NEC

Preliminary User's Manual

μPD780065 Subseries

8-Bit Single-Chip Microcontrollers

μPD780065
μPD78F0066

Document No. U13420EJ2V0UM00 (2nd edition)
Date Published May 1999 N CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

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

License not needed:

μPD78F0066GC-8BT

The customer must judge the need for a license for the following products:

μPD780065GC-xxx-8BT

- **The information contained in this document is being issued in advance of the production cycle for the device. The parameters for the device may change before final production or NEC Corporation, at its own discretion, may withdraw the device prior to its production.**
- **Not all devices/types available in every country. Please check with local NEC 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 Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.
- NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation or others.
- 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 the customer's equipment shall be done under the full responsibility of the customer. NEC Corporation assumes no responsibility for any losses incurred by the customer or third parties arising from the use of these circuits, software, and information.
- While NEC Corporation has been making continuous effort to enhance the reliability of its semiconductor devices, the possibility of defects cannot be eliminated entirely. To minimize risks of damage or injury to persons or property arising from a defect in an NEC semiconductor device, customers must incorporate sufficient safety measures in its design, such as redundancy, fire-containment, and anti-failure features.
- NEC devices are classified into the following three quality grades:
  "Standard", "Special", and "Specific". The Specific quality grade applies only to devices developed based on a customer designated "quality assurance program" for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device before using it in a particular application.
  - Standard: Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots
  - Special: Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support)
  - Specific: Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems or medical equipment for life support, etc.

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

M5D 98. 12

Regional Information

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

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

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

NEC Electronics Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC Electronics (Germany) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 02
Fax: 0211-65 03 490

NEC Electronics (UK) Ltd.
Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics Italiana s.r.l.
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

NEC Electronics (Germany) GmbH
Benelux Office
Eindhoven, The Netherlands
Tel: 040-2445845
Fax: 040-2444580

NEC Electronics (France) S.A.
Velizy-Villacoublay, France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

NEC Electronics (France) S.A.
Spain Office
Madrid, Spain
Tel: 91-504-2787
Fax: 91-504-2860

NEC Electronics (Germany) GmbH
Scandinavia Office
Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Singapore Pte. Ltd.
United Square, Singapore 1130
Tel: 65-253-8311
Fax: 65-250-3583

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377
Fax: 02-2719-5951

NEC do Brasil S.A.
Electron Devices Division
Rodovia Presidente Dutra, Km 214
07210-902-Guarulhos-SP Brasil
Tel: 55-11-6465-6810
Fax: 55-11-6465-6829

J99.1

Major Revisions in This Edition

Page	Description
p.29	Renewal of **1.5 78K/0 Series Lineup**
p.33	Modification of description of on-chip pull-up resistor specification and function description of TI00 pin in **2.1 Pin Functions**
pp.41, 42	Addition of input/output circuit type and **Figure 2-1 Pin Input/Output Circuits** in **2.3 Pin I/O Circuits and Recommended Connections of Unused Pins**
p.43	Modification of **Caution** in **3.1 Memory Space**
p.56	Modification of following register symbols in **Table 3-3 Special Function Register List** ADTC → ADTC0, ADTP → ADTP0, ADTI → ADTI0
pp.74 to 83	Modification of block diagrams of ports in **Figures 4-2** through **4-11**
p.86	Modification of description and **Caution** in **4.3 (2) Pull-up resistor option registers (PUO, PU2 to PU9)**
p.104	Addition of PPG output column and modification of number of interval timers in **Table 6-1 Timer/ Event Counter Operations**
p.105	Modification of **Figure 6-1 16-Bit Timer/Event Counter Block Diagram**
p.107	Addition of column of CR01 Capture Trigger in **Table 6-3 TI00/TO0/P20 Pin Valid Edge and Capture/Compare Register Capture Register Capture Trigger**
p.111	Addition of a **Caution** on OSPT in **Figure 6-4 Format of 16-Bit Timer Output Control Register 0 (TOC0)**
p.112	Modification of **Cautions** in **Figure 6-5 Format of Prescaler Mode Register 0 (PRM0)**
p.115	Addition of the noise elimination circuit in **Figure 6-8 Interval Timer Configuration Diagram**
p.133	Addition of cautions on sampling clock in **6.6 (12) Edge detection**
p.187	Modification of the input buffer to the schmitt-triggered input in **Figure 13-1 Serial Interface (UART0) Block Diagram**
pp.209 to 250	Modification of following register symbols and bit names in **CHAPTER 14 SERIAL INTERFACE (SIO1)** • Automatic Data Transmit/Receive Address Pointer (ADTP0) • Automatic Data Transmit/Receive Control Register (ADTC0) • Automatic Data Transmit/Receive Interval Specification Register (ADTI0) • Bit name in Serial Operation Mode Register 1 (CSIM1)
pp.251, 257	Deletion of the direction control circuit in **Figure 15-1 Serial Interface (SIO30) Block Diagram**, **Figure 16-1 Serial Interface (SIO31) Block Diagram**
p.264	Addition of descriptions about INTTM00 and INTTM01 triggers in **Table 17-1 Interrupt Source List**
pp.267, 270	Modification of **Table 17-2 Flags Corresponding to Interrupt Request Source** and **Figure 17-4 Format of Priority Specify Flag Registers (PR0L, PR0H, PR1L)** (WTPR → WTPR0)
p.306	Addition of **Caution** in **21.1 Memory Size Switching Register** and modification of **Figure 21-1 Format of Memory Size Switching Register (IMS)** (W → R/W)
pp.308, 331	Addition of Flashpro III in **21.3 Flash Memory Programming** and **A.2 Flash Memory Writing Tools**
pp.327 to 338	Addition of descriptions of Windows™ for supporting PC98-NX series, modification of supported OS version, addition of Solaris™ to OSs, addition of the performance board IE-78K0-NS-PA, and interface adapter IE-7000-PCI-IF in **APPENDIX A DEVELOPMENT TOOLS**

The mark ★ shows major revised points.

INTRODUCTION

Readers

This manual has been prepared for user engineers who understand the functions of the μPD780065 Subseries and wish to design and develop application systems and programs for these devices.

