt servicing and to set standby clear enable/disable.

MK0L and MK0H are set with a 1-bit or 8-bit memory manipulation instruction. If MK0L and MK0H are used as a 16-bit register MK0, use a 16-bit memory manipulation instruction for setting.

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

Figure 16-3. Format of Interrupt Mask Flag Register

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
MK0L	TMMK3	CSIMK1	CSIMK0	PMK3	PMK2	PMK1	PMK0	TMMK4	FFE4H	FFH	R/W

	7	6	<5>	<4>	<3>	<2>	<1>	<0>			
MK0H	1	1	WTMK[Note]	KSMK	ADMK	TMMK2	TMMK1	TMMK0	FFE5H	FFH	R/W

xxMK	Interrupt servicing and standby mode control
0	Interrupt servicing enabled, standby mode release enabled
1	Interrupt servicing disabled, standby mode release disabled

Note WTMK controls standby mode release enable/disable. This bit does not control the interrupt function.

Cautions 1. If the TMMK4 flag is read when the watchdog timer is used in watchdog timer mode 1, the MK0 value becomes undefined.

2. Because port 0 has an alternate function as an 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, the interrupt mask flag should be set to 1 before using the output mode.

3. Always set bits 6 and 7 of MK0H to 1.

(3) Priority specification flag registers (PR0L, PR0H)

The priority specification flag is used to set the corresponding maskable interrupt priority order.

PR0L and PR0H are set with a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H are used as a 16-bit register PR0, use a 16-bit memory manipulation instruction for setting.

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

Figure 16-4. Format of Priority Specification Flag Register

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
PR0L	TMPR3	CSIPR1	CSIPR0	PPR3	PPR2	PPR1	PPR0	TMPR4	FFE8H	FFH	R/W

	7	6	5	<4>	<3>	<2>	<1>	<0>			
PR0H	1	1	1	KSPR	ADPR	TMPR2	TMPR1	TMPR0	FFE9H	FFH	R/W

xxPR	Priority level selection
0	High priority level
1	Low priority level

Cautions 1. When the watchdog timer is used in watchdog timer mode 1, set the TMPR4 flag to 1.

2. Always set bits 5 to 7 of PR0H to 1.

(4) External interrupt mode register (INTM0)

This register sets the valid edge for INTP0 to INTP2 and TI0.

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

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

Remarks 1. INTP0 is also used for TI0/P00.

2. INTP3 is fixed to the falling edge.

Figure 16-5. Format of External Interrupt Mode Register

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

ES11	ES10	INTP0/TI0 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

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

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

★ **Caution When using the INTP0/TI0/P00 pin as a timer input pin (TI0), stop the operation of the 16-bit timer by clearing bits 1 to 3 (TMC01 to TMC03) of the 16-bit timer mode control register (TMC0) to 0, 0, 0, before setting the valid edge of TI0. When using the INTP0/TI0/P00 pin as an external interrupt input pin (INTP0), the valid edge of INTP0 may be set while the 16-bit timer is operating.**

(5) Sampling clock select register (SCS)

This register is used to set the clock used to sample the valid edge input to INTP0. When remote controlled data reception is carried out using INTP0, digital noise is eliminated using the sampling clock.

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

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

Figure 16-6. Format of Sampling Clock Select Register

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

SCS1	SCS0	INTP0 sampling clock selection
0	0	$f_X/2^{N+1}$
0	1	Setting prohibited
1	0	$f_X/2^6$ (78.1 kHz)
1	1	$f_X/2^7$ (39.1 kHz)

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

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

2. f_X: Main system clock oscillation frequency

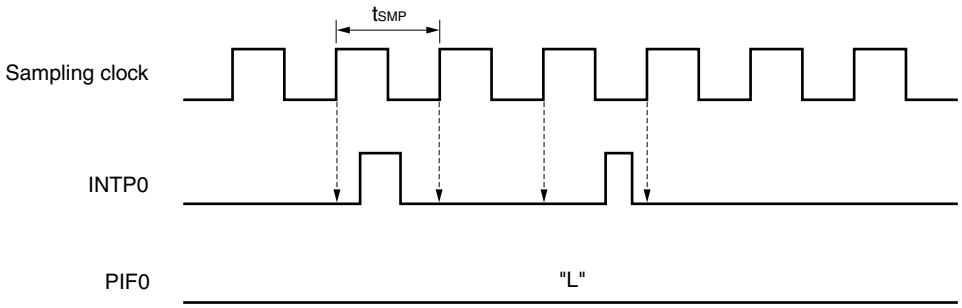
3. Figures in parentheses apply to operation with f_X = 5.0 MHz.

The noise eliminator sets the interrupt request flag (PIF0) to 1 if the input level of the sampled INTP0 is active twice in succession.

Figure 16-7 shows the noise eliminator I/O timing.

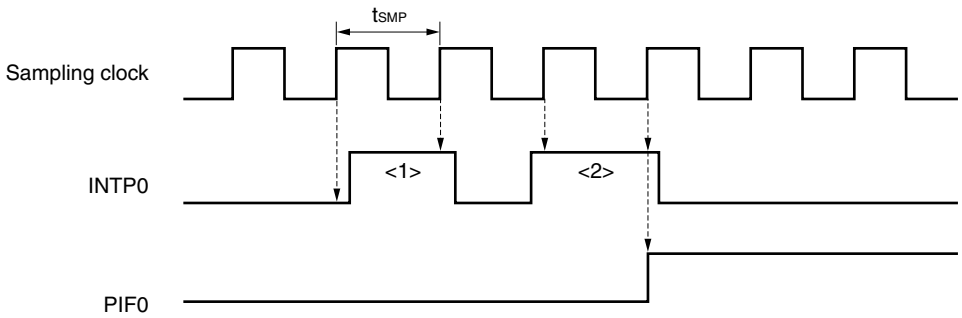
Figure 16-7. Noise Eliminator I/O Timing (When Rising Edge Is Detected)

(a) When input is less than the sampling cycle (t_{SMP})



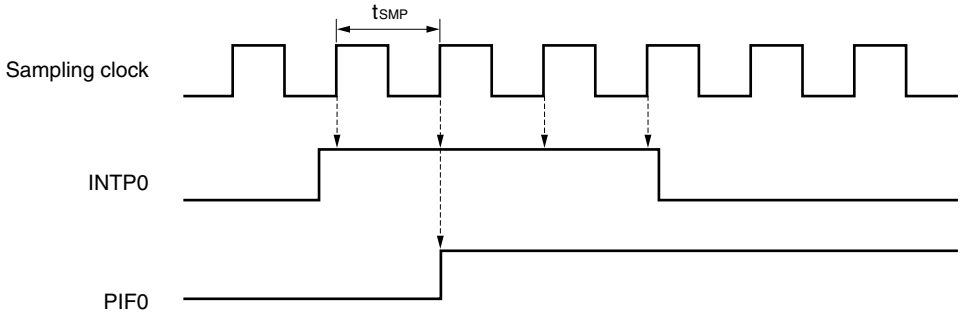
The PIF0 output remains low because the level of INTP0 is not high when it is sampled.

(b) When input is equal to or twice the sampling cycle (t_{SMP})



The PIF0 flag is set to 1 because the sampled INTP0 level is high twice in succession in <2>.

(c) When input is twice or more than the sampling cycle (t_{SMP})



The PIF0 flag is set to 1 when INTP0 goes high twice in succession.

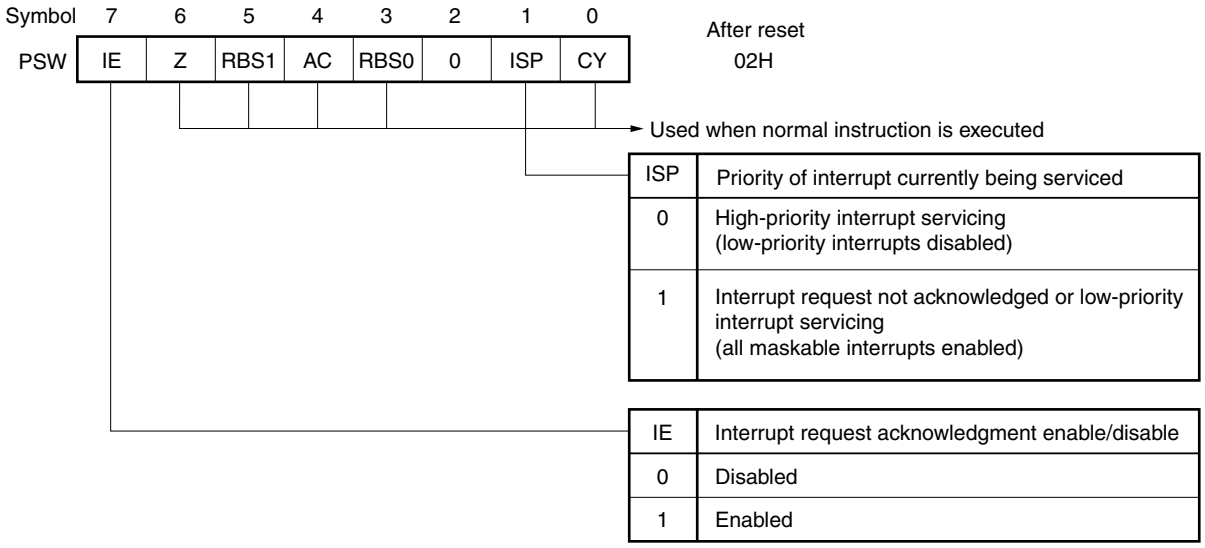
(6) Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for interrupt requests. The IE flag used to set maskable interrupt enable/disable and the ISP flag used to control multiple interrupt servicing are mapped to the PSW.

Besides 8-bit unit read/write, this register can carry out operations using bit manipulation and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged or when the BRK instruction is executed, the contents of the PSW are automatically saved into the stack, and the IE flag is reset to 0. If a maskable interrupt request is acknowledged, the contents of the priority specification flag of the acknowledged interrupt are transferred to the ISP flag. The contents of the PSW are also saved to the stack by the PUSH PSW instruction. They are reset from the stack with the RETI, RETB, and POP PSW instructions.

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

Figure 16-8. Format of Program Status Word



ISP	Priority of interrupt currently being serviced
0	High-priority interrupt servicing (low-priority interrupts disabled)
1	Interrupt request not acknowledged or low-priority interrupt servicing (all maskable interrupts enabled)

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

16.4 Interrupt Servicing Operations

16.4.1 Non-maskable interrupt request acknowledgment operation

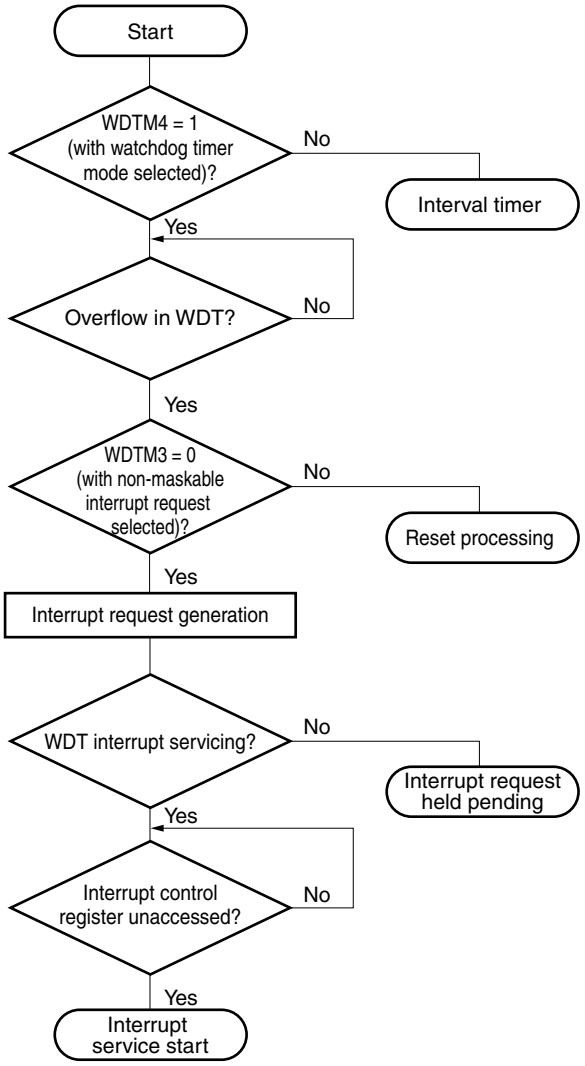
A non-maskable interrupt request is unconditionally acknowledged even if in an interrupt request acknowledgment disabled state. It does not undergo interrupt priority control and has highest priority over all other interrupts.

If a non-maskable interrupt request is acknowledged, the contents of program status word (PSW) and program counter (PC) are saved in the stacks in that order. Then, the IE and ISP flags are reset to 0, and the vector table contents are loaded into the PC and branched.

A new non-maskable interrupt request generated during execution of a non-maskable interrupt servicing program is acknowledged after execution of the current 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.

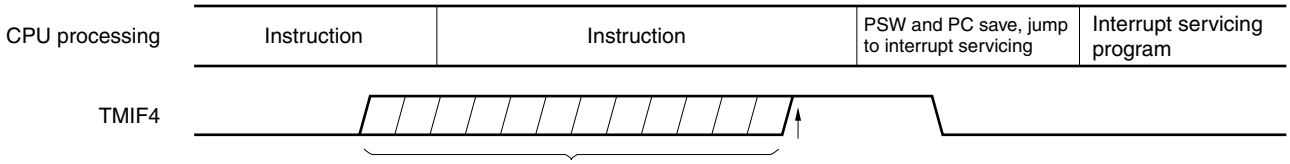
Figure 16-9 shows the flowchart illustrating generation and acknowledgment of a non-maskable interrupt request. Figure 16-10 shows the timing of acknowledging a non-maskable interrupt request. Figure 16-11 illustrates how nested non-maskable interrupt requests are acknowledged.

Figure 16-9. Non-Maskable Interrupt Request Acknowledgment Flowchart



WDTM: Watchdog timer mode register
WDT: Watchdog timer

Figure 16-10. Non-Maskable Interrupt Request Acknowledgment Timing

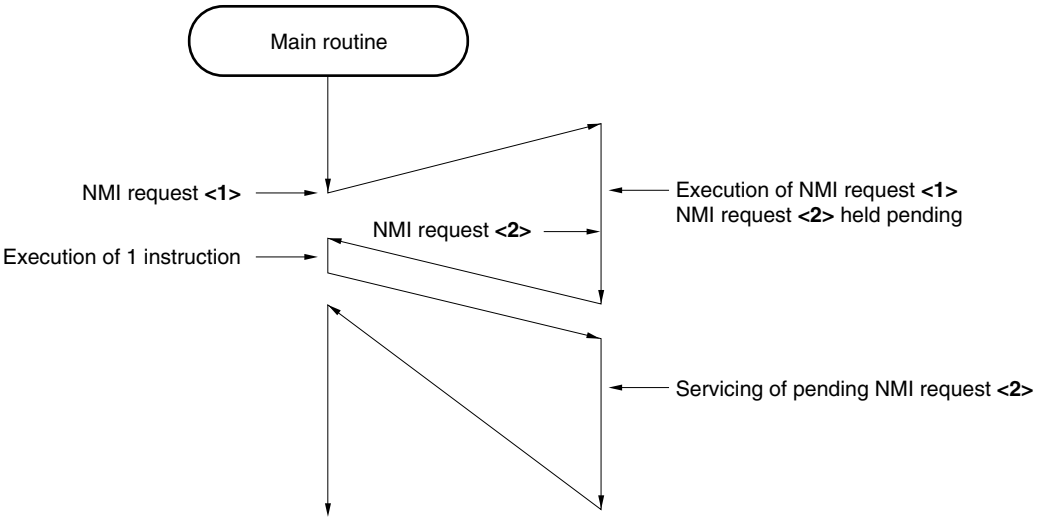


Interrupt request generated during this interval is acknowledged at ↑.

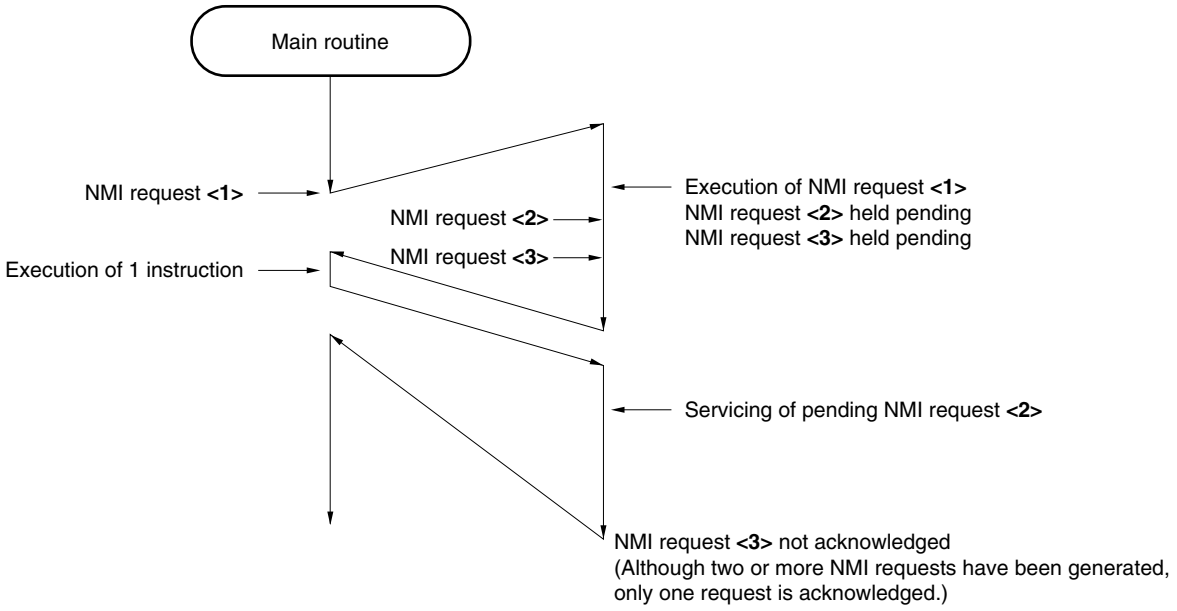
TMIF4: Watchdog timer interrupt request flag

Figure 16-11. Non-Maskable Interrupt Request Acknowledgment 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



16.4.2 Maskable interrupt request acknowledgment operation

A maskable interrupt request becomes acknowledgeable when an interrupt request flag is set to 1 and the mask (MK) flag of the interrupt request is cleared to 0. A vectored interrupt request is acknowledged in the interrupt enabled state (with IE flag set to 1). However, a low-priority interrupt is not acknowledged during high-priority interrupt request servicing (with ISP flag reset to 0).

Table 16-3 shows the time required until interrupt servicing is executed after a maskable interrupt request has been generated.

For the interrupt request acknowledgment timing, refer to Figures 16-13 and 16-14.

Table 16-3. Times from Maskable Interrupt Request Generation to Interrupt Servicing

	Minimum Time	Maximum Time[Note]
When ××PR = 0	7 clock cycles	32 clock cycles
When ××PR = 1	8 clock cycles	33 clock cycles

Note If an interrupt request is generated just before a divide instruction, the wait time is maximized.

Remark 1 clock cycle = $1/f_{CPU}$ (f_{CPU}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request specified as higher priority by the priority specification flag is acknowledged first. If the same priority is specified by the priority specification flag, the interrupt with the highest default priority is acknowledged first.

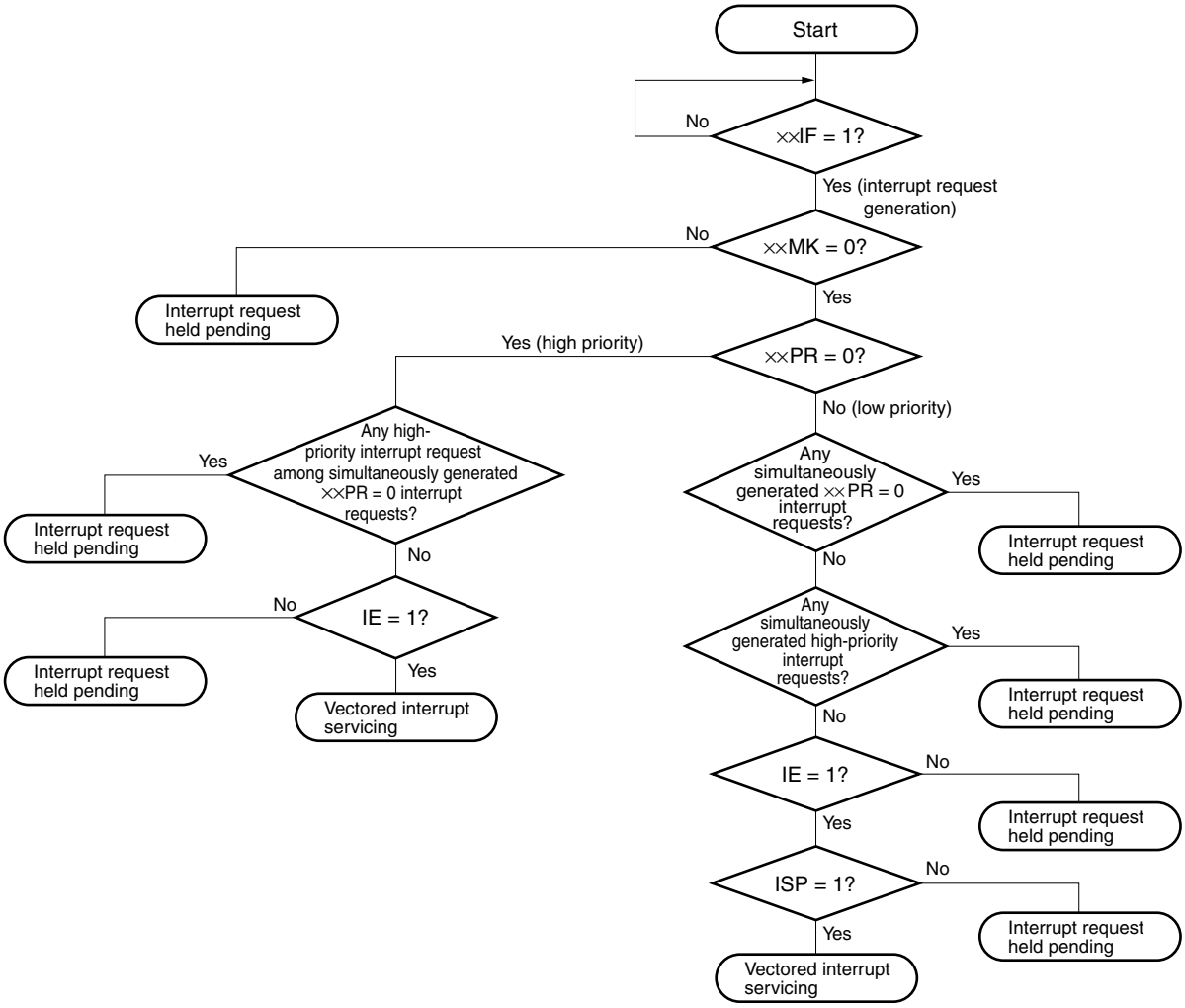
Any pending interrupt requests are acknowledged when they become acknowledgeable.

Figure 16-12 shows interrupt request acknowledgment algorithms.

If a maskable interrupt request is acknowledged, the contents of the program status word (PSW) and program counter (PC) are saved in the stacks in that order. Then, the IE flag is reset to 0, and the acknowledged interrupt request priority specification flag contents are transferred to the ISP flag. Further, the vector table data determined for each interrupt request is loaded into the PC and branched.

Return from the interrupt is possible with the RETI instruction.

Figure 16-12. Interrupt Request Acknowledge Processing Algorithm



××IF: Interrupt request flag

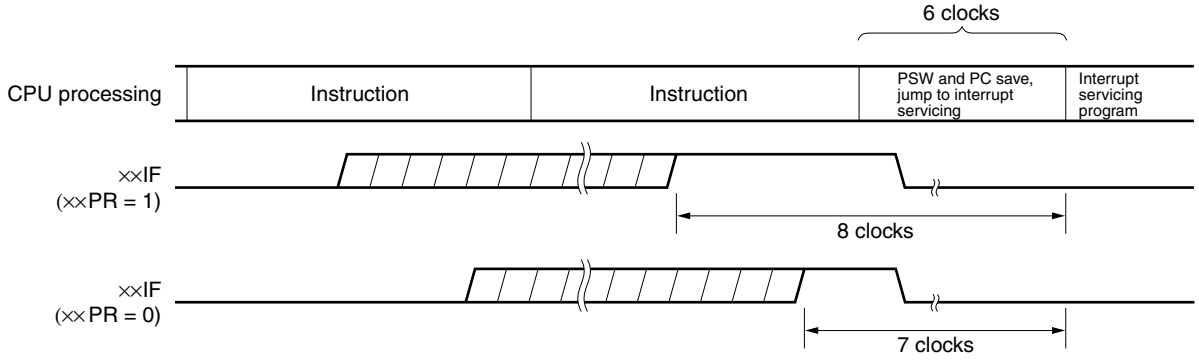
××MK: Interrupt mask flag

××PR: Priority specification flag

IE: Flag controlling acknowledgment of maskable interrupt request (1 = Enabled, 0 = Disabled)

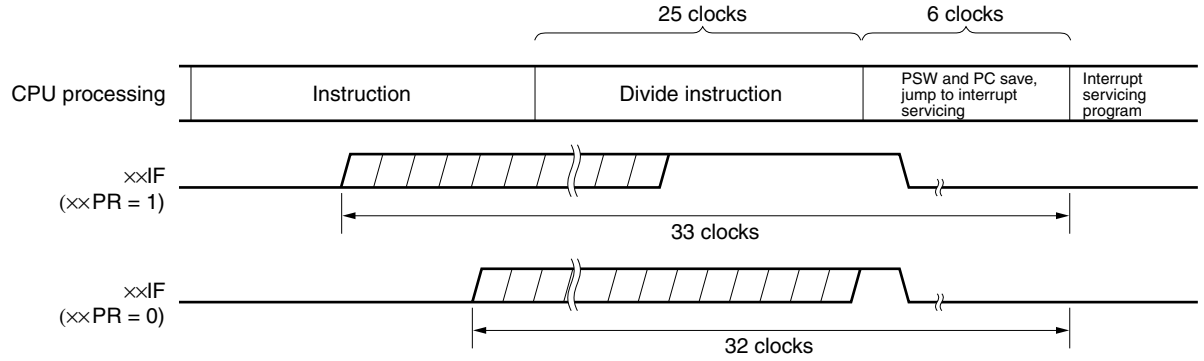
ISP: Flag indicating priority of interrupt currently serviced (0 = Interrupt with high priority is serviced, 1 = No interrupt request is acknowledged, or interrupt with low priority is serviced)

Figure 16-13. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock cycle = 1/f_{CPU} (f_{CPU}: CPU clock)

Figure 16-14. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock cycle = 1/f_{CPU} (f_{CPU}: CPU clock)

16.4.3 Software interrupt request acknowledgment operation

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents of the program status word (PSW) and program counter (PC) are saved in the stacks in that order, the IE flag is reset to 0 and the contents of the vector tables (003EH and 003FH) are loaded into the 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.

16.4.4 Multiple interrupt servicing

Multiple interrupt servicing occurs when an interrupt request is acknowledged during execution of another interrupt.

Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1) (except non-maskable interrupts). Also, when an interrupt request is received, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set (to 1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgment.

Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing.

In the interrupt enabled state, if an interrupt request with a priority equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing.

Interrupt requests that are not enabled because the interrupt disabled state is set or they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of one main processing instruction.

Multiple interrupt servicing is not possible during non-maskable interrupt servicing.

Table 16-4 shows interrupt requests enabled for multiple interrupt servicing, and Figure 16-15 shows multiple interrupt servicing examples.

Table 16-4. Interrupt Request Enabled for Multiple Interrupt During Interrupt Servicing

Multiple Interrupt Servicing Request / Interrupt Servicing		Non-Maskable Interrupt Request	Maskable Interrupt Request				Software Interrupt Request
			××PR = 0		××PR = 1		
			IE = 1	IE = 0	IE = 1	IE = 0	
Non-maskable interrupt		D	D	D	D	D	E
Maskable interrupt	ISP = 0	E	E	D	D	D	E
	ISP = 1	E	E	D	E	D	E
Software interrupt		E	E	D	E	D	E

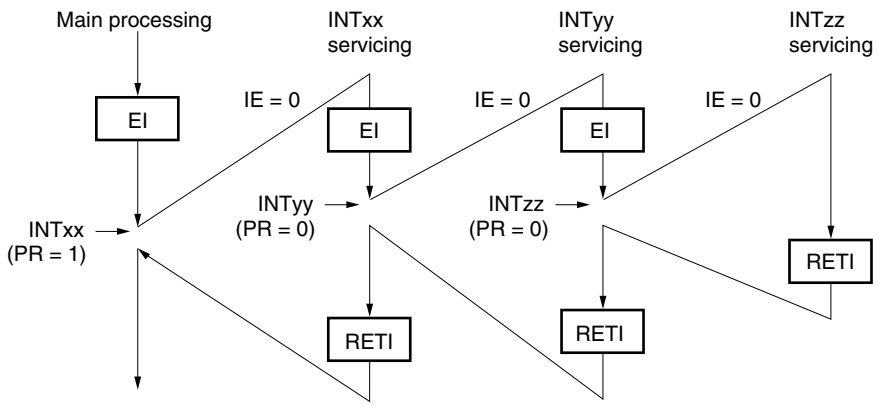
Remarks 1. E: Multiple interrupt servicing enabled
D: Multiple interrupt servicing disabled

2. ISP and IE are flags contained in the PSW.
ISP = 0: An interrupt with higher priority is being serviced
ISP = 1: An interrupt request is not acknowledged or an interrupt with lower priority is being serviced
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled

3. ××PR is a flag contained in PR0L and PR0H.
××PR = 0: Higher priority level
××PR = 1: Lower priority level

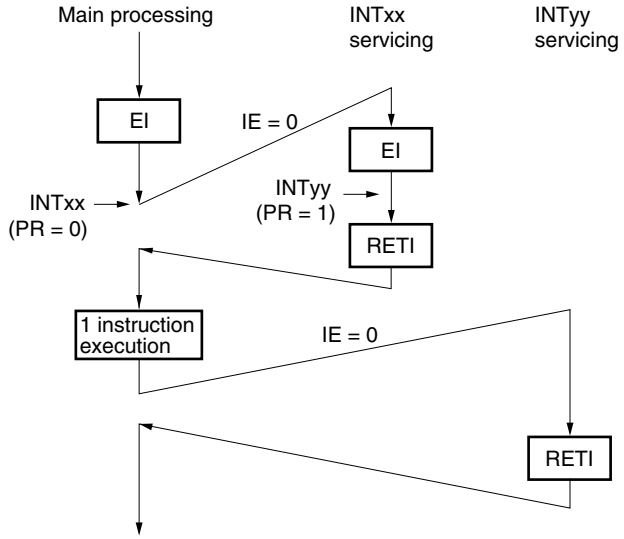
Figure 16-15. Multiple Interrupt Servicing Example (1/2)

Example 1. Two interrupts are generated



During interrupt INTxx servicing, two interrupt requests, INTyy and INTzz are acknowledged, and multiple interrupt servicing is generated. An EI instruction is issued before each interrupt request acknowledgment, and the interrupt request acknowledgment enabled state is set.

Example 2. Multiple interrupt servicing is not generated due to priority control

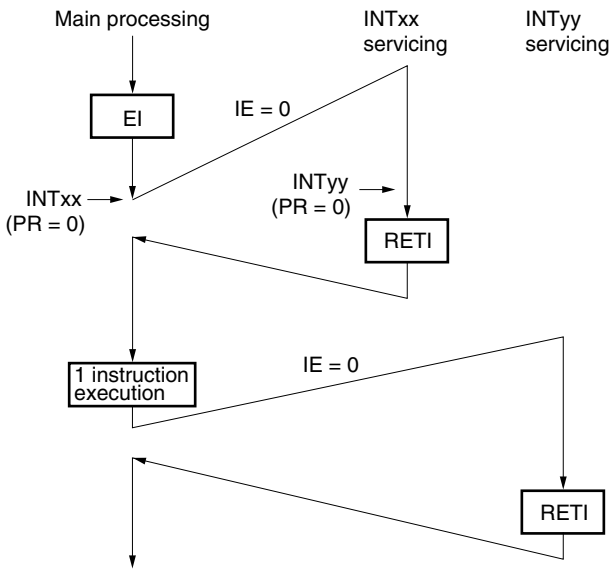


The interrupt request INTyy generated during interrupt INTxx servicing is not acknowledged because its interrupt priority is lower than that of INTxx, and multiple interrupt servicing is not generated. The INTyy request is held pending and acknowledged after execution of 1 instruction of the main processing.

PR = 0: Higher priority level
PR = 1: Lower priority level
IE = 0: Interrupt request acknowledgment disabled

Figure 16-15. Multiple Interrupt Servicing Example (2/2)

Example 3. Multiple interrupt servicing 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 multiple interrupt servicing is not generated. The INTyy request is held pending and acknowledged after execution of 1 instruction of the main processing.

PR = 0: Higher priority level
IE = 0: Interrupt request acknowledgment disabled

16.4.5 Interrupt request hold

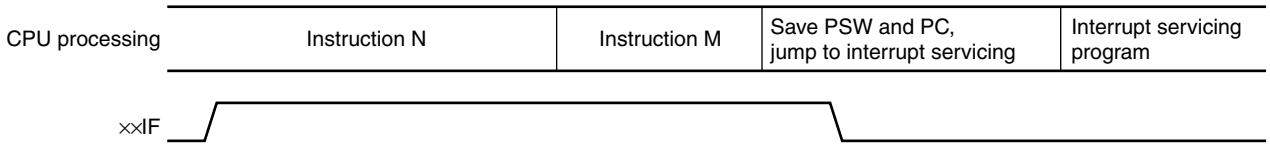
Some instructions hold an interrupt request, if any, pending until the completion of execution of the next instruction. These instructions (that hold an interrupt request pending) are listed below.

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW.bit, CY
- MOV1 CY, PSW.bit
- AND1 CY, PSW.bit
- OR1 CY, PSW.bit
- XOR1 CY, PSW.bit
- SET1 PSW.bit
- CLR1 PSW.bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW.bit, $addr16
- BF PSW.bit, $addr16
- BTCLR PSW.bit, $addr16
- EI
- DI
- Manipulation instructions for IF0L, IF0H, MK0L, MK0H, PR0L, PR0H and INTM0 registers

Caution The BRK instruction does not belong to the above group of instructions. However, the software interrupt that is started by execution of the BRK instruction clears the IE flag to 0. Therefore, even if a maskable interrupt request is generated, it is not acknowledged when the BRK instruction is executed. However, a non-maskable interrupt request is acknowledged.

The timing at which interrupt requests are held pending is shown in Figure 16-16.

Figure 16-16. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

3. The ××PR (priority level) values do not affect the operation of ××IF (interrupt request).

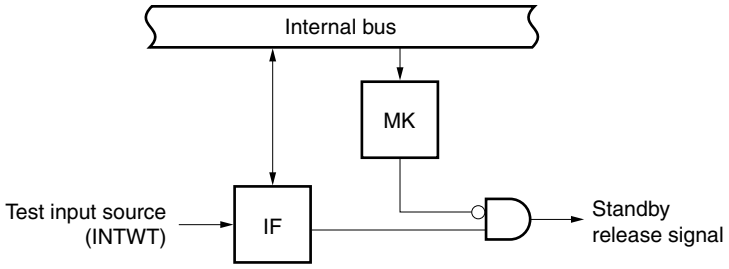
16.5 Test Functions

The internal test input flag (WTIF) is set to 1 and a standby release signal is generated when the watch timer overflows.

Unlike the interrupt function, this function does not perform vector processing.

The basic configuration is shown in Figure 16-17.

Figure 16-17. Basic Configuration of Test Function



IF: Test input flag
MK: Test mask flag

16.5.1 Test function control registers

The test function is controlled by the following two registers.

- Interrupt request flag register 0H (IF0H)
- Interrupt mask flag register 0H (MK0H)

The names of the test input flag and test mask flag corresponding to the test input signal name are as follows.

Test Input Signal Name	Test Input Flag	Test Mask Flag
INTWT	WTIF	WTMK

(1) Interrupt request flag register (IF0H)

This register indicates whether a watch timer overflow is detected or not.

IF0H is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears IF0H to 00H.

Figure 16-18. Format of Interrupt Request Flag Register 0H

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
IF0H	0	0	WTIF	KSIF	ADIF	TMIF2	TMIF1	TMIF0	FFE1H	00H	R/W

WTIF	Watch timer overflow detection flag
0	No detection
1	Detection

(2) Interrupt mask flag register (MK0H)

This register is used to set the standby mode enable/disable at the time the standby mode is released by the watch timer.

MK0H is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets MK0H to FFH.

Figure 16-19. Format of Interrupt Mask Flag Register 0H

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
MK0H	1	1	WTMK	KSMK	ADMK	TMMK2	TMMK1	TMMK0	FFE5H	FFH	R/W

WTMK	Standby mode control by watch timer
0	Standby mode release enabled
1	Standby mode release disabled

16.5.2 Test input signal acknowledgment operation

The internal test input signal (INTWT) is generated when the watch timer overflows. This signal sets the WTIF flag. At this time, the standby release signal is generated if it is not masked by the interrupt mask flag (WTMK). By checking the WTIF flag in a cycle shorter than the overflow cycle of the watch timer, a watch function can be realized.