μPD780065 Subseries: μPD780065
μPD78F0066

Purpose

This manual is intended to provide users an understanding of the functions described in the organization below.

Organization

The μPD780065 Subseries manual is separated into two parts: this manual and the instructions edition (common to the 78K/0 Series).

μPD780065 Subseries User's Manual (This Manual)	**78K/0 Series User's Manual Instructions**
• Pin functions • Internal block functions • Interrupt • Other on-chip peripheral functions	• CPU functions • Instruction set • Explanation of each instruction

How To Read This Manual

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

- To gain a general understanding of functions:
  → Read this manual in the order of the contents.
- How to interpret the register format:
  → For the bit number enclosed in square, the bit name is defined as a reserved word in RA78K/0, and in CC78K/0, already defined in the header file named sfrbit.h.
- To check the details of a register when you know the register name.
  → Refer to **APPENDIX C**.

Conventions

Data representation weight:	Higher digits on the left and lower digits on the right
Active low representations:	$\overline{\times\times\times}$ (overscore over pin or signal name)
Note:	Footnote for item marked with Note in the text.
Caution:	Information requiring particular attention
Remarks:	Supplementary information
Numerical representations:	Binary ··· ×××× or ××××B
	Decimal ··· ××××
	Hexadecimal ··· ××××H

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

- **Related documents for the device**

Document Name	Document No.	
	English	Japanese
μPD780065 Preliminary Product Information	U13732E	U13732J
μPD78F0065 Preliminary Product Information	U13419E	U13419J
μPD780065 Subseries User's Manual	This manual	U13420J
78K/0 Series User's Manual-Instructions	U12326E	U12326J
78K/0 Series Instruction Application Table	—	U10903J
78K/0 Series Instruction Set	—	U10904J

- **Related documents for development tool (User's Manual)**

Document Name		Document No.	
		English	Japanese
RA78K0 Assembler Package	Operation	U11802E	U11802J
	Language	U11801E	U11801J
	Structured Assembly Language	U11789E	U11789J
RA78K Series Structured Assembler Preprocessor		EEU-1402	U12323J
CC78K0 C Compiler	Operation	U11517E	U11517J
	Language	U11518E	U11518J
CC78K/0 C Compiler Application Note	Programming Know-how	U13034E	U13034J
IE-78K0-NS		To be prepared	To be prepared
IE-78001-R-A		To be prepared	To be prepared
IE-78K0-R-EX1		To be prepared	To be prepared
IE-780066-NS-EM4		To be prepared	To be prepared
EP-78230		EEU-1515	EEU-985
SM78K0 System Simulator Windows Based	Reference	U10181E	U10181J
SM78K Series System Simulator	External part user open Interface	U10092E	U10092J
ID78K0-NS Integrated Debugger Windows Based	Reference	U12900E	U12900J
ID78K0 Integrated Debugger EWS Based	Reference	—	U11151J
ID78K0 Integrated Debugger PC Based	Reference	U11539E	U11539J
ID78K0 Integrated Debugger Windows Based	Guide	U11649E	U11649J

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

- **Related documents for embedded software (User's Manual)**

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

- **Other Documents**

Document Name	Document No.	
	English	Japanese
SEMICONDUCTORS SELECTION GUIDE Products & Packages (CD-ROM)	X13769X	
Semiconductor Device Mounting Technology Manual	C10535E	C10535J
Quality Grades on NEC Semiconductor Devices	C11531E	C11531J
NEC Semiconductor Device Reliability/Quality Control System	C10983E	C10983J
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E	C11892J
Guide to Microcomputer-Related Products by Third Party	—	U11416J

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

[MEMO]

CONTENTS

LIST OF FIGURES (1/5)

LIST OF FIGURES (2/5)

LIST OF FIGURES (3/5)

LIST OF FIGURES (4/5)

LIST OF FIGURES (5/5)

[MEMO]

LIST OF TABLES (1/2)

LIST OF TABLES (2/2)

CHAPTER 1 OUTLINE

1.1 Features

- Internal Memory

Type / Part Number	Program Memory (Mask ROM/Flash memory)	Data Memory		
		High-Speed RAM	Expansion RAM	Buffer RAM
μPD780065	40 Kbytes	1024 bytes	4096 bytes	32 bytes
μPD78F0066	48 Kbytes **Note**			

Note The capacity of internal flash memory can be changed by means of the memory size switching register (IMS).

- External Memory Expansion Space: 64 Kbytes
- Minimum instruction execution time changeable from high speed (0.24 μs: @ 8.38-MHz operation with main system clock) to ultra-low speed (122 μs: @ 32.768-kHz operation with subsystem clock)
- Instruction set suited to system control
  - Bit manipulation possible in all address spaces
  - Multiply and divide instructions
- Sixty I/O ports
- 8-bit resolution A/D converter: 8 channels
- Serial interface: 4 channels
  - 3-wire serial I/O mode (provided with automatic transmission/reception function): 1 channel
  - 3-wire serial I/O mode: 1 channel
  - 2-wire serial I/O mode: 1 channel
  - UART mode: 1 channel
- Timer: Five channels
  - 16-bit timer/event counter: 1 channel
  - 8-bit timer/event counter: 2 channels
  - Watch timer: 1 channel
  - Watchdog timer: 1 channel
- ★ Vectored interrupts: 20
- Two types of on-chip clock oscillators (main system clock and subsystem clock)
- Power supply voltage: V_{DD} = 2.7 to 5.5 V

1.2 Applications

Car audio supporting CD text, etc.