CHAPTER 17 STANDBY FUNCTION

17.1 Standby Function and Configuration

17.1.1 Standby function

The standby function is used to decrease the power consumption of the system. The following two modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock stops. The system clock oscillator continues oscillation. In this mode, the power consumption cannot be decreased as much as in the STOP mode, but the HALT mode is effective for restarting immediately upon interrupt request and carrying out intermittent operations like clock operations.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the main system clock oscillator stops and the whole system stops, so the CPU power consumption can be considerably decreased. Data memory low-voltage hold (down to V_{DD} = 2 V) is possible. Thus, the STOP mode is effective for holding data memory contents with ultra-low power consumption.

Because this mode can be released by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is necessary to secure the oscillation stabilization time after the STOP mode is released, select the HALT mode if it is necessary to start processing immediately upon interrupt request.

In either mode, all the contents of the registers, flags, and data memory just before standby mode is set are held. The I/O 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 operated with the main system clock and execute the STOP instruction.

3. To reduce the power consumption of the A/D converter, set bit 7 (CS) of the A/D converter mode register (ADM) to 0 to stop the A/D converter's operation before executing the HALT or STOP instruction.

17.1.2 Standby function control register

The wait time after the STOP mode is released by an interrupt request until the oscillation stabilizes is controlled by the oscillation stabilization time select register (OSTS).

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

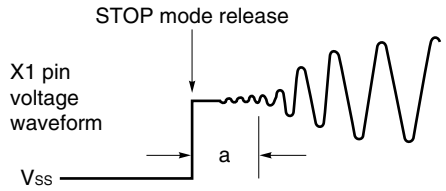
$\overline{\text{RESET}}$ input sets OSTS to 04H. Therefore, when the STOP mode is released by $\overline{\text{RESET}}$ input, the time until it is released is 2^{17}/fx.

Figure 17-1. Format of Oscillation Stabilization Time Select Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFAH	04H	R/W

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time after STOP mode is released
0	0	0	2^{12}/fx (819 μs)
0	0	1	2^{14}/fx (3.28 ms)
0	1	0	2^{15}/fx (6.55 ms)
0	1	1	2^{16}/fx (13.1 ms)
1	0	0	2^{17}/fx (26.2 ms)
Other than above			Setting prohibited

Caution The wait time after STOP mode release does not include the time from STOP mode release to clock oscillation start (see "a" below), regardless of whether the STOP mode is released by $\overline{\text{RESET}}$ input or by interrupt request generation.



Remarks 1. fx: Main system clock oscillation frequency

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

17.2 Standby Function Operations

17.2.1 HALT mode

(1) HALT mode set and operating status

The HALT mode is set by executing the HALT instruction. It can be set during main system clock or the subsystem clock operation.

The operating status in the HALT mode is described below.

Table 17-1. HALT Mode Operating Status

HALT Mode Setting / Item		When HALT Instruction Is Executed During Main System Clock Operation		When HALT Instruction Is Executed During Subsystem Clock Operation	
		Without Subsystem Clock[Note 1]	With Subsystem Clock[Note 2]	When Main System Clock Oscillation Continues	When Main System Clock Oscillation Stops
Clock generator		Both main system clock and subsystem clock can be oscillated. Clock supply to the CPU stops.			
CPU		Operation stopped			
Ports (output latch)		Status before HALT instruction execution is held.			
16-bit timer/event counter		Operation enabled			Operation stopped
8-bit timer/event counter					Operation enabled when TI1 and TI2 are selected for the count clock.
Watchdog timer					
A/D converter					Operation stopped
					Operation stopped
Watch timer		Operation enabled when $f_X/2^8$ is selected for the count clock.	Operation enabled		Operation enabled when f_{XT} is selected for the count clock.
★ Clock output		Operation enabled when $f_X/2^3$ to $f_X/2^8$ is selected for the output clock.	Operation enabled		Operation enabled when f_{XT} is selected for the output clock.
★ Buzzer output		Operation enabled			BUZ is low level.
VFD controller/driver		Operation disabled			
Serial interface	Other than automatic transmit/ receive function	Operation enabled			Operation enabled when external SCK is selected.
	Automatic transmit/ receive function	Operation stopped			
External interrupts	INTP0	Operation enabled when the clock for the peripheral hardware ($f_X/2^6$ or $f_X/2^7$) is selected as the sampling clock.			Operation stopped
	INTP1 to INTP3	Operation enabled			

Notes 1. Including the case where an external clock is not supplied as the subsystem clock

2. Including the case where an external clock is supplied as the subsystem clock

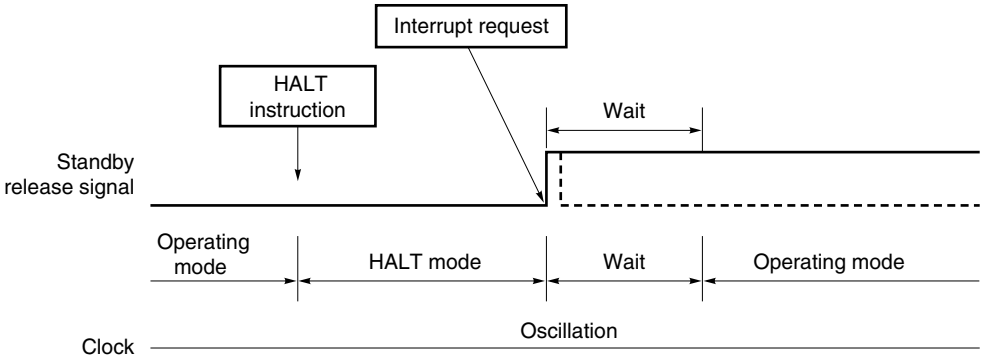
(2) HALT mode release

The HALT mode can be released by the following four sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt request acknowledgment is enabled, vectored interrupt servicing is carried out. If disabled, the instruction at the next address is executed.

Figure 17-2. HALT Mode Release 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 servicing is carried out: 8 to 9 clocks
- When vectored interrupt servicing is not carried out: 2 to 3 clocks

(b) Release by non-maskable interrupt request

When a non-maskable interrupt request is generated, the HALT mode is released and vectored interrupt servicing is carried out regardless of whether interrupt request acknowledgment is enabled or disabled.

However, a non-maskable interrupt request is not generated during subsystem clock operation.

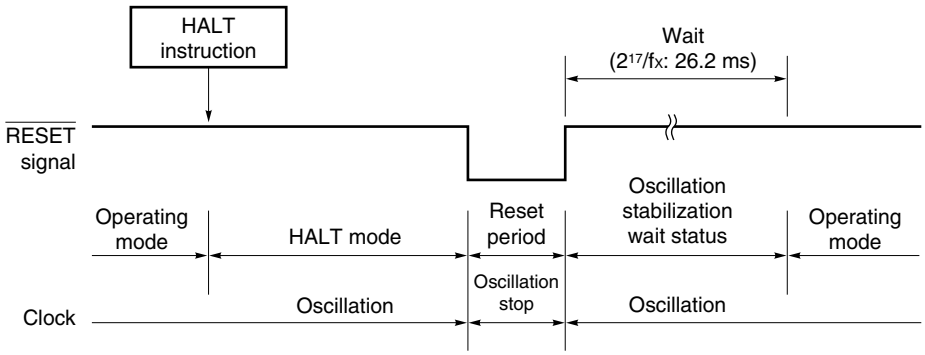
(c) Release by unmasked test input

When an unmasked test signal is input, the HALT mode is released and the instruction at the next address to the HALT instruction is executed.

(d) Release by $\overline{\text{RESET}}$ input

When a $\overline{\text{RESET}}$ signal is input, the HALT mode is released. As is the case with a normal reset operation, the program is executed after branch to the reset vector address.

Figure 17-3. HALT Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fx: Main system clock oscillation frequency

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

Table 17-2. Operation After HALT Mode Release

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing
	1	×	×	×	HALT mode hold
Non-maskable interrupt request	–	–	×	×	Interrupt servicing
Test input	0	–	×	×	Next address instruction execution
	1	–	×	×	HALT mode hold
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

×: don't care

17.2.2 STOP mode

(1) STOP mode set and operating status

The STOP mode is set by executing the STOP instruction. It can be set only during main system clock operation.

Cautions 1. When the STOP mode is set, the X2 pin is internally connected to V_{DD} via a pull-up resistor to suppress the leakage 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 released if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction. After the wait set using the oscillation stabilization time select register (OSTS), the operating mode is set.

The operating status in the STOP mode is described below.

Table 17-3. STOP Mode Operating Status

	STOP Mode Setting / Item		With Subsystem Clock	Without Subsystem Clock
	Clock generator		Only main system clock stops oscillation.	
	CPU		Operation stopped	
	Output ports (output latches)		Status before STOP instruction execution is held.	
	16-bit timer/event counter		Operation stopped	
	8-bit timer/event counter		Operation enabled only when TI1 and TI2 are selected for the count clock.	
	Watchdog timer		Operation stopped	
	A/D converter			
	Watch timer		Operation enabled only when f_{XT} is selected for the count clock.	Operation stopped
★	Clock output		Operation enabled when f_{XT} is selected for the output clock.	PCL is low level.
★	Buzzer output		BUZ is low level.	
	VFD controller/driver		Operation disabled	
	Serial interface	Other than automatic transmit/receive function	Operation enabled only when external input clock is selected as serial clock.	
		Automatic transmit/ receive function	Operation stopped	
	External interrupts	INTP0	Operation disabled	
		INTP1 to INTP3	Operation enabled	

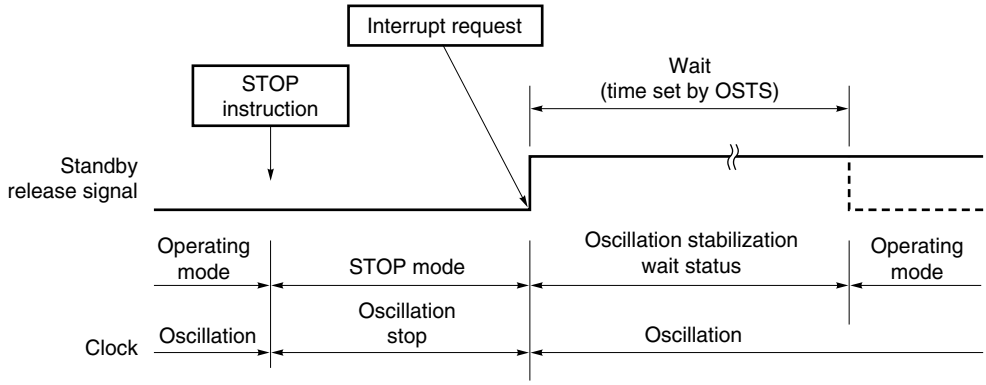
(2) STOP mode release

The STOP mode can be released by the following three sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the STOP mode is released. If interrupt request acknowledgment is enabled after the lapse of oscillation stabilization time, vectored interrupt servicing is carried out. If interrupt request acknowledgment is disabled, the instruction at the next address is executed.

Figure 17-4. STOP Mode Release by Interrupt Request Generation



Remark The broken line indicates the case when the interrupt request which has released the standby status is acknowledged.

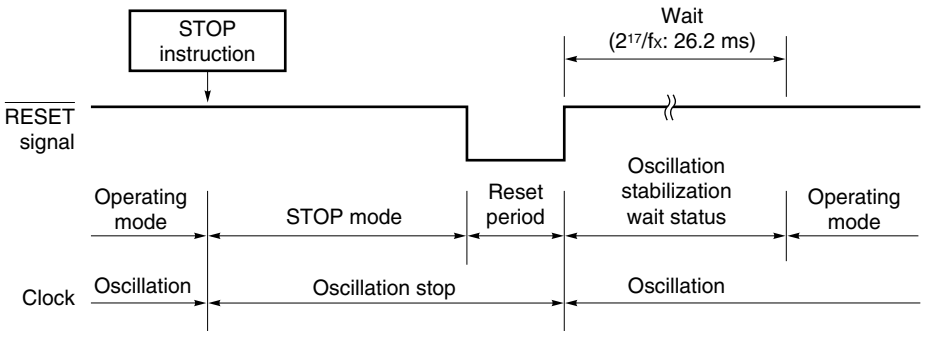
(b) Release by unmasked test input

When an unmasked test signal is input, the STOP mode is released. After the lapse of oscillation stabilization time, the instruction at the next address to the STOP instruction is executed.

(c) Release by $\overline{\text{RESET}}$ input

When a $\overline{\text{RESET}}$ signal is input, the STOP mode is released. After the lapse of oscillation stabilization time, a reset operation is carried out.

Figure 17-5. STOP Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fx: Main system clock oscillation frequency

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

Table 17-4. Operation After STOP Mode Release

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing
	1	×	×	×	STOP mode hold
Test input	0	–	×	×	Next address instruction execution
	1	–	×	×	STOP mode hold
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

×: don't care

CHAPTER 18 RESET FUNCTION

18.1 Reset Function

The following two operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop time detection

External reset and internal reset have no functional differences. In both cases, program execution starts at addresses 0000H and 0001H by $\overline{\text{RESET}}$ input.

When a low level is input to the $\overline{\text{RESET}}$ pin or the watchdog timer overflows, a reset is applied and each hardware is set to the status as shown in Table 18-1. Each pin is high impedance during reset input or during the oscillation stabilization time just after reset release.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is released and program execution starts after the lapse of the oscillation stabilization time (2^{17}/fx). The reset applied by watchdog timer overflow is automatically released after the reset and program execution starts after the lapse of the oscillation stabilization time (2^{17}/fx) (see **Figures 18-2** to **18-4**).

Cautions 1. For an external reset, input a low level to the $\overline{\text{RESET}}$ pin for 10 μs or more.
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 pins become high impedance.

Figure 18-1. Block Diagram of Reset Function

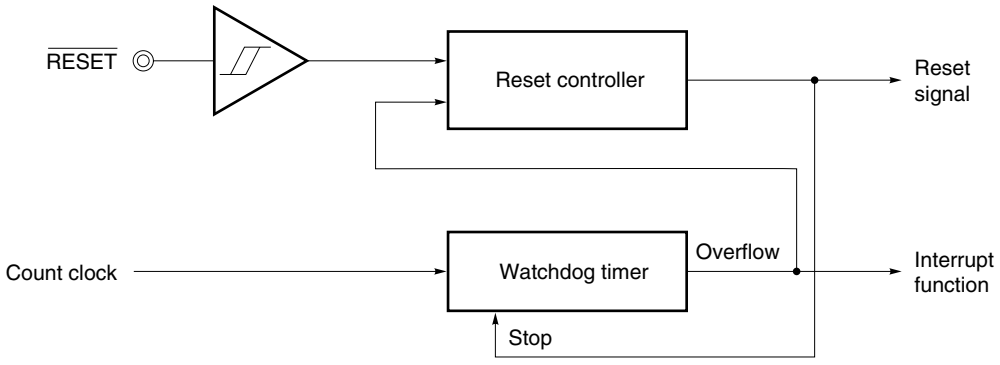


Figure 18-2. Timing of Reset by $\overline{\text{RESET}}$ Input

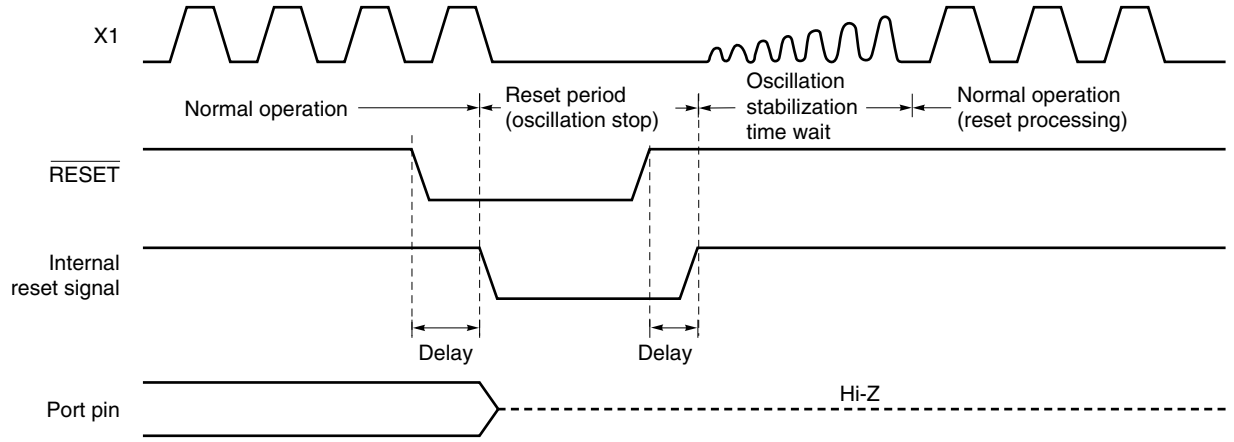


Figure 18-3. Timing of Reset due to Watchdog Timer Overflow

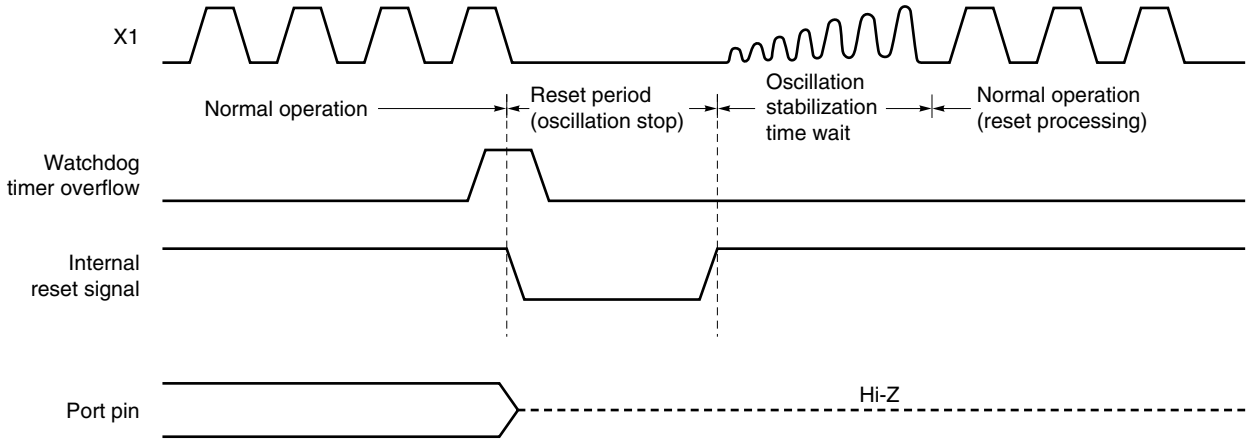


Figure 18-4. Timing of Reset by $\overline{\text{RESET}}$ Input in STOP Mode

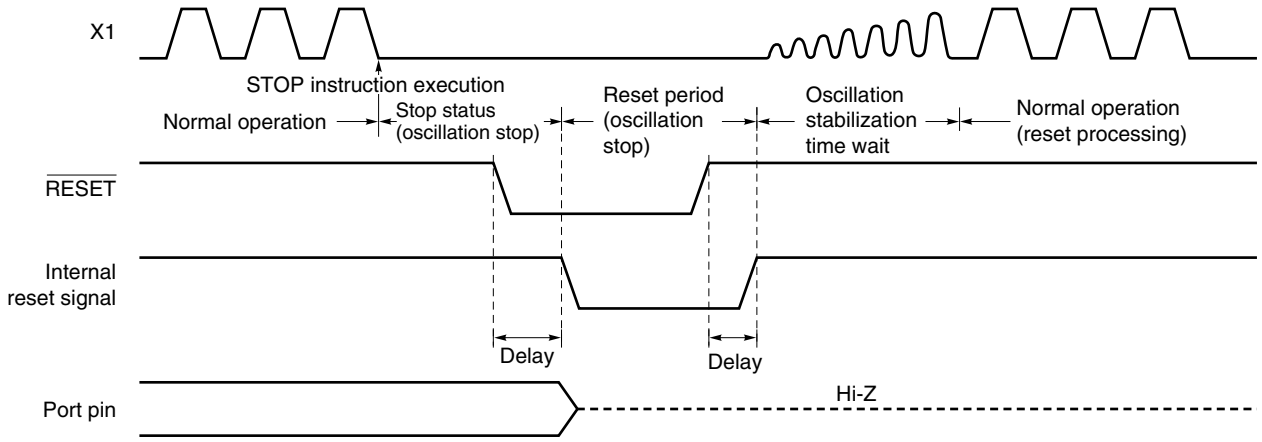


Table 18-1. Hardware Status After Reset (1/2)