1.3 Ordering Information

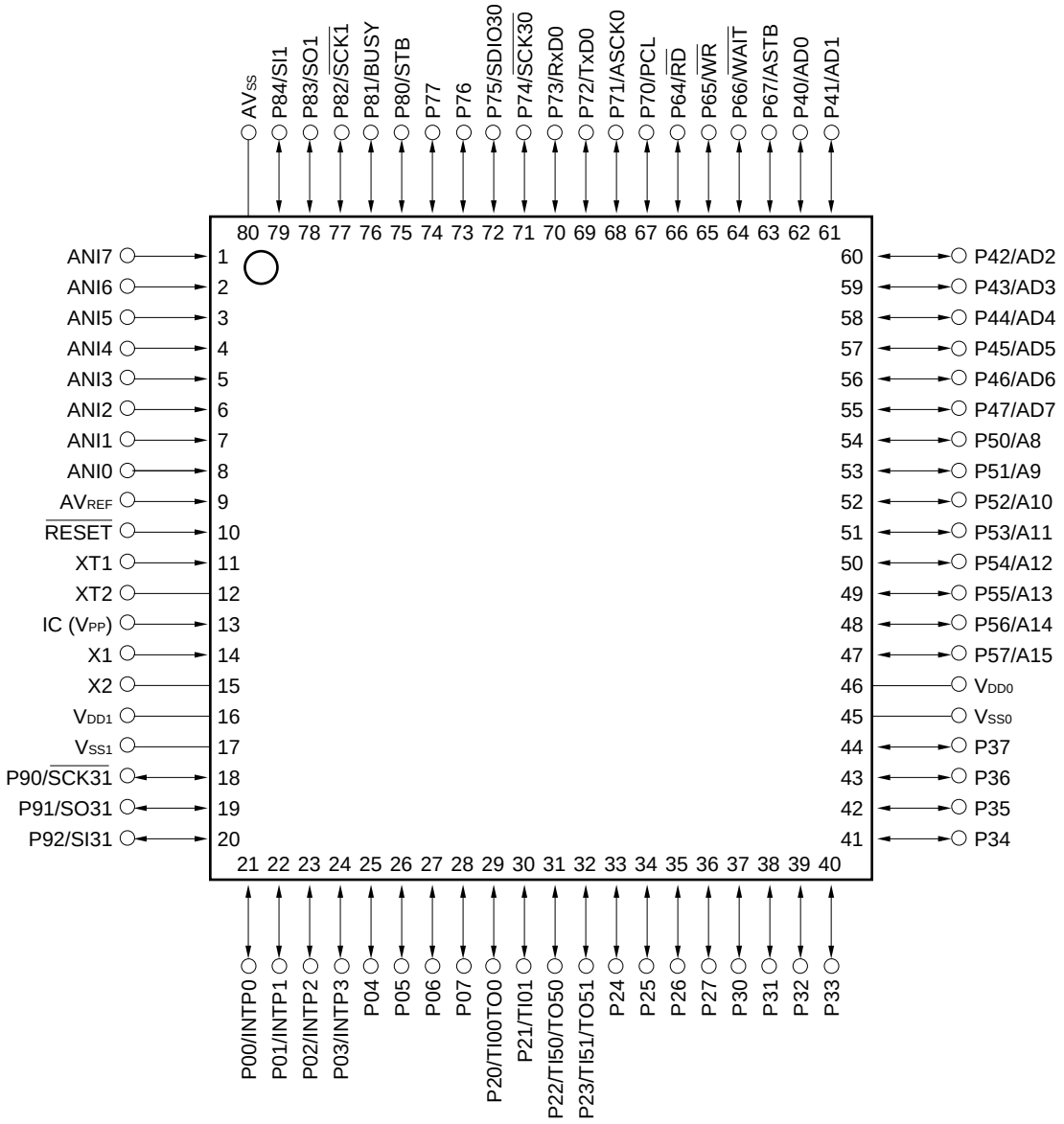
Part Number	Package	Internal ROM
μPD780065GC-×××-8BT	80-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78F0066GC-×××-8BT	80-pin plastic QFP (14 × 14 mm)	Flash memory

Remark ××× indicates ROM code suffix.

1.4 Pin Configuration (Top View)

- **80-pin plastic QFP (14 × 14 mm)**

  μPD780065GC-×××-8BT
  μPD78F0066GC-8BT



Cautions 1. Connect directly IC (Internally Connected) pin to V_{SS0} or V_{SS1}.

2. Connect AV_{SS} pin to V_{SS0}.

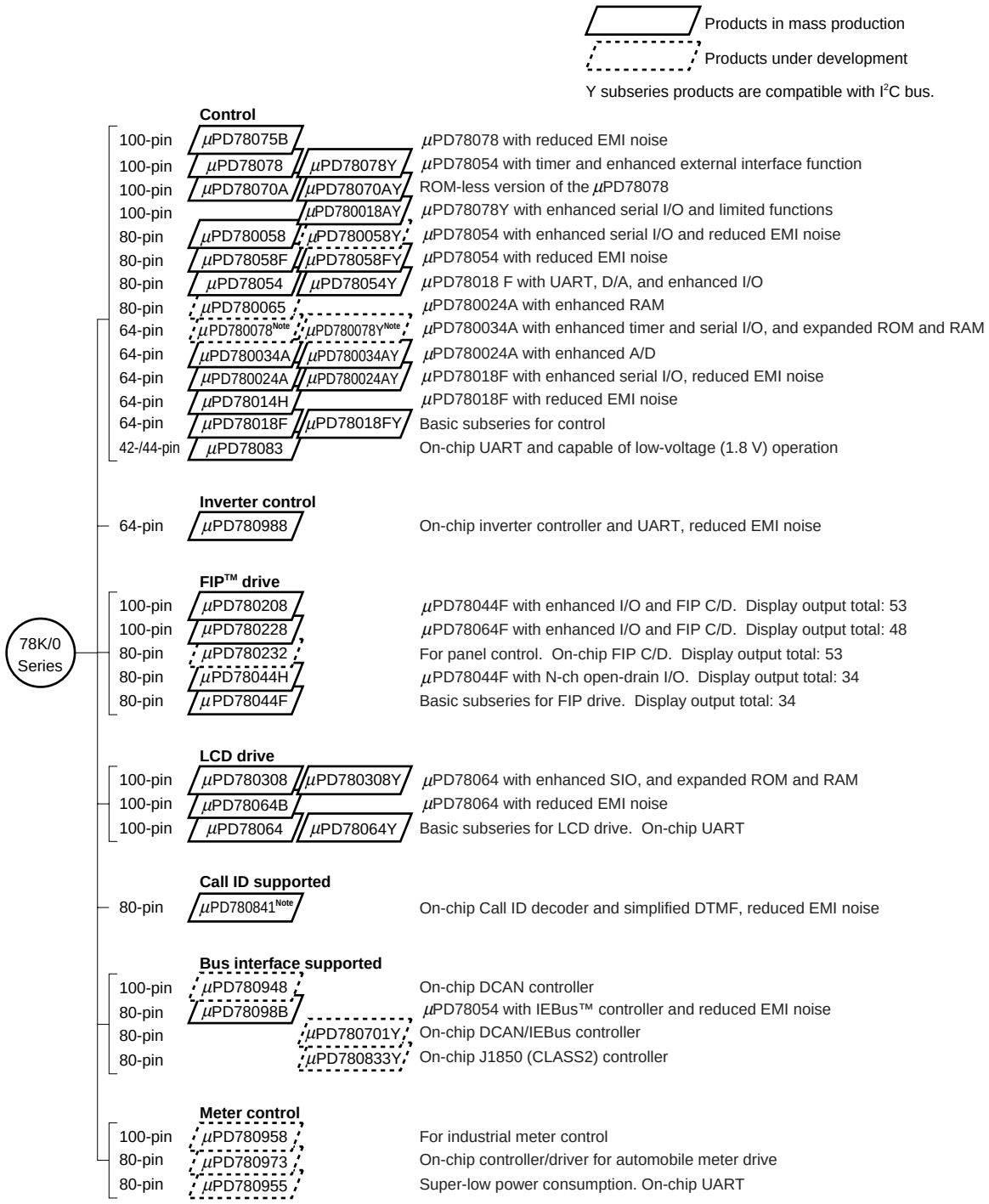
Remarks 1. When these devices are used in applications that require the reduction of noise generated from an on-chip microcontroller, the implementation of anti-noise measures is recommended, such as supplying to V_{DD0} and V_{DD1} independently, connecting V_{SS0} and V_{SS1} independently to ground lines, and so on.

2. Pin connection in parentheses is intended for the μPD78F0066.

A8 to A15:	Address Bus	PCL:	Programmable Clock
AD0 to AD7:	Address/Data Bus	$\overline{RD}$:	Read Strobe
ANI0 to ANI7:	Analog Input	$\overline{RESET}$:	Reset
ASCK0:	Asynchronous Serial Clock	RxD0:	Receive Data
ASTB:	Address Strobe	$\overline{SCK1}$, $\overline{SCK30}$, $\overline{SCK31}$:	Serial Clock
AV_{REF}:	Analog Reference Voltage	SDIO30:	Serial Data Input/Output
AV_{SS}:	Analog Ground	SI1, SI31:	Serial Input
BUSY:	Busy	SO1, SO31:	Serial Output
IC:	Internally Connected	STB:	Strobe
INTP0 to INTP3:	Interrupt from Peripherals	TI00, TI01, TI50, TI51:	Timer Input
P00 to P07:	Port0	TO0, TO50, TO51:	Timer Output
P20 to P27:	Port2	TxD0:	Transmit Data
P30 to P37:	Port3	V_{DD0}, V_{DD1}:	Power Supply
P40 to P47:	Port4	V_{PP}:	Programming Power Supply
P50 to P57:	Port5	V_{SS0}, V_{SS1}:	Ground
P64 to P67:	Port6	$\overline{WAIT}$:	Wait
P70 to P77:	Port7	$\overline{WR}$:	Write Strobe
P80 to P84:	Port8	X1, X2:	Crystal (Main System Clock)
P90 to P92:	Port9	XT1, XT2:	Crystal (Subsystem Clock)