Hardware		Status After Reset
Program counter (PC)[Note 1]		The contents of reset vector tables (0000H and 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Ports (output latches)	Ports 0 to 3, 7 to 12 (P0 to P3, P7 to P12)	00H
Port mode registers	(PM0, PM7)	1FH
	(PM1, PM2, PM3, PM10, PM11, PM12)	FFH
Pull-up resistor option register (PUO)		00H
Processor clock control register (PCC)		04H
Internal memory size switching register (IMS)		**Note 3**
Internal expansion RAM size switching register (IXS)		**Note 3**
Oscillation stabilization time select register (OSTS)		04H
16-bit timer/event counter	Timer register (TM0)	00H
	Compare register (CR00)	Undefined
	Capture register (CR01)	Undefined
	Clock select register (TCL0)	00H
	Mode control register (TMC0)	00H
	Output control register (TOC0)	00H
8-bit timer/event counter	Timer registers (TM1, TM2)	00H
	Compare registers (CR10, CR20)	Undefined
	Clock select register (TCL1)	00H
	Mode control registers (TMC1, TMC2)	00H
	Output control register (TOC1)	00H

Notes 1. During reset input or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. If the reset is applied in the standby mode, the status before reset will be held after reset.

3. The after-reset values of the internal memory size switching register (IMS) and internal expansion RAM size switching register (IXS) depend on the product.

	μPD780204	μPD780204A	μPD780205	μPD780205A	μPD780206	μPD780208	μPD78P0208
IMS	C8H	CFH	CAH	CFH	CCH	CFH	CFH
IXS	None				0AH		

Table 18-1. Hardware Status After Reset (2/2)

Hardware		Status After Reset
Watch timer	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)	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 specification register (ADTI)	00H
	Interrupt timing specification register (SINT)	00H
A/D converter	Mode register (ADM)	01H
	Conversion result register (ADCR)	Undefined
	Input select register (ADIS)	00H
VFD controller/driver	Display mode register 0 (DSPM0)	00H
	Display mode register 1 (DSPM1)	00H
	Display mode register 2 (DSPM2)	00H
Interrupts	Request flag registers (IF0L, IF0H)	00H
	Mask flag registers (MK0L, MK0H)	FFH
	Priority specification flag registers (PR0L, PR0H)	FFH
	External interrupt mode register (INTM0)	00H
	Sampling clock select register (SCS)	00H

CHAPTER 19 μPD78P0208

The μPD78P0208 is a product integrating a one-time programmable ROM (one-time PROM). Table 19-1 shows the differences between the μPD78P0208 and the mask ROM versions (μPD780204, 780204A, 780205, 780205A, 780206, and 780208).

Table 19-1. Differences Between μPD78P0208 and Mask ROM Versions

Item	μPD78P0208	Mask ROM Versions
Internal ROM configuration	One-time PROM	Mask ROM
Internal ROM capacity	60 KB	μPD780204: 32 KB μPD780204A: 32 KB μPD780205: 40 KB μPD780205A: 40 KB μPD780206: 48 KB μPD780208: 60 KB
Internal expansion RAM capacity	1024 bytes	μPD780204: None μPD780204A: None μPD780205: None μPD780205A: None μPD780206: 1024 bytes μPD780208: 1024 bytes
Change in capacity of internal ROM by means of internal memory size switching register (IMS)	Possible[Note 1]	Impossible
Internal expansion RAM size switching register (IXS)	Provided (Internal expansion RAM capacity can be changed using IXS[Note 2].)	μPD780204, 780204A, μPD780205, 780205A: Not provided μPD780206, 780208: Provided (However, internal expansion RAM capacity cannot be changed.)
IC pin	None	Provided
V_{PP} pin	Provided	None
P30/TO0 to P32/TO2, P33/TI1, P34/TI2, P35/PCL, P36/BUZ, P37	On-chip pull-down resistors are not provided.	On-chip pull-down resistors can be specified in 1-bit units by mask option.
P70 to P74	On-chip pull-up resistors are not provided.	On-chip pull-up resistors can be specified in 1-bit units by mask option.
FIP0 to FIP12	On-chip pull-down resistors are provided (connected to V_{LOAD}).	On-chip pull-down resistors can be specified in 1-bit units by mask option.
P80/FIP13 to P87/FIP20, P90/FIP21 to P97/FIP28, P100/FIP29 to P107/FIP36, P110/FIP37 to P117/FIP44, P120/FIP45 to P127/FIP52	On-chip pull-down resistors are not provided.	On-chip pull-down resistors can be specified in 1-bit units by mask option. Pull-down resistors can be specified to be connected to either V_{LOAD} or V_{SS} in 4-bit units from P80.
Electrical specifications	Refer to the separate data sheet.	

Notes 1. After $\overline{\text{RESET}}$ input, the internal PROM capacity is set to 60 KB.

2. After $\overline{\text{RESET}}$ input, the internal expansion RAM capacity is set to 1024 bytes.

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 commercial samples (not engineering samples) of the mask ROM version.

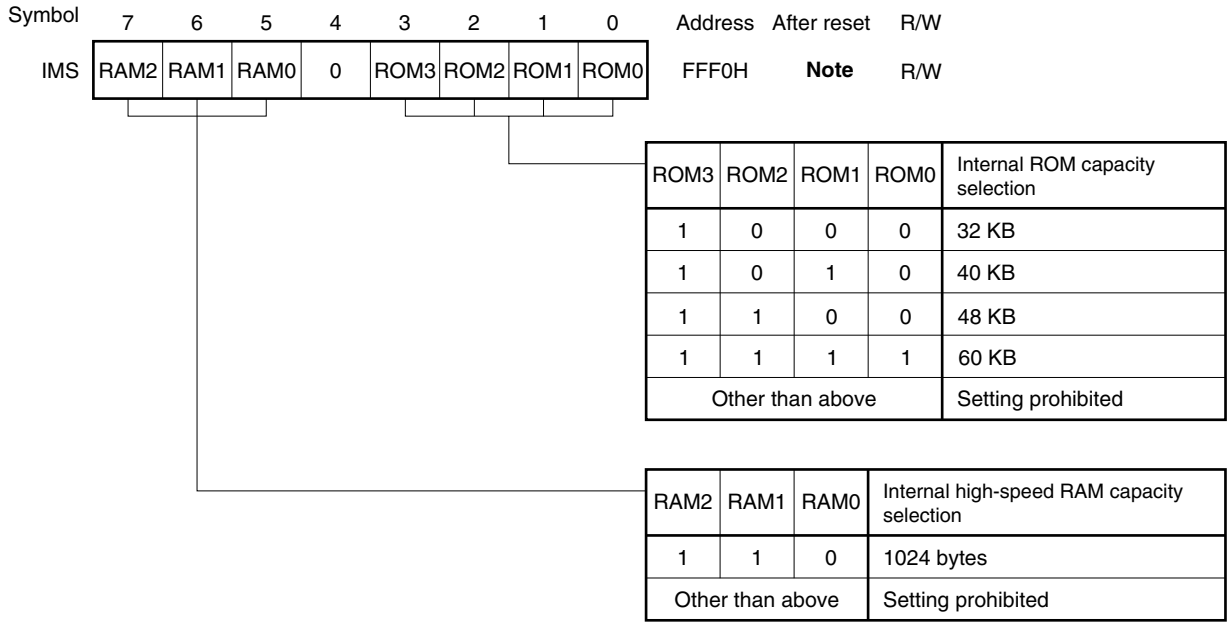
19.1 Internal Memory Size Switching Register

The internal memory capacity of the μPD78P0208 can be selected by using the internal memory size switching register (IMS). The same memory map as that of the mask ROM version with a different internal memory capacity is possible by setting IMS.

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

$\overline{\text{RESET}}$ input sets IMS as shown in Table 19-2.

Figure 19-1. Format of Internal Memory Size Switching Register (IMS)



ROM3	ROM2	ROM1	ROM0	Internal ROM capacity selection
1	0	0	0	32 KB
1	0	1	0	40 KB
1	1	0	0	48 KB
1	1	1	1	60 KB
Other than above				Setting prohibited

RAM2	RAM1	RAM0	Internal high-speed RAM capacity selection
1	1	0	1024 bytes
Other than above			Setting prohibited

Note The value of the internal memory size switching register after reset differs depending on the product (see **Table 19-2**).

Table 19-2 lists the IMS setting values for a memory map equivalent to the mask ROM versions.

★ **Table 19-2. Internal Memory Size Switching Register Setting Values**

Target Product	IMS Value After Reset	IMS Setting Value
μPD780204	C8H	
μPD780204A	CFH	C8H
μPD780205	CAH	
μPD780205A	CFH	CAH
μPD780206	CCH	
μPD780208	CFH	
μPD78P0208	CFH	

Caution When using the μPD780204, 780205, 780206, and 780208, do not set any value other than the above IMS Value After Reset to IMS.
When using the μPD780204A and 780205A, be sure to set the IMS Setting Value shown in Table 19-2 to IMS.

19.2 Internal Expansion RAM Size Switching Register

By setting the internal expansion RAM size swtiching register (IXS), the μPD78P0208 can have the same memory map as used in mask ROM versions that have a different internal expansion RAM capacity.

For the mask ROM versions, IXS does not need to be set.

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

$\overline{\text{RESET}}$ input sets IXS to 0AH.

Cautions 1. The internal expansion RAM size switching register (IXS) is only incorporated in the μPD780206, μPD780208, and μPD78P0208.

2. When using a mask ROM version μPD780204, μPD780204A, μPD780205, μPD780205A, μPD780206, μPD780208, do not set a value other than those listed in Table 19-3 to IXS.

Figure 19-2. Format of Internal Expansion RAM Size Switching Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
IXS	0	0	0	0	IX RAM3	IX RAM2	IX RAM1	IX RAM0	FFF4H	0AH	W

IX RAM3	IX RAM2	IX RAM1	IX RAM0	Internal expansion RAM capacity selection
1	0	1	0	1024 bytes
1	1	0	0	No internal expansion RAM (0 bytes)
Other than above				Setting prohibited

Table 19-3 lists the IXS setting values for a memory map equivalent to the mask ROM versions.

Table 19-3. Internal Expansion RAM Size Switching Register Setting Values

Target Mask ROM Version	IXS Setting Value
μPD780204, 780204A μPD780205, 780205A	0CH
μPD780206 μPD780208	0AH

IXS is not incorporated in the μPD780204, μPD780204A, μPD780205, and μPD780205A.

However, if a write instruction to IXS is executed in the μPD780204, μPD780204A, μPD780205, or μPD780205A, the operation is not affected.

19.3 PROM Programming

The μPD78P0208 incorporates a 60 KB PROM as program memory. When programming, the PROM programming mode is set by means of the V_{PP} pin and the $\overline{RESET}$ pin. For the connection of unused pins, refer to **1.5 Pin Configuration (Top View) (2) PROM programming mode**.

Caution Programs must be written in addresses 0000H to EFFFH (the last address EFFFH must be specified). Programs cannot be written by a PROM programmer that cannot specify the write address.

19.3.1 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 $\overline{RESET}$ pin, the μPD78P0208 is set to the PROM programming mode. This is one of the operating modes shown in Table 19-4 below according to the setting of the $\overline{CE}$, $\overline{OE}$, and $\overline{PGM}$ pins.

The PROM contents can be read by setting the read mode.

Table 19-4. PROM Programming Operating Modes

Pin / Operating Mode	$\overline{RESET}$	V_{PP}	V_{DD}	$\overline{CE}$	$\overline{OE}$	$\overline{PGM}$	D0 to D7
Page data latch	L	+12.5 V	+6.5 V	H	L	H	Data input
Page write				H	H	L	High impedance
Byte write				L	H	L	Data input
Program verify				L	L	H	Data output
Program inhibit				×	H	H	High impedance
				×	L	L	
Read		+5 V	+5 V	L	L	H	Data output
Output disabled				L	H	×	High impedance
Standby				H	×	×	High impedance

×: L or H

(1) Read mode

Read mode is set by setting $\overline{CE}$ to L and $\overline{OE}$ to L.

(2) Output disabled mode

If $\overline{OE}$ is set to H, data output becomes high impedance and the output disabled mode is set.

Therefore, if multiple μPD78P0208s are connected to the data bus, data can be read from any one device by controlling the $\overline{OE}$ pin.

(3) Standby mode

Setting $\overline{CE}$ to H sets the standby mode.

In this mode, data output becomes high impedance irrespective of the status of $\overline{OE}$.

(4) Page data latch mode

Setting $\overline{CE}$ to H, $\overline{PGM}$ to H, and $\overline{OE}$ to L at the start of the page write mode sets the page data latch mode. In this mode, 1-page 4-byte data is latched in the internal address/data latch circuit.

(5) Page write mode

After a 1-page 4-byte address and data are latched by the page data latch mode, a page write is executed by applying a 0.1 ms program pulse (active-low) to the $\overline{PGM}$ pin while $\overline{CE}$ = H and $\overline{OE}$ = H. After this, program verification can be performed by setting $\overline{CE}$ to L and $\overline{OE}$ to L.

If programming is not performed by one program pulse, repeated write and verify operations are executed X times (X ≤ 10).

(6) Byte write mode

A byte write is executed by applying a 0.1 ms program pulse (active-low) to the $\overline{PGM}$ pin while $\overline{CE}$ = L and $\overline{OE}$ = H. After this, program verification can be performed by setting $\overline{OE}$ to L.

If programming is not performed by one program pulse, repeated write and verify operations are executed X times (X ≤ 10).

(7) Program verify mode

Setting $\overline{CE}$ to L, $\overline{PGM}$ to H, and $\overline{OE}$ to L sets the program verify mode.

After writing is performed, this mode should be used to check whether the data was written correctly.

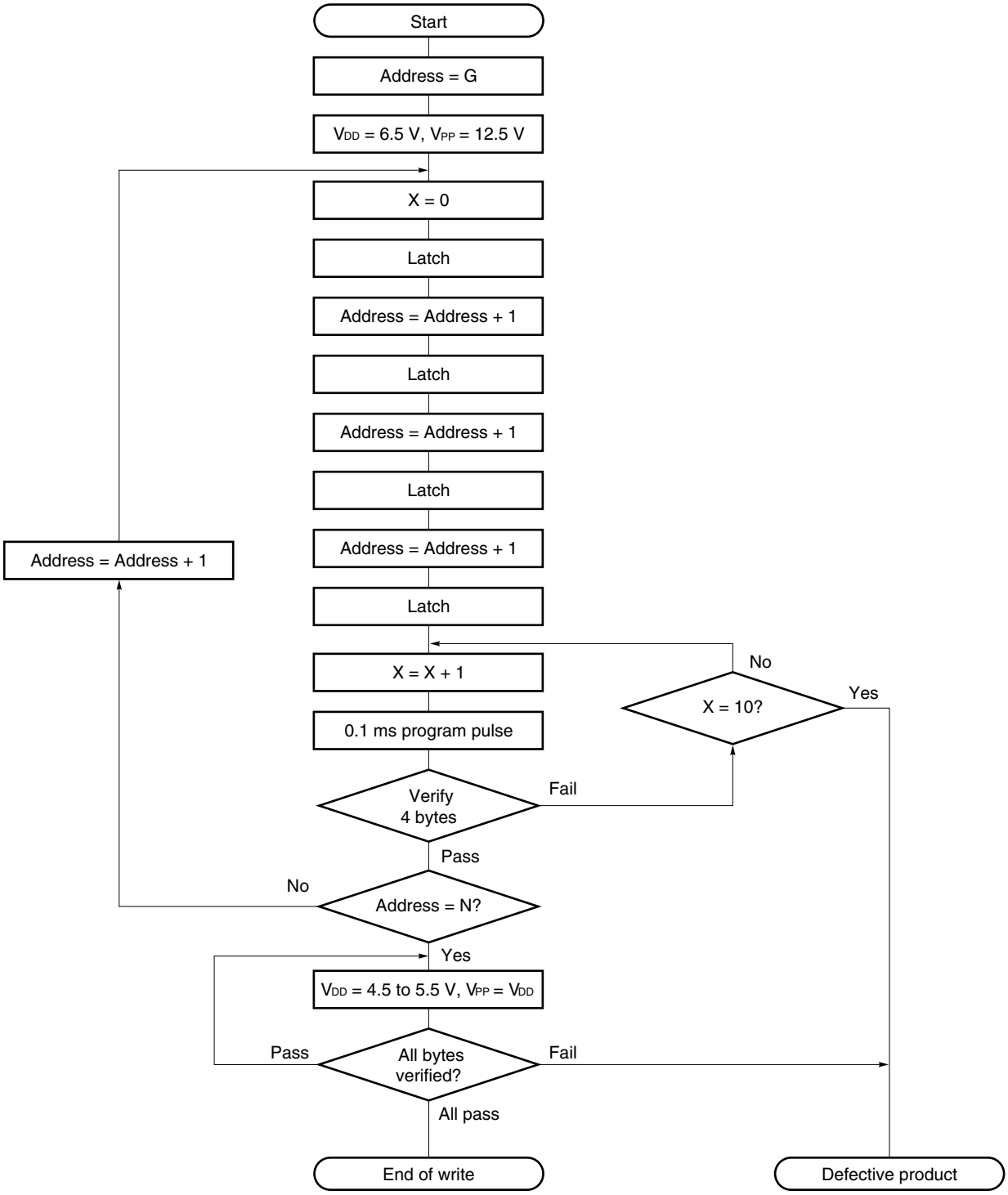
(8) Program inhibit mode

The program inhibit mode is used when the $\overline{OE}$, V_{PP}, and D0 to D7 pins of multiple μPD78P0208s are connected in parallel, and when you wish to write to one of these devices.