★ 1.5 78K/0 Series Lineup

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



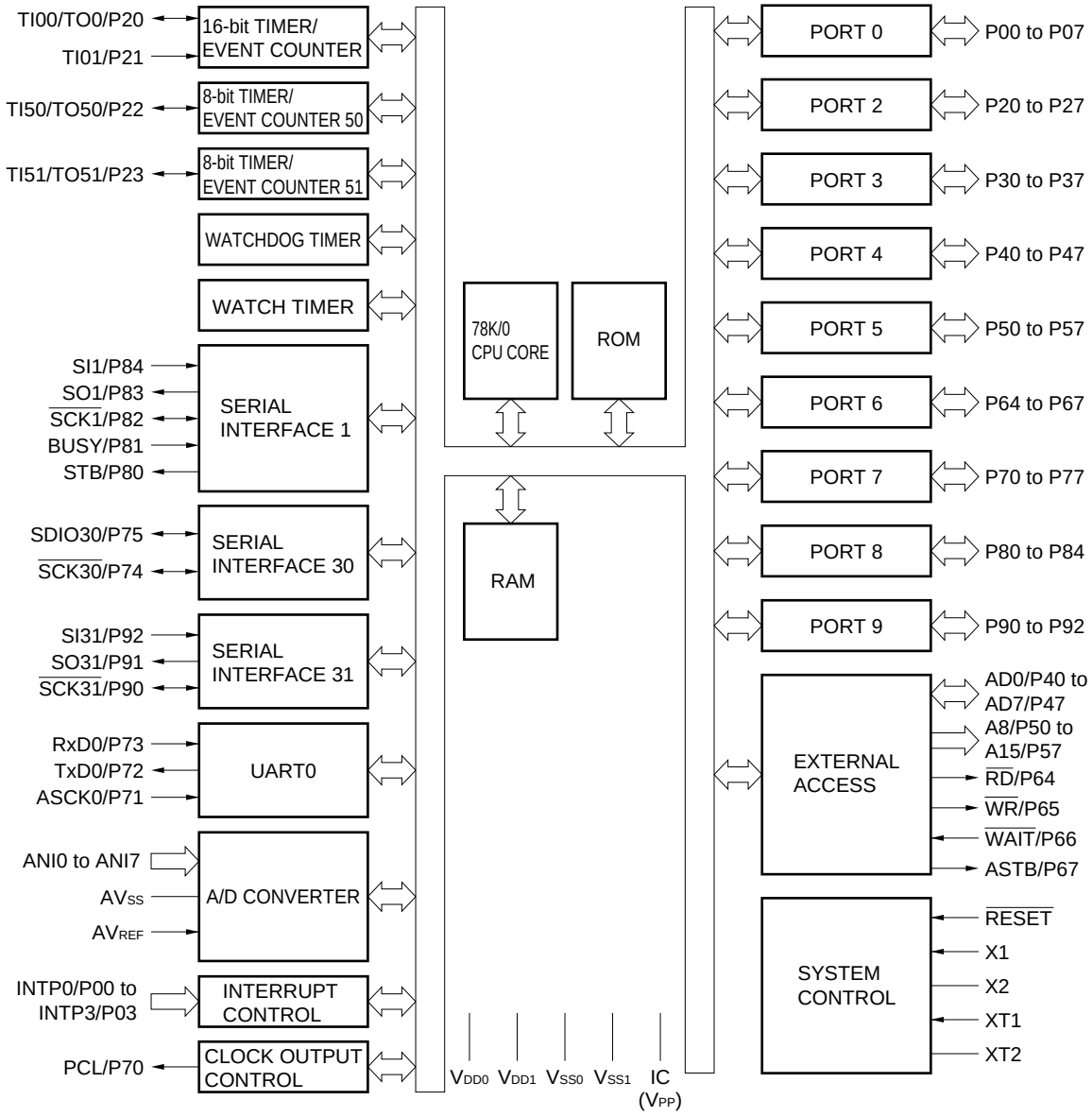
Note Under planning

Major functional differences among the subseries are listed below.

Subseries Name \ Function		ROM Capacity	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, time-division 3-wire: 1 ch)	51		
	μPD780024A						8 ch	—					
	μPD78014H									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	√
FIP drive	μPD780208	32 K to 60 K	2 ch	1 ch	1 ch	1 ch	8 ch	—	—	2 ch	74	2.7 V	—
	μPD780228	48 K to 60 K	3 ch	—	—					1 ch	72	4.5 V	
	μPD780232	16 K to 24 K					4 ch			2 ch	40		
	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	μPD780308	48 K to 60 K	2 ch	1 ch	1 ch	1 ch	8 ch	—	—	3 ch (time-division UART: 1 ch)	57	2.0 V	—
	μPD78064B	32 K								2 ch (UART: 1 ch)			
	μPD78064	16 K to 32 K											
Call ID supported	μPD780841	24 K to 32 K	2 ch	—	1 ch	1 ch	2 ch	—	—	2 ch (UART: 1 ch)	58	2.7 V	—
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	—
Meter control	μPD780958	48 K to 60 K	4 ch	2 ch	—	1 ch	—	—	—	2 ch (UART: 1 ch)	69	2.2 V	—
	μPD780973	24 K to 32 K	3 ch	1 ch	1 ch		5 ch				56	4.5 V	
	μPD780955	40 K	6 ch		—		1 ch			2 ch (UART: 2 ch)	50	4.2 V	

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

1.6 Block Diagram



Remarks 1. The internal ROM capacity depends on the product.

2. Pin connection in parentheses is intended for the μPD78F0066.

1.7 Outline of Function

Item \ Part Number		μPD780065	μPD78F0066
Internal memory	ROM	40 Kbytes (Mask ROM)	48 Kbytes **Note** (Flash memory)
	High-speed RAM	1024 bytes	
	Expansion RAM	4096 bytes	
	Buffer RAM	32 bytes	
Memory space		64 Kbytes	
General register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)	
Minimum instruction execution time		Minimum instruction execution time changeable function	
	When main system clock selected	0.24 μs/0.48 μs/0.95 μs/1.91 μs/3.81 μs (@ 8.38-MHz operation)	
	When subsystem clock selected	122 μs (@ 32.768-kHz operation)	
Instruction set		• 16-bit operation • Multiply/divide (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc.	
I/O port		CMOS I/O: 60	
A/D converter		• 8-bit resolution × 8 channels • Low-voltage operation: AV_{DD} = 2.7 to 5.5 V	
Serial interface		• 3-wire serial I/O mode: 1 channel • 3-wire serial I/O mode (provided with automatic transmit/receive function): 1 channel • 2-wire serial I/O mode: 1 channel • UART mode: 1 channel	
Timer		• 16-bit timer/event counter: 1 channel • 8-bit timer/event counter: 2 channels • Watch timer: 1 channel • Watchdog timer: 1 channel	
Timer output		Three outputs: (8-bit PWM output enable: 2)	
Clock output		• 65.5 kHz, 131 kHz, 262 kHz, 524 kHz, 1.05 MHz, 2.10 MHz, 4.19 MHz, 8.38 MHz (8.38 MHz with main system clock) • 32.768 kHz (32.768 kHz with subsystem clock)	
★ Vectored interrupt	Maskable	Internal: 14, External: 4	
	Non-maskable	Internal: 1	
	Software	1	
Power supply voltage		V_{DD} = 2.7 to 5.5 V	
Operating ambient temperature		T_A = –40 to +85°C	
Package		80-pin plastic QFP (14 × 14 mm)	

Note The capacity of internal flash memory can be changed by means of the memory size switching register (IMS).

CHAPTER 2 PIN FUNCTION

2.1 Pin Functions

(1) Port Pins (1/2)

Pin Name	Input/Output	Function	After Reset	Alternate Function
P00 to P03	Input/Output	Port 0 8-bit input/output port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by means of software.	Input	INTP0 to INTP3
★ P04 to P07				—
P20	Input/Output	Port 2 8-bit input/output port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by means of software.	Input	TI00/TC10
P21				TI01
P22				TI50/TO50
P23				TI51/TO51
P24 to P27				—
P30 to P37	Input/Output	Port 3 8-bit input/output port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by means of software.	Input	—
P40 to P47	Input/Output	Port 4 8-bit input/output port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by means of software.	Input	AD0 to AD7
P50 to P57	Input/Output	Port 5 8-bit input/output port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by means of software.	Input	A8 to A15
P64	Input/Output	Port 6 4-bit input/output port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by means of software.	Input	$\overline{RD}$
P65				$\overline{WR}$
★ P66				$\overline{WAIT}$
P67				ASTB

(1) Port Pins (2/2)

Pin Name	Input/Output	Function	After Reset	Alternate Function
P70	Input/Output	Port 7 8-bit input/output port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by means of software.	Input	PCL
P71				ASCK0
★ P72				TxD0
P73				RxD0
P74				$\overline{\text{SCK30}}$
P75				SDIO30
P76, P77				—
P80	Input/Output	Port 8 5-bit input/output port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by means of software.	Input	STB
P81				BUSY
P82				$\overline{\text{SCK1}}$
P83				SO1
P84				SI1
P90	Input/Output	Port 9 3-bit input/output port Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by means of software.	Input	$\overline{\text{SCK31}}$
P91				SO31
★ P92				SI31

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

Pin Name	Input/Output	Function	After Reset	Alternate Function
INTP0 to INTP3	Input	External interrupt request input with specifiable valid edges (rising edge, falling edge, both rising and falling edges)	Input	P00 to P03
SI1	Input	Serial interface serial data input	Input	P84
SI31				P92
SO1	Output	Serial interface serial data output	Input	P83
SO31				P91
SDIO30	Input/Output	Serial interface serial data input/output	Input	P75
$\overline{\text{SCK1}}$	Input/Output	Serial interface serial clock input/output	Input	P82
$\overline{\text{SCK30}}$				P74
$\overline{\text{SCK31}}$				P90
STB	Output	Strobe output for serial interface automatic transmission/ reception	Input	P80
BUSY	Input	Busy input for serial interface automatic transmission/ reception	Input	P81
RxD0	Input	Asynchronous serial interface serial data input	Input	P73
TxD0	Output	Asynchronous serial interface serial data output	Input	P72

(2) Non-port Pins (2/2)

Pin Name	Input/Output	Function	After Reset	Alternate Function
ASCK0	Input	Asynchronous serial interface serial clock input	Input	P71
TI00	Input	External count clock input to 16-bit timer (TM0) Capture trigger input to TM0 capture registers (CR00, CR01)	Input	P20/TO0
TI01		Capture trigger input to TM0 capture register (CR00)		P21
TI50		External count clock input to 8-bit timer (TM50)		P22/TO50
TI51		External count clock input to 8-bit timer (TM51)		P23/TO51
TO0	Output	16-bit timer TM0 output	Input	P20/TI00
TO50		8-bit timer (TM50) output (also used for 8-bit PWM output)	Input	P22/TI50
TO51		8-bit timer (TM51) output (also used for 8-bit PWM output)		P23/TI51
PCL	Output	Clock output (for main system clock and subsystem clock trimming)	Input	P70
AD0 to AD7	Input/Output	Lower-order address/data bus when expanding external memory	Input	P40 to P47
A8 to A15	Output	High-order address bus when expanding external memory	Input	P50 to P57
$\overline{RD}$	Output	Strobe signal output for read operation from external memory	Input	P64
$\overline{WR}$		Strobe signal output for write operation from external memory		P65
$\overline{WAIT}$	Input	Wait insertion when accessing external memory	Input	P66
ASTB	Output	Strobe output externally latching address information output to ports 4, 5 to access external memory	Input	P67
ANI0 to ANI7	Input	A/D converter analog input	Input	—
AV_{REF}	Input	A/D converter reference voltage input (also used for analog power supply)	—	—
AV_{SS}	—	A/D converter ground potential. Connect to V_{SS0} or V_{SS1}	—	—
$\overline{RESET}$	Input	System reset input	—	—
X1	Input	Crystal connection for main system clock oscillation	—	—
X2	—		—	—
XT1	Input	Crystal connection for subsystem clock oscillation	—	—
XT2	—		—	—
V_{DD0}	—	Positive power supply	—	—
V_{SS0}	—	Ground potential	—	—
V_{DD1}	—	Positive power supply other than port	—	—
V_{SS1}	—	Ground potential other than port	—	—
IC	—	Internally connected. Connect directly to V_{SS0} or V_{SS1}	—	—
V_{PP}	—	High-voltage application for program write/verify Connect directly to V_{SS0} or V_{SS1} in normal operating mode.	—	—

2.2 Description of Pin Functions

2.2.1 P00 to P07 (Port 0)

This is an 8-bit input/output port. Besides serving as an input/output port, these pins function as external interrupt request inputs.

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

(1) Port mode

These pins function as an 8-bit input/output port.

These pins can be specified for input or output in 1-bit units with port mode register 0 (PM0). On-chip pull-up resistors can be specified by defining pull-up resistor option register 0 (PU0).

(2) Control mode

In this mode, these pins function as external interrupt request inputs (INTP0 to INTP3).

INTP0 to INTP3 are external interrupt request input pins for which valid edges (rising edge, falling edge, and both rising and falling edges) can be specified.

2.2.2 P20 to P27 (Port 2)

This is an 8-bit input/output port. Besides serving as an input/output port, these pins function as timer input/outputs.

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

(1) Port mode

These pins function as an 8-bit input/output port. They can be specified in 1-bit units as input or output ports with port mode register 2 (PM2). On-chip pull-up resistors can be specified by defining pull-up resistor option register 2 (PU2). P20 and P21 also function as 16-bit timer/event counter capture trigger signal input pins through a valid edge input.

(2) Control mode

These pins function as timer input/outputs.

(a) TI00

A 16-bit timer/event counter external count clock input pin and 16-bit timer/event counter capture trigger signal input pin for the capture register (CR01).

(b) TI01

A 16-bit timer/event counter capture trigger signal input pin for the capture register (CR00).

(c) TI50, TI51

An 8-bit timer/event counter external count clock input pins.

(d) TO0, TO50, TO51

Timer output pins.

2.2.3 P30 to P37 (Port 3)

This is an 8-bit input/output port. These pins can be specified in 1-bit units as input or output ports with port mode register 3 (PM3). On-chip pull-up resistors can be specified by defining pull-up resistor option register 3 (PU3).

2.2.4 P40 to P47 (Port 4)

This is an 8-bit input/output port. Besides serving as input/output ports, these pins function as an address/data bus.