The page write mode or byte write mode described above is used to perform a write. At this time, the write is not performed on the device which has the $\overline{PGM}$ pin driven high.

19.3.2 PROM write procedure

Figure 19-3. Page Program Mode Flowchart



G = Start address

N = Last address of program

Figure 19-4. Page Program Mode Timing

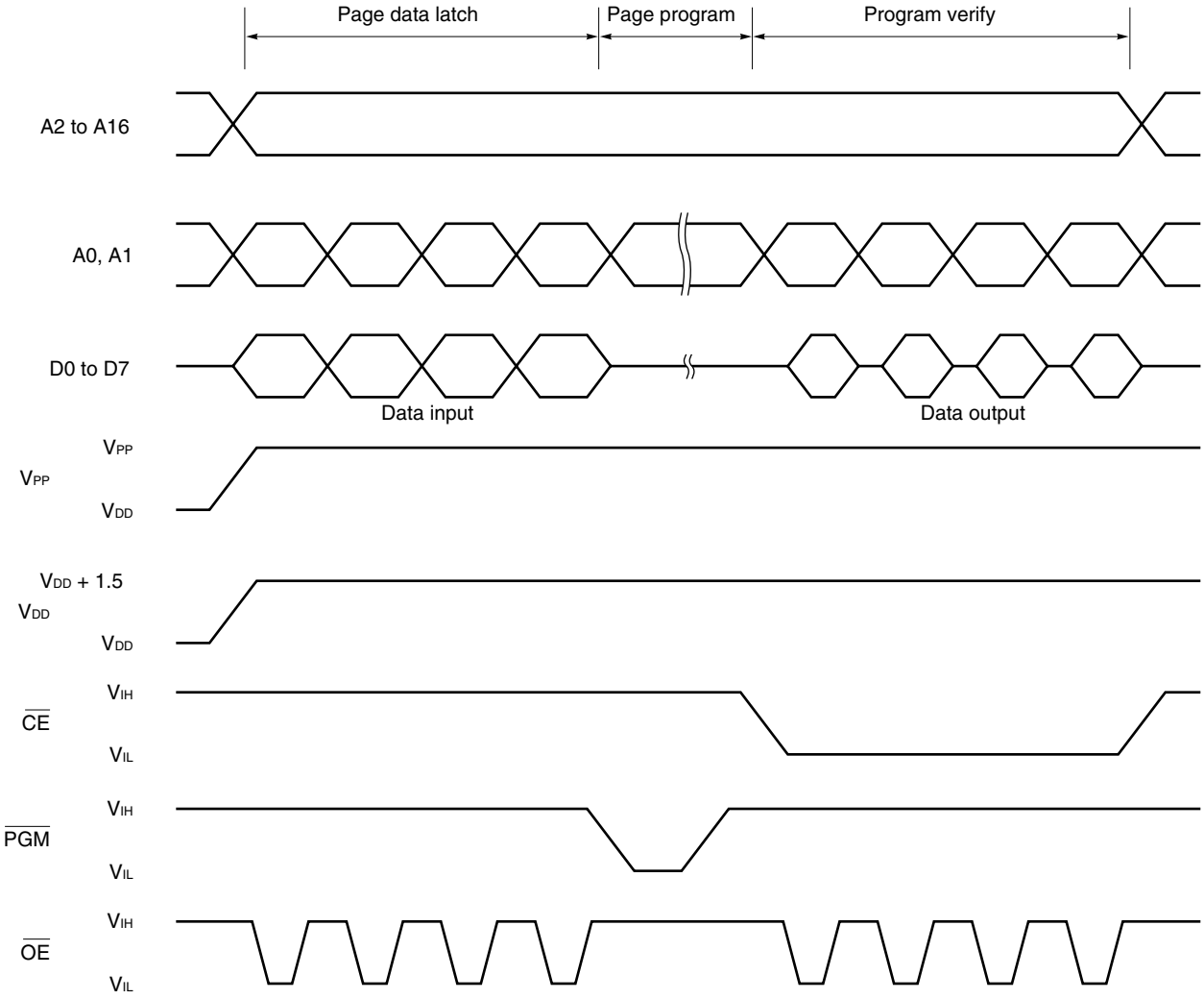
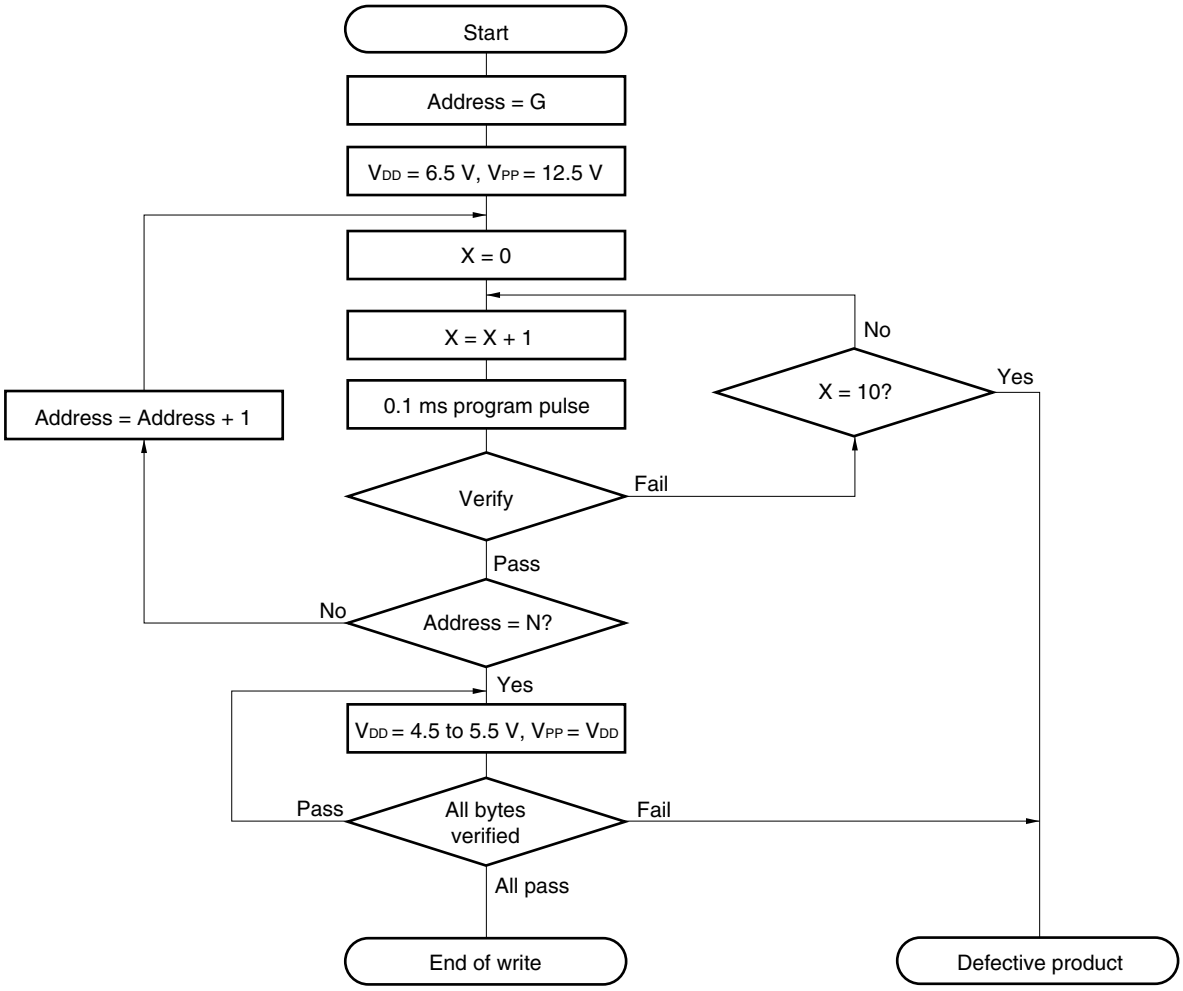


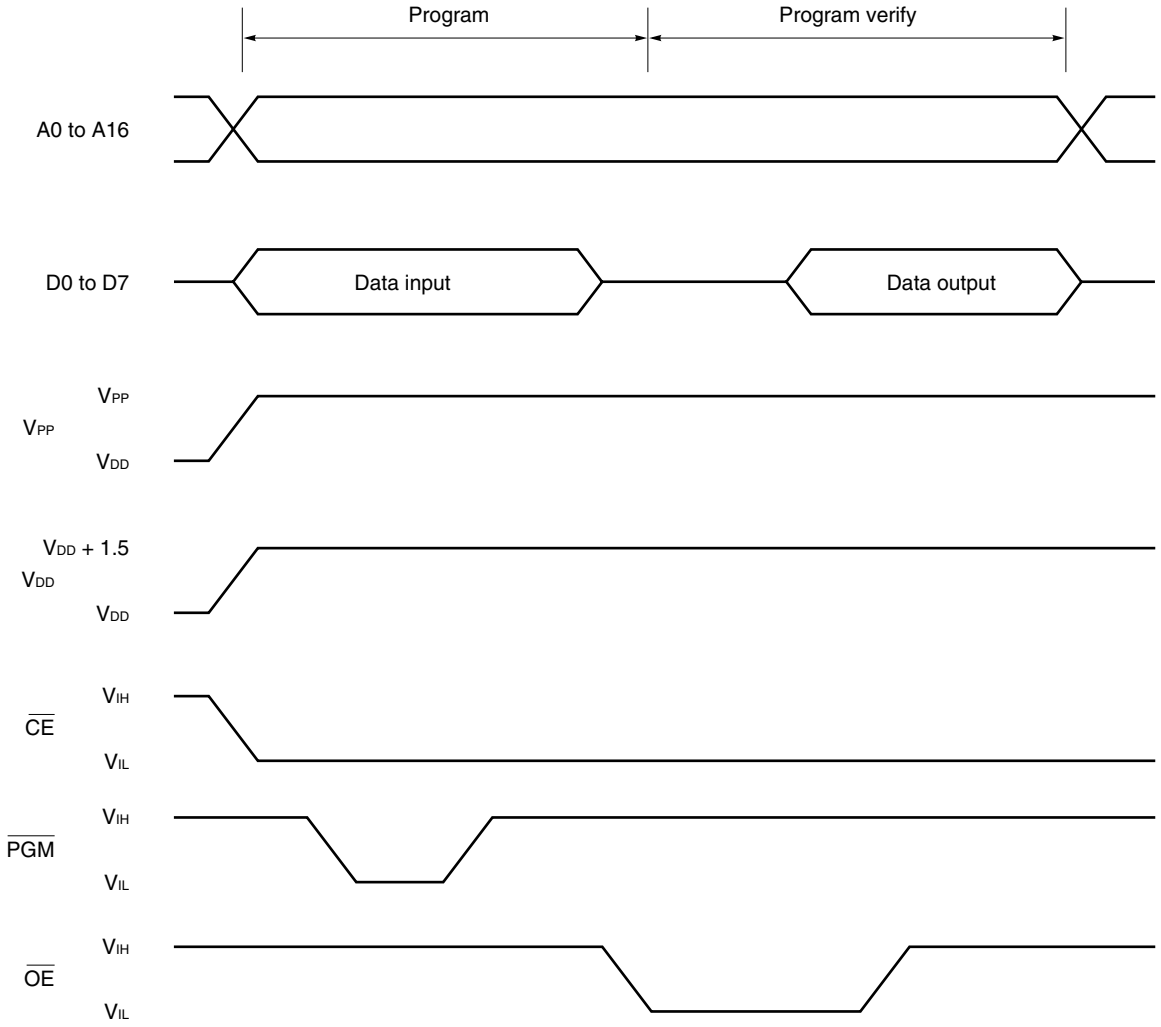
Figure 19-5. Byte Program Mode Flowchart



G = Start address

N = Last address of program

Figure 19-6. Byte Program Mode Timing



Cautions 1. Ensure that V_{DD} is applied before V_{PP} and removed after V_{PP}.

2. Ensure that V_{PP} does not exceed +13.5 V including overshoot.

3. Disconnecting the device while +12.5 V is being applied to V_{PP} may have an adverse affect on reliability.

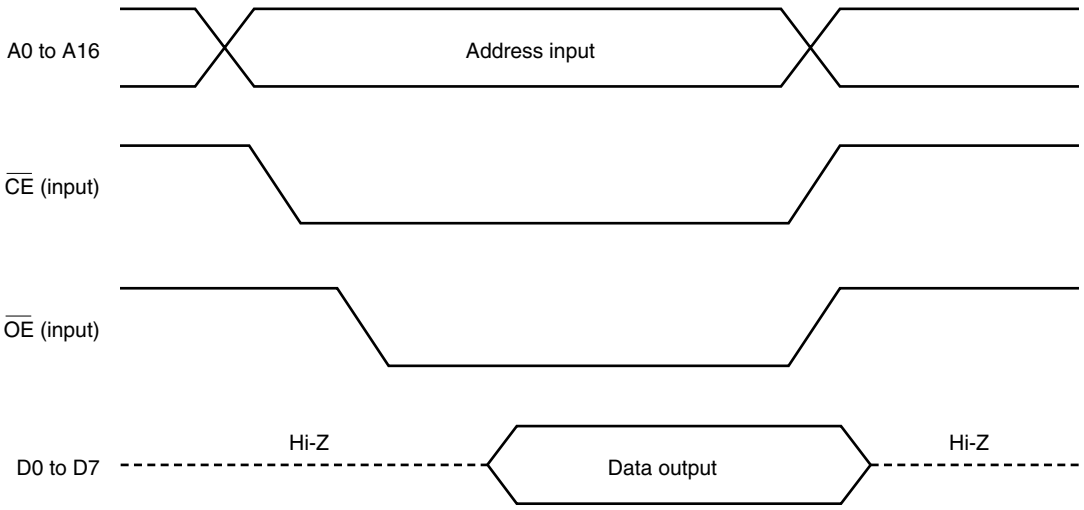
19.3.3 PROM read procedure

PROM contents can be read onto the external data bus (D0 to D7) using the following procedure.

(1) Fix the $\overline{\text{RESET}}$ pin low, and supply +5 V to the V_{PP} pin. Unused pins are handled as shown in **1.5 Pin Configuration (Top View) (2) PROM programming mode**.
(2) Supply +5 V to the V_{DD} and V_{PP} pins.
(3) Input the address of the data to be read to pins A0 to A16.
(4) Read mode.
(5) Output data to pins D0 to D7.

The timing for steps (2) through (5) above is shown in Figure 19-7.

Figure 19-7. PROM Read Timing



19.4 Screening of One-Time PROM Version

A one-time PROM device (μPD78P0208GF-3BA) cannot be fully tested by NEC Electronics before shipment due to the nature of PROM. After the necessary data has been written, it is recommended to implement a screening process, that is, the written contents should be verified after the device has been stored under the following high-temperature conditions.

Storage Temperature	Storage Time
125°C	24 hours

CHAPTER 20 INSTRUCTION SET

This chapter describes the instruction set for the μPD780208 Subseries. For details of the operations and mnemonics (instruction codes) of each instruction, refer to the **78K/0 Series Instructions User's Manual (U12326E)**.

20.1 Conventions

20.1.1 Operand identifiers and description methods

Operands are described in the “Operand” column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more description methods, select one of them. Uppercase letters and the symbols #, !, $, and [ ] are keywords and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- $: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, $, and [ ] symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for description.

Table 20-1. 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 label
saddrp	FE20H to FF1FH Immediate data or label (even addresses only)
addr16	0000H to FFFFH Immediate data or label (only even addresses for 16-bit data transfer instructions)
addr11	0800H to 0FFFH Immediate data or label
addr5	0040H to 007FH Immediate data or label (even addresses only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note FFD0H to FFDFH cannot be addressed.

Remark For special-function register symbols, see **Table 3-3 Special-Function Register List**.

20.1.2 Description of “operation” column

A: A register; 8-bit accumulator
X: X register
B: B register
C: C register
D: D register
E: E register
H: H register
L: L register
AX: AX register pair; 16-bit accumulator
BC: BC register pair
DE: DE register pair
HL: HL register pair
PC: Program counter
SP: Stack pointer
PSW: Program status word
CY: Carry flag
AC: Auxiliary carry flag
Z: Zero flag
RBS: Register bank select flag
IE: Interrupt request enable flag
NMIS: Non-maskable interrupt servicing flag
( ): Memory contents indicated by address or register contents in parentheses
X_H, X_L: Higher 8 bits and lower 8 bits of 16-bit register
$\wedge$: Logical product (AND)
$\vee$: Logical sum (OR)
$\veebar$: Exclusive logical sum (exclusive OR)
$\overline{\quad}$: Inverted data
addr16: 16-bit immediate data or label
jdisp8: Signed 8-bit data (displacement value)

20.1.3 Description of “flag operation” column

(Blank): Unchanged
0: Cleared to 0
1: Set to 1
×: Set/cleared according to the result
R: Previously saved value is restored

20.2 Operation List

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	MOV	r, #byte	2	4	–	r ← byte	
		saddr, #byte	3	6	7	(saddr) ← byte	
		sfr, #byte	3	–	7	sfr ← byte	
		A, r Note 3	1	2	–	A ← r	
		r, A Note 3	1	2	–	r ← A	
		A, saddr	2	4	5	A ← (saddr)	
		saddr, A	2	4	5	(saddr) ← A	
		A, sfr	2	–	5	A ← sfr	
		sfr, A	2	–	5	sfr ← A	
		A, !addr16	3	8	9	A ← (addr16)	
		!addr16, A	3	8	9	(addr16) ← A	
		PSW, #byte	3	–	7	PSW ← byte	× × ×
		A, PSW	2	–	5	A ← PSW	
		PSW, A	2	–	5	PSW ← A	× × ×
		A, [DE]	1	4	5	A ← (DE)	
		[DE], A	1	4	5	(DE) ← A	
		A, [HL]	1	4	5	A ← (HL)	
		[HL], A	1	4	5	(HL) ← A	
		A, [HL+byte]	2	8	9	A ← (HL+byte)	
		[HL+byte], A	2	8	9	(HL+byte) ← A	
		A, [HL+B]	1	6	7	A ← (HL+B)	
		[HL+B], A	1	6	7	(HL+B) ← A	
		A, [HL+C]	1	6	7	A ← (HL+C)	
		[HL+C], A	1	6	7	(HL+C) ← A	
	XCH	A, r Note 3	1	2	–	A ↔ r	
		A, saddr	2	4	6	A ↔ (saddr)	
		A, sfr	2	–	6	A ↔ sfr	
		A, !addr16	3	8	10	A ↔ (addr16)	
		A, [DE]	1	4	6	A ↔ (DE)	
		A, [HL]	1	4	6	A ↔ (HL)	
		A, [HL+byte]	2	8	10	A ↔ (HL+byte)	
		A, [HL+B]	2	8	10	A ↔ (HL+B)	
		A, [HL+C]	2	8	10	A ↔ (HL+C)	

Notes 1. When the internal high-speed RAM area is accessed or an instruction with no data access.

2. When an area except the internal high-speed RAM area is accessed.

3. Except r = A

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
16-bit data transfer	MOVW	rp, #word	3	6	–	rp ← word	
		saddrp, #word	4	8	10	(saddrp) ← word	
		sfrp, #word	4	–	10	sfrp ← word	
		AX, saddrp	2	6	8	AX ← (saddrp)	
		saddrp, AX	2	6	8	(saddrp) ← AX	
		AX, sfrp	2	–	8	AX ← sfrp	
		sfrp, AX	2	–	8	sfrp ← AX	
		AX, rp Note 3	1	4	–	AX ← rp	
		rp, AX Note 3	1	4	–	rp ← AX	
		AX, !addr16	3	10	12	AX ← (addr16)	
		!addr16, AX	3	10	12	(addr16) ← AX	
	XCHW	AX, rp Note 3	1	4	–	AX ↔ rp	
8-bit operation	ADD	A, #byte	2	4	–	A, CY ← A+byte	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr)+byte	× × ×
		A, r Note 4	2	4	–	A, CY ← A+r	× × ×
		r, A	2	4	–	r, CY ← r+A	× × ×
		A, saddr	2	4	5	A, CY ← A+(saddr)	× × ×
		A, !addr16	3	8	9	A, CY ← A+(addr16)	× × ×
		A, [HL]	1	4	5	A, CY ← A+(HL)	× × ×
		A, [HL+byte]	2	8	9	A, CY ← A+(HL+byte)	× × ×
		A, [HL+B]	2	8	9	A, CY ← A+(HL+B)	× × ×
		A, [HL+C]	2	8	9	A, CY ← A+(HL+C)	× × ×
	ADDC	A, #byte	2	4	–	A, CY ← A+byte+CY	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr)+byte+CY	× × ×
		A, r Note 4	2	4	–	A, CY ← A+r+CY	× × ×
		r, A	2	4	–	r, CY ← r+A+CY	× × ×
		A, saddr	2	4	5	A, CY ← A+(saddr)+CY	× × ×
		A, !addr16	3	8	9	A, CY ← A+(addr16)+CY	× × ×
		A, [HL]	1	4	5	A, CY ← A+(HL)+CY	× × ×
		A, [HL+byte]	2	8	9	A, CY ← A+(HL+byte)+CY	× × ×
		A, [HL+B]	2	8	9	A, CY ← A+(HL+B)+CY	× × ×
		A, [HL+C]	2	8	9	A, CY ← A+(HL+C)+CY	× × ×

Notes 1. When the internal high-speed RAM area is accessed or an instruction with no data access.

2. When an area except the internal high-speed RAM area is accessed.

3. Only when rp = BC, DE, or HL

4. Except r = A

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	SUB	A, #byte	2	4	–	A, CY ← A–byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr)–byte	×	×	×
		A, r Note 3	2	4	–	A, CY ← A–r	×	×	×
		r, A	2	4	–	r, CY ← r–A	×	×	×
		A, saddr	2	4	5	A, CY ← A–(saddr)	×	×	×
		A, !addr16	3	8	9	A, CY ← A–(addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A–(HL)	×	×	×
		A, [HL+byte]	2	8	9	A, CY ← A–(HL+byte)	×	×	×
		A, [HL+B]	2	8	9	A, CY ← A–(HL+B)	×	×	×
		A, [HL+C]	2	8	9	A, CY ← A–(HL+C)	×	×	×
	SUBC	A, #byte	2	4	–	A, CY ← A–byte–CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr)–byte–CY	×	×	×
		A, r Note 3	2	4	–	A, CY ← A–r–CY	×	×	×
		r, A	2	4	–	r, CY ← r–A–CY	×	×	×
		A, saddr	2	4	5	A, CY ← A–(saddr)–CY	×	×	×
		A, !addr16	3	8	9	A, CY ← A–(addr16)–CY	×	×	×
		A, [HL]	1	4	5	A, CY ← A–(HL)–CY	×	×	×
		A, [HL+byte]	2	8	9	A, CY ← A–(HL+byte)–CY	×	×	×
		A, [HL+B]	2	8	9	A, CY ← A–(HL+B)–CY	×	×	×
		A, [HL+C]	2	8	9	A, CY ← A–(HL+C)–CY	×	×	×
	AND	A, #byte	2	4	–	A ← A∧byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∧byte	×		
		A, r Note 3	2	4	–	A ← A∧r	×		
		r, A	2	4	–	r ← r∧A	×		
		A, saddr	2	4	5	A ← A∧(saddr)	×		
		A, !addr16	3	8	9	A ← A∧(addr16)	×		
		A, [HL]	1	4	5	A ← A∧(HL)	×		
		A, [HL+byte]	2	8	9	A ← A∧(HL+byte)	×		
		A, [HL+B]	2	8	9	A ← A∧(HL+B)	×		
		A, [HL+C]	2	8	9	A ← A∧(HL+C)	×		

Notes 1. When the internal high-speed RAM area is accessed or an instruction with no data access.

2. When an area except the internal high-speed RAM area is accessed.

3. Except r = A

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit operation	OR	A, #byte	2	4	–	A ← A∨ byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr)∨ byte	×
		A, r Note 3	2	4	–	A ← A∨ r	×
		r, A	2	4	–	r ← r∨A	×
		A, saddr	2	4	5	A ← A∨ (saddr)	×
		A, !addr16	3	8	9	A ← A∨ (addr16)	×
		A, [HL]	1	4	5	A ← A∨ (HL)	×
		A, [HL+byte]	2	8	9	A ← A∨ (HL+byte)	×
		A, [HL+B]	2	8	9	A ← A∨ (HL+B)	×
		A, [HL+C]	2	8	9	A ← A∨ (HL+C)	×
	XOR	A, #byte	2	4	–	A ← A∀ byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∀ byte	×
		A, r Note 3	2	4	–	A ← A∀ r	×
		r, A	2	4	–	r ← r∀ A	×
		A, saddr	2	4	5	A ← A∀ (saddr)	×
		A, !addr16	3	8	9	A ← A∀ (addr16)	×
		A, [HL]	1	4	5	A ← A∀ (HL)	×
		A, [HL+byte]	2	8	9	A ← A∀ (HL+byte)	×
		A, [HL+B]	2	8	9	A ← A∀ (HL+B)	×
		A, [HL+C]	2	8	9	A ← A∀ (HL+C)	×
	CMP	A, #byte	2	4	–	A–byte	× × ×
		saddr, #byte	3	6	8	(saddr)–byte	× × ×
		A, r Note 3	2	4	–	A–r	× × ×
		r, A	2	4	–	r–A	× × ×
		A, saddr	2	4	5	A–(saddr)	× × ×
		A, !addr16	3	8	9	A–(addr16)	× × ×
		A, [HL]	1	4	5	A–(HL)	× × ×
		A, [HL+byte]	2	8	9	A–(HL+byte)	× × ×
		A, [HL+B]	2	8	9	A–(HL+B)	× × ×
		A, [HL+C]	2	8	9	A–(HL+C)	× × ×

Notes 1. When the internal high-speed RAM area is accessed or an instruction with no data access.

2. When an area except the internal high-speed RAM area is accessed.

3. Except r = A

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	ADDW	AX, #word	3	6	–	AX, CY ← AX+word	×	×	×
	SUBW	AX, #word	3	6	–	AX, CY ← AX–word	×	×	×
	CMPW	AX, #word	3	6	–	AX–word	×	×	×
Multiply/ divide	MULU	X	2	16	–	AX ← A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX÷C			
Increase/ decrease	INC	r	1	2	–	r ← r+1	×	×	
		saddr	2	4	6	(saddr) ← (saddr)+1	×	×	
	DEC	r	1	2	–	r ← r–1	×	×	
		saddr	2	4	6	(saddr) ← (saddr)–1	×	×	
	INCW	rp	1	4	–	rp ← rp+1			
	DECW	rp	1	4	–	rp ← rp–1			
Rotation	ROR	A, 1	1	2	–	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1			×
	ROL	A, 1	1	2	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1			×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1			×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1			×
	ROR4	[HL]	2	10	12	A_{3-0} ← $(HL)_{3-0}$, $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$			
	ROL4	[HL]	2	10	12	A_{3-0} ← $(HL)_{7-4}$, $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$			
BCD adjust	ADJBA		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipulation	MOV1	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	8	sfr.bit ← CY			
		A.bit, CY	2	4	–	A.bit ← CY			
		PSW.bit, CY	3	–	8	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	6	8	(HL).bit ← CY			

Notes 1. When the internal high-speed RAM area is accessed or an instruction with no data access.

2. When an area except the internal high-speed RAM area is accessed.

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Bit manipulation	AND1	CY, saddr.bit	3	6	7	CY ← CY ∧ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ∧ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ∧ A.bit	×
		CY, PSW.bit	3	–	7	CY ← CY ∧ PSW.bit	×
		CY, [HL].bit	2	6	7	CY ← CY ∧ (HL).bit	×
	OR1	CY, saddr.bit	3	6	7	CY ← CY ∨ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ∨ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ∨ A.bit	×
		CY, PSW.bit	3	–	7	CY ← CY ∨ PSW.bit	×
		CY, [HL].bit	2	6	7	CY ← CY ∨ (HL).bit	×
	XOR1	CY, saddr.bit	3	6	7	CY ← CY ⊻ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ⊻ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ⊻ A.bit	×
		CY, PSW.bit	3	–	7	CY ← CY ⊻ PSW.bit	×
		CY, [HL].bit	2	6	7	CY ← CY ⊻ (HL).bit	×
	SET1	saddr.bit	2	4	6	(saddr.bit) ← 1	
		sfr.bit	3	–	8	sfr.bit ← 1	
		A.bit	2	4	–	A.bit ← 1	
		PSW.bit	2	–	6	PSW.bit ← 1	× × ×
		[HL].bit	2	6	8	(HL).bit ← 1	
	CLR1	saddr.bit	2	4	6	(saddr.bit) ← 0	
		sfr.bit	3	–	8	sfr.bit ← 0	
		A.bit	2	4	–	A.bit ← 0	
		PSW.bit	2	–	6	PSW.bit ← 0	× × ×
		[HL].bit	2	6	8	(HL).bit ← 0	
	SET1	CY	1	2	–	CY ← 1	1
	CLR1	CY	1	2	–	CY ← 0	0
	NOT1	CY	1	2	–	CY ← $\overline{CY}$	×

Notes 1. When the internal high-speed RAM area is accessed or an instruction with no data access.

2. When an area except the internal high-speed RAM area is accessed.

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Call return	CALL	!addr16	3	7	–	(SP–1) ← (PC+3)H, (SP–2) ← (PC+3)L, PC ← addr16, SP ← SP–2	
	CALLF	!addr11	2	5	–	(SP–1) ← (PC+2)H, (SP–2) ← (PC+2)L, PC15–11 ← 00001, PC10–0 ← addr11, SP ← SP–2	
	CALLT	[addr5]	1	6	–	(SP–1) ← (PC+1)H, (SP–2) ← (PC+1)L, PCH ← (00000000, addr5+1), PCL ← (00000000, addr5), SP ← SP–2	
	BRK		1	6	–	(SP–1) ← PSW, (SP–2) ← (PC+1)H, (SP–3) ← (PC+1)L, PCH ← (003FH), PCL ← (003EH), SP ← SP–3, IE ← 0	
	RET		1	6	–	PCH ← (SP+1), PCL ← (SP), SP ← SP+2	
	RETI		1	6	–	PCH ← (SP+1), PCL ← (SP), PSW ← (SP+2), SP ← SP+3, NMIS ← 0	R R R
	RETB		1	6	–	PCH ← (SP+1), PCL ← (SP), PSW ← (SP+2), SP ← SP+3	R R R
Stack manipulation	PUSH	PSW	1	2	–	(SP–1) ← PSW, SP ← SP–1	
		rp	1	4	–	(SP–1) ← rpH, (SP–2) ← rpL, SP ← SP–2	
	POP	PSW	1	2	–	PSW ← (SP), SP ← SP+1	R R R
		rp	1	4	–	rpH ← (SP+1), rpL ← (SP), SP ← SP+2	
	MOVW	SP, #word	4	–	10	SP ← word	
		SP, AX	2	–	8	SP ← AX	
		AX, SP	2	–	8	AX ← SP	
Unconditional branch	BR	!addr16	3	6	–	PC ← addr16	
		$addr16	2	6	–	PC ← PC + 2 + jdisp8	
		AX	2	8	–	PCH ← A, PCL ← X	
Conditional branch	BC	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 1	
	BNC	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 0	
	BZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 1	
	BNZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 0	

Notes 1. When the internal high-speed RAM area is accessed or an instruction with no data access.

2. When an area except the internal high-speed RAM area is accessed.

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Conditional branch	BT	saddr.bit, $addr16	3	8	9	PC ← PC+3+jdisp8 if (saddr.bit) = 1	
		sfr.bit, $addr16	4	–	11	PC ← PC+4+jdisp8 if sfr.bit = 1	
		A.bit, $addr16	3	8	–	PC ← PC+3+jdisp8 if A.bit = 1	
		PSW.bit, $addr16	3	–	9	PC ← PC+3+jdisp8 if PSW.bit = 1	
		[HL].bit, $addr16	3	10	11	PC ← PC+3+jdisp8 if (HL).bit = 1	
	BF	saddr.bit, $addr16	4	10	11	PC ← PC+4+jdisp8 if (saddr.bit) = 0	
		sfr.bit, $addr16	4	–	11	PC ← PC+4+jdisp8 if sfr.bit = 0	
		A.bit, $addr16	3	8	–	PC ← PC+3+jdisp8 if A.bit = 0	
		PSW.bit, $addr16	4	–	11	PC ← PC+4+jdisp8 if PSW.bit = 0	
		[HL].bit, $addr16	3	10	11	PC ← PC+3+jdisp8 if (HL).bit = 0	
	BTCLR	saddr.bit, $addr16	4	10	12	PC ← PC+4+jdisp8 if (saddr.bit) = 1 then reset (saddr.bit)	
		sfr.bit, $addr16	4	–	12	PC ← PC+4+jdisp8 if sfr.bit = 1 then reset sfr.bit	
		A.bit, $addr16	3	8	–	PC ← PC+3+jdisp8 if A.bit = 1 then reset A.bit	
		PSW.bit, $addr16	4	–	12	PC ← PC+4+jdisp8 if PSW.bit = 1 then reset PSW.bit	× × ×
		[HL].bit, $addr16	3	10	12	PC ← PC+3+jdisp8 if (HL).bit = 1 then reset (HL).bit	
	DBNZ	B, $addr16	2	6	–	B ← B–1, then PC ← PC+2+jdisp8 if B ≠ 0	
		C, $addr16	2	6	–	C ← C–1, then PC ← PC+2+jdisp8 if C ≠ 0	
		saddr, $addr16	3	8	10	(saddr) ← (saddr)–1, then PC ← PC+3+jdisp8 if (saddr) ≠ 0	
CPU control	SEL	RBn	2	4	–	RBS1, 0 ← n	
	NOP		1	2	–	No Operation	
	EI		2	–	6	IE ← 1 (Enable Interrupt)	
	DI		2	–	6	IE ← 0 (Disable Interrupt)	
	HALT		2	6	–	Set HALT Mode	
	STOP		2	6	–	Set STOP Mode	

Notes 1. When the internal high-speed RAM area is accessed or an instruction with no data access.

2. When an area except the internal high-speed RAM area is accessed.

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

20.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

First Operand \ Second Operand	#byte	A	r[Note]	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL+byte] [HL+B] [HL+C]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL+byte] [HL+B] [HL+C]		MOV											
X													MULU
C													DIVUW

Note Except r = A

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

First Operand \ Second Operand	#word	AX	rp[Note]	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW[Note]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, or HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

First Operand \ Second Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

First Operand \ Second Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

APPENDIX A DIFFERENCES BETWEEN μPD78044H, 780228, AND 780208 SUBSERIES

Table A-1 shows the major differences between the μPD78044H, 780228, and 780208 Subseries.

Table A-1. Major Differences Between μPD78044H, 780228, and 780208 Subseries

Item \ Part Number		μPD78044H Subseries	μPD780228 Subseries	μPD780208 Subseries
PROM or flash memory version		μPD78P048B (PROM)	μPD78F0228 (flash memory)	μPD78P0208 (PROM)
Supply voltage		V_{DD} = 2.7 to 5.5 V	V_{DD} = 4.5 to 5.5 V	V_{DD} = 2.7 to 5.5 V
Internal ROM size		μPD78044H: 32 KB μPD78045H: 40 KB μPD78046H: 48 KB μPD78P048B: 60 KB	μPD780226: 48 KB μPD780228: 60 KB μPD78F0228: 60 KB	μPD780204: 32 KB μPD780204A: 32 KB μPD780205: 40 KB μPD780205A: 40 KB μPD780206: 48 KB μPD780208: 60 KB μPD78P0208: 60 KB
Internal expansion RAM size		μPD78P048B only: 1024 bytes	512 bytes	μPD780206, 780208, and 78P0208 only: 1024 bytes
Internal buffer RAM size		μPD78P048B only: 64 bytes	None	64 bytes
VFD display RAM size		48 bytes	96 bytes	80 bytes
CPU clock		Main system clock or subsystem clock selectable	Main system clock only	Main system clock or subsystem clock selectable
I/O ports		68 pins	72 pins	74 pins
Total of VFD display output pins		34 pins	48 pins	53 pins
Serial interface		1 channel		2 channels
Timer		16-bit timer/event counter: 1 channel 8-bit timer/event counter: 2 channels Watch timer: 1 channel Watchdog timer: 1 channel	8-bit remote control timer: 1 channel 8-bit PWM timer: 2 channels Watchdog timer: 1 channel	16-bit timer/event counter: 1 channel 8-bit timer/event counter: 2 channels Watch timer: 1 channel Watchdog timer: 1 channel
Clock output		Provided	None	Provided
Buzzer output		Provided	None	Provided
Vectored interrupt source	Internal	10	8	11
	External	4	4	4
Test input		Provided	None	Provided
★ Package		80-pin plastic QFP (14 × 20)	100-pin plastic QFP (14 × 20)	100-pin plastic QFP (14 × 20)
Electrical specifications and recommended soldering conditions		Refer to individual data sheet.		