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

(1) Port mode

These pins function as an 8-bit input/output port. They can be specified in 1-bit units as input or output ports with port mode register 4 (PM4). On-chip pull-up resistors can be specified by defining pull-up resistor option register 4 (PU4).

(2) Control mode

These pins function as low-order address/data bus pins (AD0 to AD7) in external memory expansion mode.

2.2.5 P50 to P57 (Port 5)

This is an 8-bit input/output port. Besides serving as input/output ports, these pins function as an address bus.

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

(1) Port mode

These pins function as an 8-bit input/output port. They can be specified in 1-bit units as input/output ports with port mode register 5 (PM5). On-chip pull-up resistors can be specified by defining pull-up resistor option register 5 (PU5).

(2) Control mode

These pins function as high-order address bus pins (A8 to A15) in external memory expansion mode.

2.2.6 P64 to P67 (Port 6)

This is an 4-bit input/output port. Besides serving as input/output ports, these pins are used for control in external memory expansion mode.

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

(1) Port mode

These pins function as a 4-bit input/output port. They can be specified in 1-bit units as input or output ports with port mode register 6 (PM6). On-chip pull-up resistors can be specified by defining pull-up resistor option register 6 (PU6).

(2) Control mode

These pins function as control signal output pins ($\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB) in external memory expansion mode.

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

2.2.7 P70 to P77 (Port 7)

This is an 8-bit input/output port. Besides serving as input/output ports, these pins function as the serial data input/output, serial clock input/output, and clock output of the serial interface.

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

(1) Port mode

These pins function as an 8-bit input/output port. They can be specified in 1-bit units as input or output ports with port mode register 7 (PM7). On-chip pull-up resistors can be specified by defining pull-up resistor option register 7 (PU7).

(2) Control mode

These pins function as serial data input/output, serial clock input/output, and clock output of the serial interface.

(a) SDIO30

The serial interface serial data input/output pin.

(b) $\overline{\text{SCK30}}$

The serial interface serial clock input/output pin.

(c) RxD0, TxD0

The asynchronous serial interface serial data input/output pin.

(d) ASCK0

The asynchronous serial interface serial clock input pin.

(e) PCL

The clock output pin.

2.2.8 P80 to P84 (Port 8)

This is a 5-bit input/output port. Besides serving as input/output ports, these pins function as the data input/output, clock input/output, automatic transmission/reception busy input, and strobe output of serial interface.

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

(1) Port mode

These pins function as a 5-bit input/output port. They can be specified in 1-bit units as an input or output ports with port mode register 8 (PM8). On-chip pull-up resistors can be specified by defining pull-up resistor option register 8 (PU8).

(2) Control mode

These pins function as the serial interface data input/output, clock input/output, busy input for automatic transmission/reception, and strobe output.

(a) SI1, SO1

The serial interface serial data input/output pins.

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

The serial interface serial clock input/output pin.

(c) BUSY

The serial interface automatic transmission/reception busy input pin.

(d) STB

The serial interface automatic transmission/reception strobe output pin.

2.2.9 P90 to P92 (Port 9)

This is a 3-bit input/output port. Besides serving as serial interface input/output ports, these pins function as the data input/output and clock input/output.

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

(1) Port mode

These pins function as a 3-bit input/output port. They can be specified in 1-bit units as an input or output ports with port mode register 9 (PM9). On-chip pull-up resistors can be specified by defining pull-up resistor option register 9 (PU9).

(2) Control mode

These pins function as the serial interface data input/output and clock input/output.

(a) SI31, SO31

The serial interface serial data input/output pins.

(b) $\overline{\text{SCK31}}$

The serial interface serial clock input/output pin.

2.2.10 ANI0 to ANI7

The analog input pin of the A/D converter.

2.2.11 AV_{REF}

This is an A/D converter reference voltage input pin. This pin is also used for analog power supply.

When no A/D converter is used, connect this pin to V_{SS0}.

2.2.12 AV_{SS}

This is a ground voltage pin of A/D converter. Always use the same voltage as that of the V_{SS0} pin even when no A/D converter is used.

2.2.13 $\overline{RESET}$

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

2.2.14 X1 and X2

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

2.2.15 XT1 and XT2

Crystal resonator connect pins for subsystem clock oscillation.

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

2.2.16 V_{DD0} and V_{DD1}

V_{DD0} is a positive power supply port pin.

V_{DD1} is a positive power supply pin other than port pin.

2.2.17 V_{SS0} and V_{SS1}

V_{SS0} is a ground potential port pin.

V_{SS1} is a ground potential pin other than port pin.

2.2.18 V_{PP} (flash memory version only)

High-voltage apply pin for flash memory programming mode setting and program write/verify.

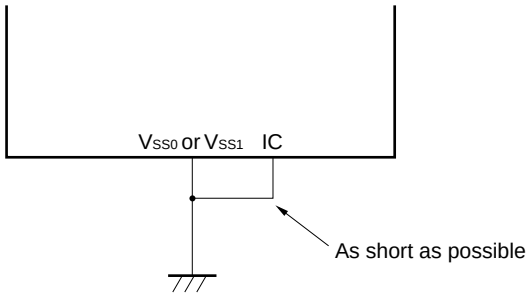
Connect directly to V_{SS0} or V_{SS1} in the normal operating mode.

2.2.19 IC (mask ROM version only)

The IC (Internally Connected) pin is provided to set the test mode to check the μPD780065 Subseries at delivery. Connect it directly to the V_{SS0} or V_{SS1} with the shortest possible wire in the normal operating mode.

When a voltage difference is produced between the IC pin and V_{SS0} pin or V_{SS1} pin, because the wiring between those two pins is too long or an external noise is input to the IC pin, the user's program may not operate normally.