Remark In addition to the above items, the configuration of the development tools also differs between the above subseries (especially between the PROM and flash memory versions). For details, refer to the user's manual of each subseries.

★

APPENDIX B DEVELOPMENT TOOLS

The following development tools are available for the development of systems which employ the μPD780208 Subseries.

Figure B-1 shows the configuration of the development tools.

- **Support for PC98-NX series**

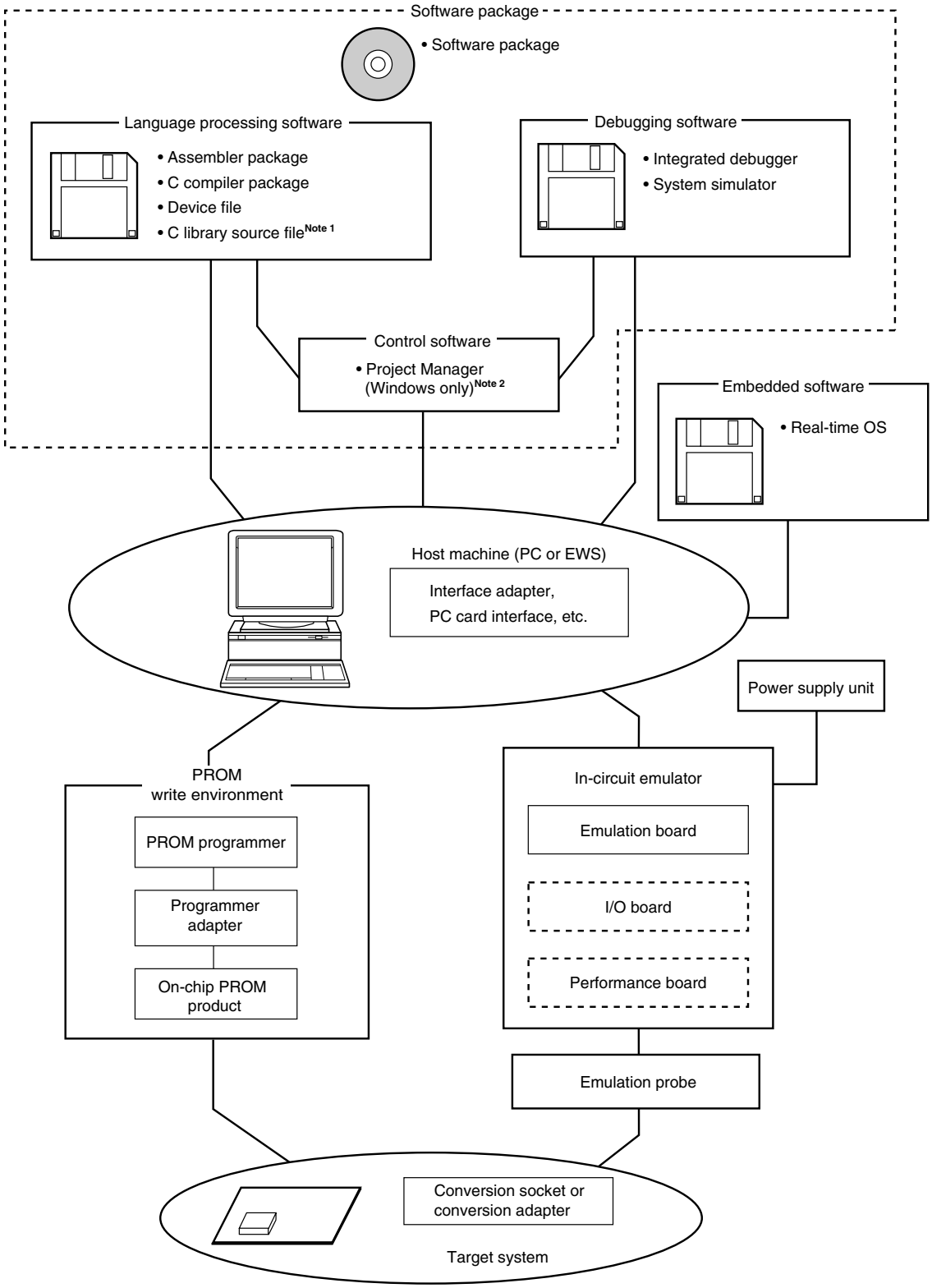
  Unless otherwise specified, products supported by IBM PC/AT™ compatible machines can be used for PC98-NX series computers. When using PC98-NX series computers, refer to the description for IBM PC/AT compatible machines.

- **Windows**

  Unless otherwise specified, "Windows" means the following OSs.

  - Windows 3.1
  - Windows 95
  - Windows 98
  - Windows 2000
  - Windows NT™ Ver. 4.0

Figure B-1. Configuration of Development Tools



Notes 1. The C library source file is not included in the software package.

2. The Project Manager is included in the assembler package.
The Project Manager is only used for Windows.

B.1 Software Package

SP78K0 Software package	This package contains various software tools for 78K/0 Series development. The following tools are included. RA78K0, CC78K0, ID78K0-NS, SM78K0, and various device files
	Part Number: μS××××SP78K0

Remark ×××× in the part number differs depending on the OS used.

μS××××SP78K0

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series,	Windows (Japanese version)	CD-ROM
BB17	IBM PC/AT and compatibles	Windows (English version)	

B.2 Language Processing Software

RA78K0 Assembler package	This assembler converts programs written in mnemonics into object codes executable with a microcontroller. Further, this assembler is provided with functions capable of automatically creating symbol tables and branch instruction optimization. This assembler should be used in combination with a device file (DF780208) (sold separately). **<Caution when using RA78K0 in PC environment>** This assembler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) in Windows.
	Part Number: μS××××RA78K0
CC78K0 C compiler package	This compiler converts programs written in C language into object codes executable with a microcontroller. This compiler should be used in combination with an assembler package and device file (both sold separately). **<Caution when using CC78K0 in PC environment>** This C compiler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) in Windows.
	Part Number: μS××××CC78K0
DF780208**Note 1** Device file	This file contains information peculiar to the device. This device file should be used in combination with tools (RA78K0, CC78K0, SM78K0, ID78K0-NS, ID78K0, and RX78K0) (sold separately). The corresponding OS and host machine differ depending on the tool used.
	Part Number: μS××××DF780208
CC78K0-L**Note 2** C library source file	This is a source file of functions configuring the object library included in the C compiler package. This file is required to match the object library included in C compiler package to the user's specifications. It does not depend on the operating environment because it is a source file.
	Part Number: μS××××CC78K0-L

Notes 1. The DF780208 can be used in common with the RA78K0, CC78K0, SM78K0, ID78K0-NS, ID78K0, and RX78K0.

2. CC78K0-L is not included in the software package (SP78K0).

Remark ××××in the part number differs depending on the host machine and OS used.

μS××××RA78K0
μS××××CC78K0

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT and compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	
AB17		Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	
3P17	HP9000 series 700™	HP-UX™ (Rel. 10.10)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.4), Solaris™ (Rel. 2.5.1)	

μS××××DF780208
μS××××CC78K0-L

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT and compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT
3K13	SPARCstation	SunOS (Rel. 4.1.4), Solaris (Rel. 2.5.1)	3.5-inch 2HD FD
3K15			1/4-inch CGMT

B.3 Control Software

Project Manager	This is control software designed to enable efficient user program development in the Windows environment. All operations used in development of a user program, such as starting the editor, building, and starting the debugger, can be performed from the Project Manager. **<Caution>** The Project Manager is included in the assembler package (RA78K0). It can only be used in Windows.

B.4 PROM Programming Tools

B.4.1 Hardware

PG-1500 PROM programmer	This PROM programmer allows users to encode the PROM in single-chip microcontrollers stand-alone or using a host machine. This requires connection of the accompanying board and separately-sold PROM programmer adapter to the PROM programmer. Besides internal PROMs, general discrete PROM devices whose capacities range from 256 Kb to 4 Mb can be programmed.
PA-78P0208GF PROM programmer adapter	This PROM programmer adapter is for the μPD78P0208 and should be connected to the PG-1500. This adapter is for a 100-pin plastic QFP (GF-3BA type).

B.4.2 Software

PG-1500 controller	This software allows users to control the PG-1500 from a host machine which is connected to the PG-1500 via serial/parallel interface cable(s).
	Part Number: μS××××PG1500

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××PG1500

××××	Host Machine	OS	Supply Medium
5A13	PC-9800 series	MS-DOS	3.5-inch 2HD
5A10		(Ver. 3.30 to Ver. 6.2[Note 1])	5-inch 2HD
7B13	IBM PC/AT and compatibles	**Note 2**	3.5-inch 2HD
7B10			5-inch 2HC

Notes 1. Although a task swap function is incorporated in MS-DOS Ver. 5.0 or later, this function cannot be used with the above software.

2. The following OSs for IBM PCs are supported (Ver. 5.0 or later of MS-DOS has a task swap function, but this function cannot be used with the above software).

OS	Version
PC DOS	Ver.5.02 to Ver.6.3 J6.1/V to J6.3/V (Only the English version is supported.)
MS-DOS	Ver.5.0 to Ver.6.22 5.0/V to 6.2/V (Only the English version is supported.)
IBM DOS™	J5.02/V (Only the English version is supported.)

B.5 Debugging Tools (Hardware)

B.5.1 When using in-circuit emulator IE-78K0-NS, IE-78K0-NS-A

IE-78K0-NS In-circuit emulator		The in-circuit emulator serves to debug hardware and software when developing application systems using a 78K/0 Series product. It can be used with an integrated debugger (ID78K0-NS). This emulator should be used in combination with a power supply unit, emulation probe, and interface adapter, which is required to connect this emulator to the host machine.
IE-78K0-NS-PA Performance board		This board is used for extending the IE-78K0-NS functions. With the addition of this board, the addition of a coverage function, enhancement of tracer and timer functions, and other such debugging function enhancements are possible.
IE-78K0-NS-A In-circuit emulator		In-circuit emulator that combines the IE-78K0-NS and IE-78K0-NS-PA
IE-70000-MC-PS-B Power supply unit		This adapter is used for supplying power from a 100 to 240 V AC outlet.
IE-70000-98-IF-C Interface adapter		This adapter is required when using a PC-9800 series computer (except notebook type) as the IE-78K0-NS host machine (C bus compatible).
IE-70000-CD-IF-A PC card interface		This is the PC card and interface cable required when using a notebook-type computer as the IE-78K0-NS host machine (PCMCIA socket compatible).
IE-70000-PC-IF-C Interface adapter		This adapter is required when using an IBM PC/AT compatible computer as the IE-78K0-NS host machine (ISA bus compatible).
IE-70000-PCI-IF-A Interface adapter		This adapter is required when using a PC with a PCI bus as the IE-78K0-NS host machine.
IE-780208-NS-EM1 Emulation board		This board emulates the operations of the peripheral hardware peculiar to a device. It should be used in combination with an in-circuit emulator.
NP-100GF-TQ NP-H100GF-TQ Emulation probe		This probe is used to connect the in-circuit emulator to the target system and is designed for a 100-pin plastic QFP (GF-3BA type). It should be used in combination with the TGF-100RBP.
	TGF-100RBP Conversion adapter	This conversion socket connects the NP-100GF-TQ or NP-H100GF-TQ to the target system board designed to mount a 100-pin plastic QFP (GF-3BA type).
NP-100GF Emulation probe		This probe is used to connect the in-circuit emulator to the target system and is designed for a 100-pin plastic QFP (GF-3BA type).
	EV-9200GF-100 Conversion socket (See **Figures B-2** and **B-3**)	This conversion socket connects the NP-100GF to the target system board designed to mount a 100-pin plastic QFP (GF-3BA type).

Remarks 1. NP-100GF, NP-100GF-TQ, and NP-H100GF-TQ are products of Naito Densei Machida Mfg. Co., Ltd.

Contact: Naito Densei Machida Mfg. Co., Ltd. +81-45-475-4191

2. TGF-100RBP is a product of TOKYO ELETECH CORPORATION.

Inquiry: Daimaru Kogyo, Ltd. Phone: Tokyo Electronics Dept. +81-3-3820-7112
Osaka Electronics 2nd Dept. +81-6-6244-6672

3. The EV-9200GF-100 is sold in a set of five units.

4. The TGF-100RBP is sold in single units.

B.5.2 When using in-circuit emulator IE-78001-R-A

IE-78001-R-A In-circuit emulator		This is an in-circuit emulator for debugging the hardware and software when an application system using the 78K/0 Series is developed. It can be used with an integrated debugger (ID78K0). This emulator is used with an emulation probe and interface adapter for connecting a host machine.
IE-70000-98-IF-C Interface adapter		This adapter is necessary when a PC-9800 series PC (except notebook type) is used as the host machine for the IE-78001-R-A (C bus compatible).
IE-70000-PC-IF-C Interface adapter		This adapter is necessary when an IBM PC/AT or compatible machine is used as the host machine for the IE-78001-R-A (ISA bus compatible).
IE-780208-R-EM Emulation board		This board is used with an in-circuit emulator to emulate device-specific peripheral hardware.
EP-78064GF-R Emulation probe		This probe is for a 100-pin plastic QFP (GF-3BA type) and connects an in-circuit emulator and the target system.
	EV-9200GF-100 Conversion socket (See **Figures B-2** and **B-3**)	This conversion socket connects the board of the target system created to mount a 100-pin plastic QFP (GF-3BA type) and EP-78064GF-R.

Remark The EV-9200GF-100 is sold in a set of five units.

B.6 Debugging Tools (Software)

SM78K0 System simulator	This is a system simulator for the 78K/0 Series. The SM78K0 is Windows-based software. It is used to perform debugging at the C source level or assembler level while simulating the operation of the target system on a host machine. Use of the SM78K0 allows the execution of application logical testing and performance testing on an independent basis from hardware development, thereby providing higher development efficiency and software quality. The SM78K0 should be used in combination with a device file (DF780208) (sold separately).
	Part Number: μS××××SM78K0
ID78K0-NS Integrated debugger (supporting in-circuit emulators IE-78K0-NS and IE-78K0-NS-A)	This debugger supports the in-circuit emulators for the 78K/0 Series. The ID78K0-NS is Windows-based software. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. It should be used in combination with a device file (sold separately).
ID78K0 Integrated debugger (supporting in-circuit emulator IE-78001-R-A)	
	Part Number: μS××××ID78K0-NS μS××××ID78K0

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SM78K0
μS××××ID78K0-NS
μS××××ID78K0

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT and compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	
AB17		Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

B.7 Embedded Software

RX78K0 Real-time OS	The RX78K0 is a real-time OS conforming to the μITRON specifications. A tool (configurator) for generating the nucleus of the RX78K0 and multiple information tables is supplied. Used in combination with an assembler package (RA78K0) and device file (DF780208) (both sold separately). **<Caution when using RX78K0 in PC environment>** The real-time OS is a DOS-based application. It should be used in the DOS prompt when using in Windows.
	Part Number: μS××××RX78013-ΔΔΔΔ

Caution When purchasing the RX78K0, fill in the purchase application form in advance and sign the user agreement.

Remark ×××× and ΔΔΔΔ in the part number differ depending on the host machine and OS used.

μS××××RX78013-ΔΔΔΔ

ΔΔΔΔ	Product Outline	Maximum Number for Use in Mass Production
001	Evaluation object	Do not use for mass-produced product.
100K	Mass-production object	0.1 million units
001M		1 million units
010M		10 million units
S01	Source program	Source program for mass-produced object

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	Windows (Japanese version)	3.5-inch 2HD FD
AB13	IBM PC/AT and compatibles	Windows (Japanese version)	
BB13		Windows (English version)	

B.8 Method for Upgrading from Former In-Circuit Emulator for 78K/0 Series to IE-78001-R-A

If you already have a former in-circuit emulator for 78K/0 Series microcontrollers (IE-78000-R or IE-78000-R-A), that in-circuit emulator can operate as an equivalent to the IE-78001-R-A by replacing its internal break board with the IE-78001-R-BK.

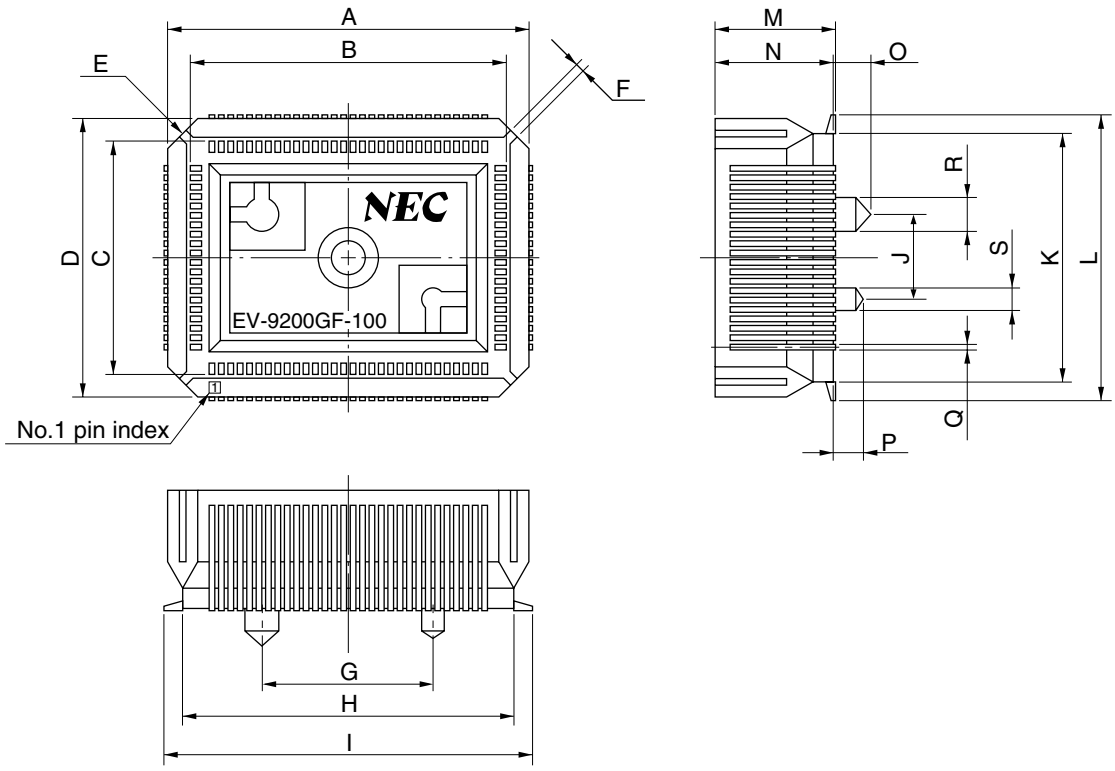
Table B-1. Method for Upgrading from Former In-Circuit Emulator for 78K/0 Series to IE-78001-R-A

In-Circuit Emulator Owned	In-Circuit Emulator Cabinet System-Up[Note]	Board to Be Purchased
IE-78000-R	Required	IE-78001-R-BK
IE-78000-R-A	Not required	

Note For upgrading a cabinet, send your in-circuit emulator to NEC Electronics.

B.9 Conversion Socket (EV-9200GF-100) Package Drawing and Recommended Footprint

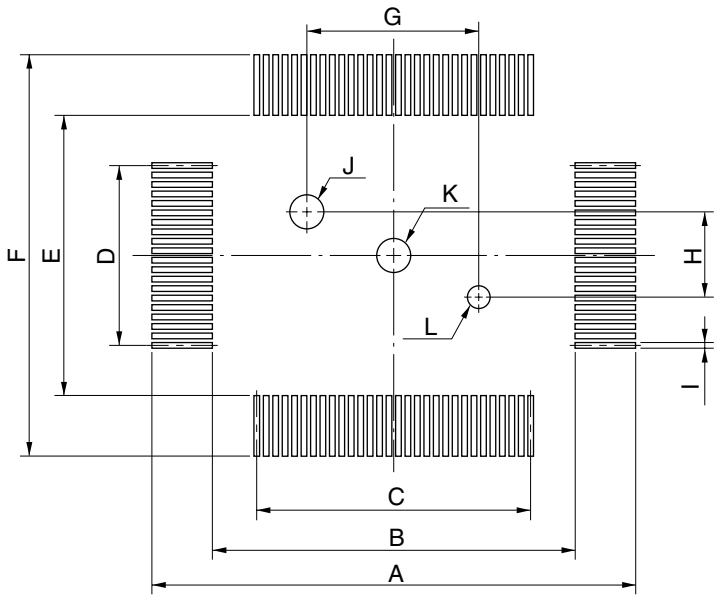
Figure B-2. EV-9200GF-100 Package Drawing (for Reference Purposes only)



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	076
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-3. Recommended Footprint for EV-9200GF-100 (for Reference Purposes only)



EV-9200GF-100-P1

ITEM	MILLIMETERS	INCHES
A	26.3	1.035
B	21.6	0.85
C	0.65±0.02 × 29=18.85±0.05	$0.026^{+0.001}_{-0.002}\times 1.142=0.742^{+0.002}_{-0.002}$
D	0.65±0.02 × 19=12.35±0.05	$0.026^{+0.001}_{-0.002}\times 0.748=0.486^{+0.003}_{-0.002}$
E	15.6	0.614
F	20.3	0.799
G	12±0.05	$0.472^{+0.003}_{-0.002}$
H	6±0.05	$0.236^{+0.003}_{-0.002}$
I	0.35±0.02	$0.014^{+0.001}_{-0.001}$
J	ϕ2.36±0.03	$\phi 0.093^{+0.001}_{-0.002}$
K	ϕ2.3	ϕ0.091
L	ϕ1.57±0.03	$\phi 0.062^{+0.001}_{-0.002}$

Caution The dimensions of the 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 the "Semiconductor Device Mount Manual" website (http://www.necel.com/pkg/en/mount/index.html).

B.10 Notes on Target System Design

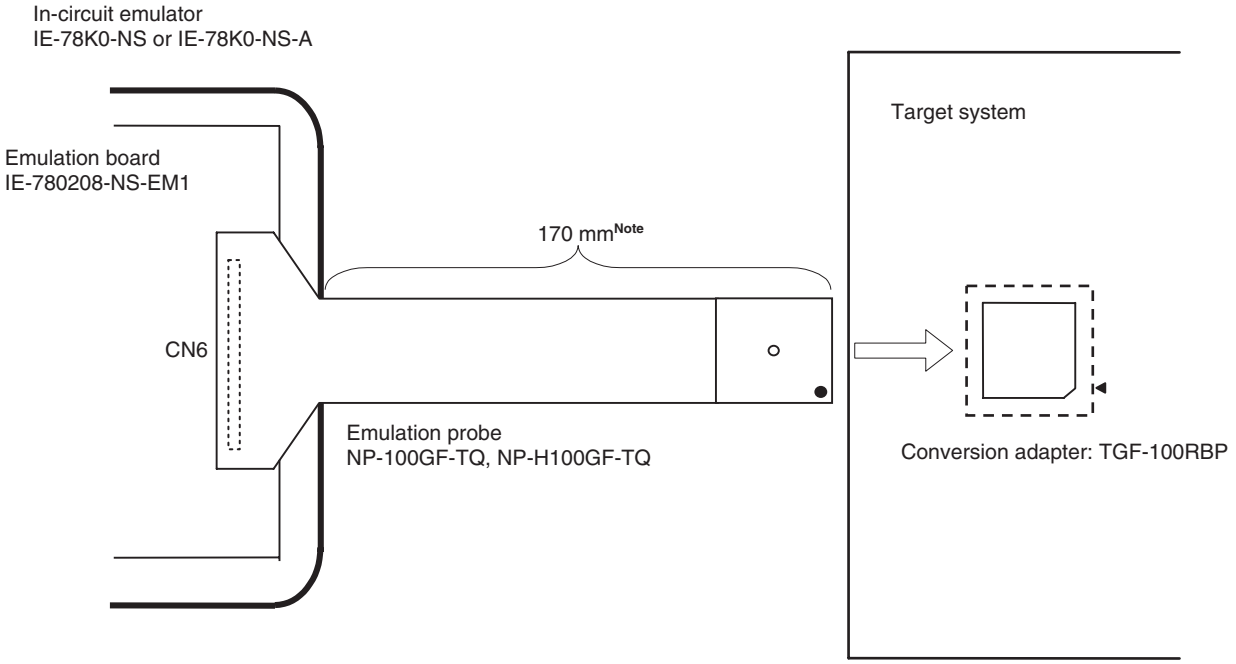
The following shows the conditions when connecting the emulation probe to the conversion adapter. Follow the configuration below and consider the shape of parts to be mounted on the target system when designing a system.

Among the products described in this appendix, the NP-100GF-TQ and NP-H100GF-TQ are products of Naito Densei Machida Mfg. Co., Ltd., and the TGF-100RBP is a product of TOKYO ELETECH CORPORATION.

Table B-2. Distance Between IE System and Conversion Adapter

Emulation Probe	Conversion Adapter	Distance Between IE System and Conversion Adapter
NP-100GF-TQ	TGF-100RBP	170 mm
NP-H100GF-TQ		370 mm

Figure B-4. Distance Between IE System and Conversion Adapter



Note Distance when the NP-100GF-TQ is used. When the NP-H100GF-TQ is used, the distance is 370 mm.

Figure B-5. Connection Conditions of Target System (When NP-100GF-TQ Is Used)

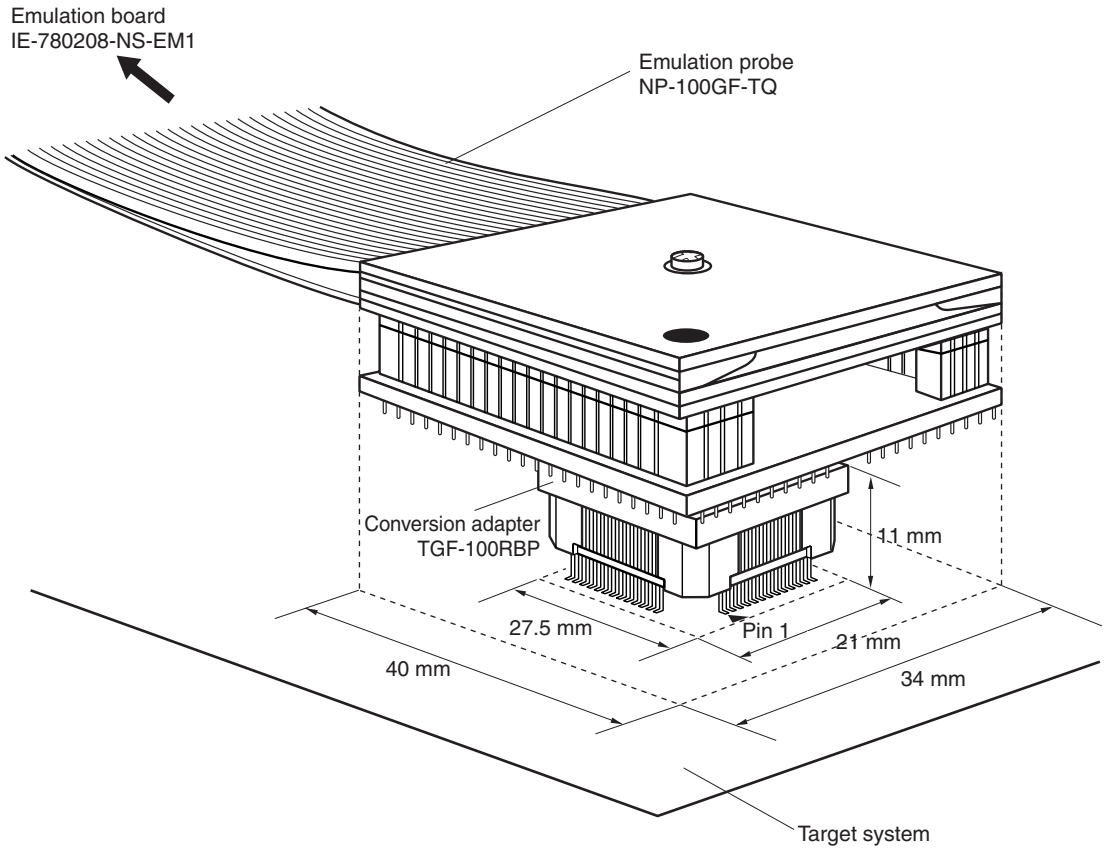
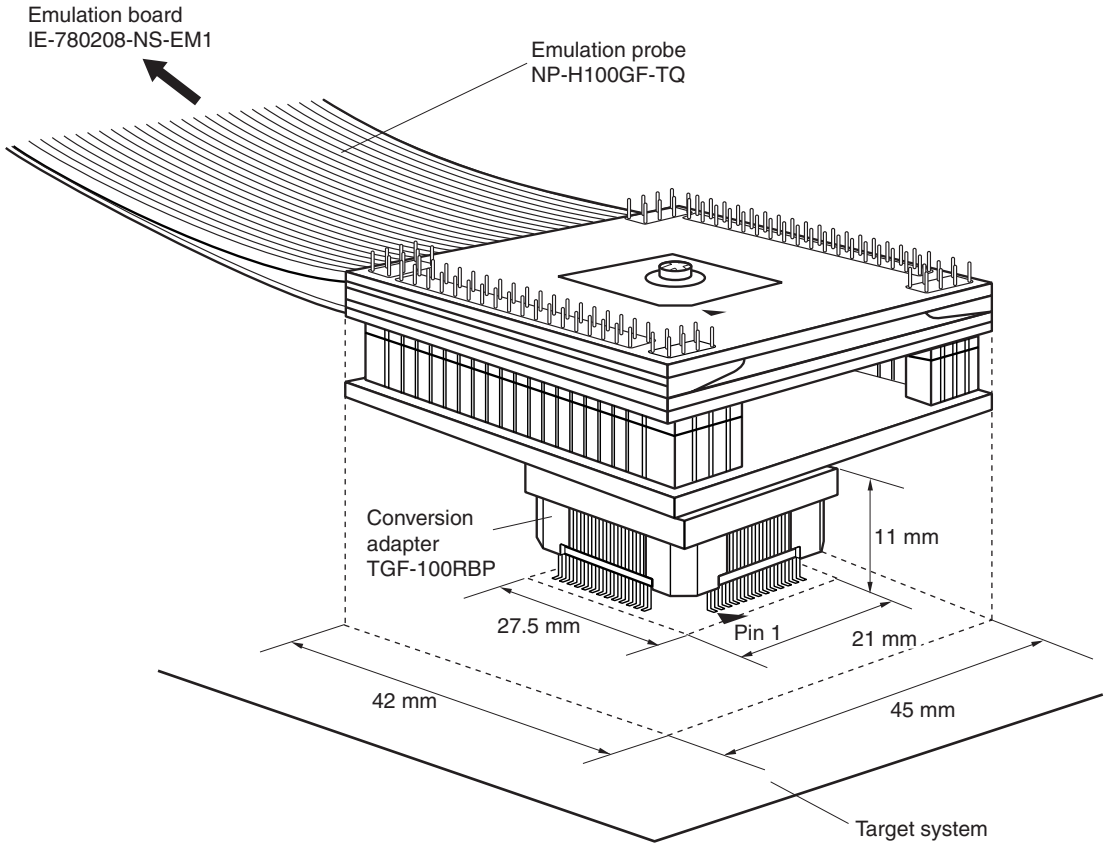


Figure B-6. Connection Conditions of Target System (When NP-H100GF-TQ Is Used)



APPENDIX C REGISTER INDEX

C.1 Register Index (by Register Name)

[S]

[T]

[W]

C.2 Register Index (by Register Symbol)

[S]

[T]

[W]

APPENDIX D REVISION HISTORY

Here is the revision history of this manual. The “Applied to:” column indicates the chapters of each edition in which the revision was applied.

(1/2)

Edition	Revisions from Previous Edition	Applied to:
Second	• The following products are already developed: μPD780204GF-×××-3BA, μPD780205GF-×××-3BA, μPD78P0208GF-×××-3BA, μPD78P0208KL-T • Addition of the μPD780206 and 780208	Throughout
	Change the power supply voltage values in **1.1 Features** and **1.7 Function Outline**	**CHAPTER 1 OUTLINE**
	Addition of the μPD78018F, 78018FY, 78078, 78078Y, 78083, and 780208 Subseries on **1.5 78K/0 Series Expansion**	
	Addition of **Caution** about the condition of input leak current in **4.2.5 Port 7**	**CHAPTER 4 PORT FUNCTIONS**
	Addition of **Note** on **Table 4-3 Port Mode Register and Output Latch Setting when Alternate Function is Used**	
	Addition of 1/2 frequency divider on **Figure 5-1 Clock Generator Block Diagram**	**CHAPTER 5 CLOCK GENERATOR**
	Addition of **Note** and **Caution** on **Figure 9-3 Watchdog Timer Mode Register Format**	**CHAPTER 9 WATCHDOG TIMER**
	Deletion of **CHAPTER 10 6-BIT UP/DOWN COUNTER**	**CHAPTER 10 6-BIT UP/DOWN COUNTER**
	Addition of **Caution** when using standby function on **Figure 12-2 A/D Converter Mode Register Format**	**CHAPTER 12 A/D CONVERTER**
	Addition of **Figure 12-11 AV_{DD} Pin Connection**	
	Addition of **Caution** on **14.4.3 (3) (d) Busy control option**	**CHAPTER 14 SERIAL INTERFACE CHANNEL 1**
	Correction of **APPENDIX A DEVELOPMENT TOOLS**	**APPENDIX A DEVELOPMENT TOOLS**
Third	The following products are already developed: μPD780206GF-×××-3BA, μPD780208GF-×××-3BA	Throughout
	Addition of **Quality Grade**	**CHAPTER 1 OUTLINE**
	Correction of block diagrams of ports 2, 3, and 10 to 12	**CHAPTER 4 PORT FUNCTIONS**
	Change **Caution** when the external clock is input	**CHAPTER 5 CLOCK GENERATOR**
	Addition of **Caution** about changing operation mode of serial interface channel 0	**CHAPTER 13 SERIAL INTERFACE CHANNEL 0**
	Correction of **Note** on bit 7 (BSYE) of serial bus interface control register (SBIC)	
	Addition of explanation of bus release signal, command signal, address, command, data, acknowledge signal, busy signal, and ready signal to the “**Definition of SBI**”	
	Addition of **Caution** for the case that SB0 (SB1) line is changed when the $\overline{SCK0}$ line is in high level in SBI mode	

(2/2)

Edition	Revisions from Previous Edition	Applied to:
Third	Correction of **Cautions** when the STOP mode is set	**CHAPTER 17 STANDBY FUNCTION**
	Addition of **APPENDIX A DIFFERENCES AMONG μPD78044H, 780228, AND 780208 SUBSERIES**	**APPENDIX A DIFFERENCES AMONG μPD78044H, 780228, AND 780208 SUBSERIES**
Fourth	Addition of the following products to target products • μPD780204A • μPD780205A Deletion of the following package from target products • μPD78P0208KL-T (100-pin ceramic WQFN)	Throughout
	• Update of **1.6 78K/0 Series Lineup** • Addition of **Note** in **1.8 Overview of Functions** • Addition of **Caution** in **Table 1-1 Mask Options in Mask ROM Versions**	**CHAPTER 1 OUTLINE**
	• Addition of **2.2.12 V_{LOAD}** • Modification of **Table 2-1 Types of Pin I/O Circuits**	**CHAPTER 2 PIN FUNCTIONS**
	• Addition of **Caution** in **3.1 Memory Space** • Modification of **Note** in **Table 3-3 Special-Function Register List**	**CHAPTER 3 CPU ARCHITECTURE**
	• Addition of **Caution** in **4.2.6 Port 8** • Addition of **Caution** in **4.2.7 Port 9** • Addition of **Caution** in **4.2.8 Port 10** • Addition of **Caution** in **4.2.9 Port 11** • Addition of **Caution** in **4.2.10 Port 12**	**CHAPTER 4 PORT FUNCTIONS**
	• Addition of **Note** in **Figure 5-3 Format of Processor Clock Control Register**	**CHAPTER 5 CLOCK GENERATOR**
	• Modification of **Caution** in **Figure 6-8 Format of External Interrupt Mode Register** • Modification of **6.6 (5) Valid edge setting**	**CHAPTER 6 16-BIT TIMER/EVENT COUNTER**
	• Modification of **Caution** in **Figure 8-2 Format of Timer Clock Select Register 2**	**CHAPTER 8 WATCH TIMER**
	• Modification of **Caution** in **Figure 9-2 Format of Timer Clock Select Register 2**	**CHAPTER 9 WATCHDOG TIMER**
	• Modification of **Caution** in **Figure 11-2 Format of Timer Clock Select Register 2**	**CHAPTER 11 BUZZER OUTPUT CONTROLLER**
	• Addition of **Caution** in **Figure 16-2 Format of Interrupt Request Flag Register** • Modification of **Caution** in **Figure 16-5 Format of External Interrupt Mode Register**	**CHAPTER 16 INTERRUPT AND TEST FUNCTIONS**
	• Addition of description in **Table 17-1 HALT Mode Operating Status** • Addition of description in **Table 17-3 STOP Mode Operating Status**	**CHAPTER 17 STANDBY FUNCTION**
	• Modification of **Table 19-2 Internal Memory Size Switching Register Setting Values**	**CHAPTER 19 μPD78P0208**
	• Modification of description in **Table A-1 Major Differences Between μPD78044H, 780228, and 780208 Subseries**	**APPENDIX A DIFFERENCES BETWEEN μPD78044H, 780228, AND 780208 SUBSERIES**
	• Modification of description	**APPENDIX B DEVELOPMENT TOOLS**