- **Connect IC pin to V_{SS0} pin or V_{SS1} pin directly.**



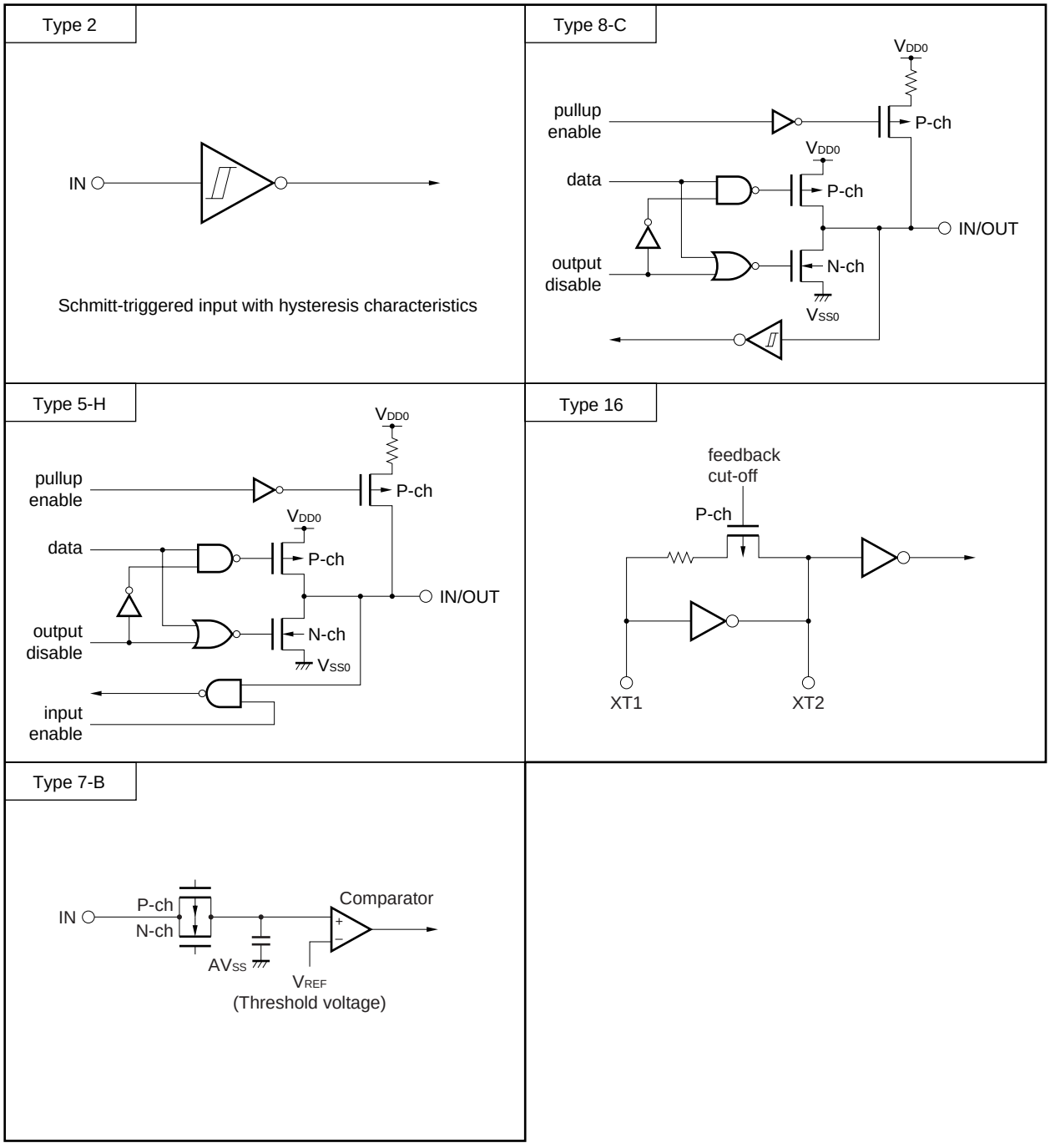
★ **2.3 Pin I/O Circuits and Recommended Connections of Unused Pins**

The input/output circuit type of each pin and recommended connections of unused pins are shown in Table 2-1. For the input/output circuit configuration of each type, refer to Figure 2-1.

Table 2-1. Connections of Unused Pins

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connections of Unused Pins
P00/INTP0 to P03/INTP3	8-C	Input/output	Independently connect to V_{SS0} via a resistor.
P04 to P07			
P20/T100/TO0			Independently connect to V_{DD0} or V_{SS0} via a resistor.
P21/T101			
P22/TI50/TO50			
P23/TI51/TO51			
P24 to P27			
P30 to P37			
P40/AD0 to P47/AD7	5-H		Independently connect to V_{DD0} via a resistor.
P50/A8 to P57/A15			Independently connect to V_{DD0} or V_{SS0} via a resistor.
P64/$\overline{RD}$			
P65/$\overline{WR}$			
P66/$\overline{WAIT}$			
P67/ASTB			
P70/PCL			
P71/ASCK0	8-C		
P72/TxD0	5-H		
P73/RxD0	8-C		
P74/$\overline{SCK30}$			
P75/SDIO30	5-H		
P76, P77	8-C		
P80/STB	5-H		
P81/BUSY	8-C		
P82/$\overline{SCK1}$			
P83/SO1	5-H		
P84/SI1	8-C		
P90/$\overline{SCK31}$			
P91/SO31	5-H		
P92/SI31	8-C		
ANI0 to ANI7	7-B	Input	Independently connect to V_{DD0} or V_{SS0} via a resistor.
XT1	16		Connect to V_{DD0}.
XT2		—	Leave open.
$\overline{RESET}$	2	Input	—
AV_{REF}	—	—	Connect to V_{SS0}.
AV_{SS}			
IC (for mask ROM version)			Directly connect to V_{SS0} or V_{SS1}.
V_{PP} (for flash memory version)			

★ **Figure 2-1. Pin Input/Output Circuits**



CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

μPD780065 Subseries can access 64-Kbyte memory space respectively.
Figures 3-1 and 3-2 show memory maps.

★ **Caution As the initial setting of the program, set the memory size switching register (IMS) and internal expansion RAM size switching register (IXS) as follows. Setting the initial value of IMS and IXS is prohibited.**

	Memory Size Switching Register (IMS)	Internal RAM Size Switching Register (IXS)
μPD780065	CAH	04H
μPD78F0066	CCH or the value corresponding to mask ROM version	

Figure 3-1. Memory Map (μPD780065)

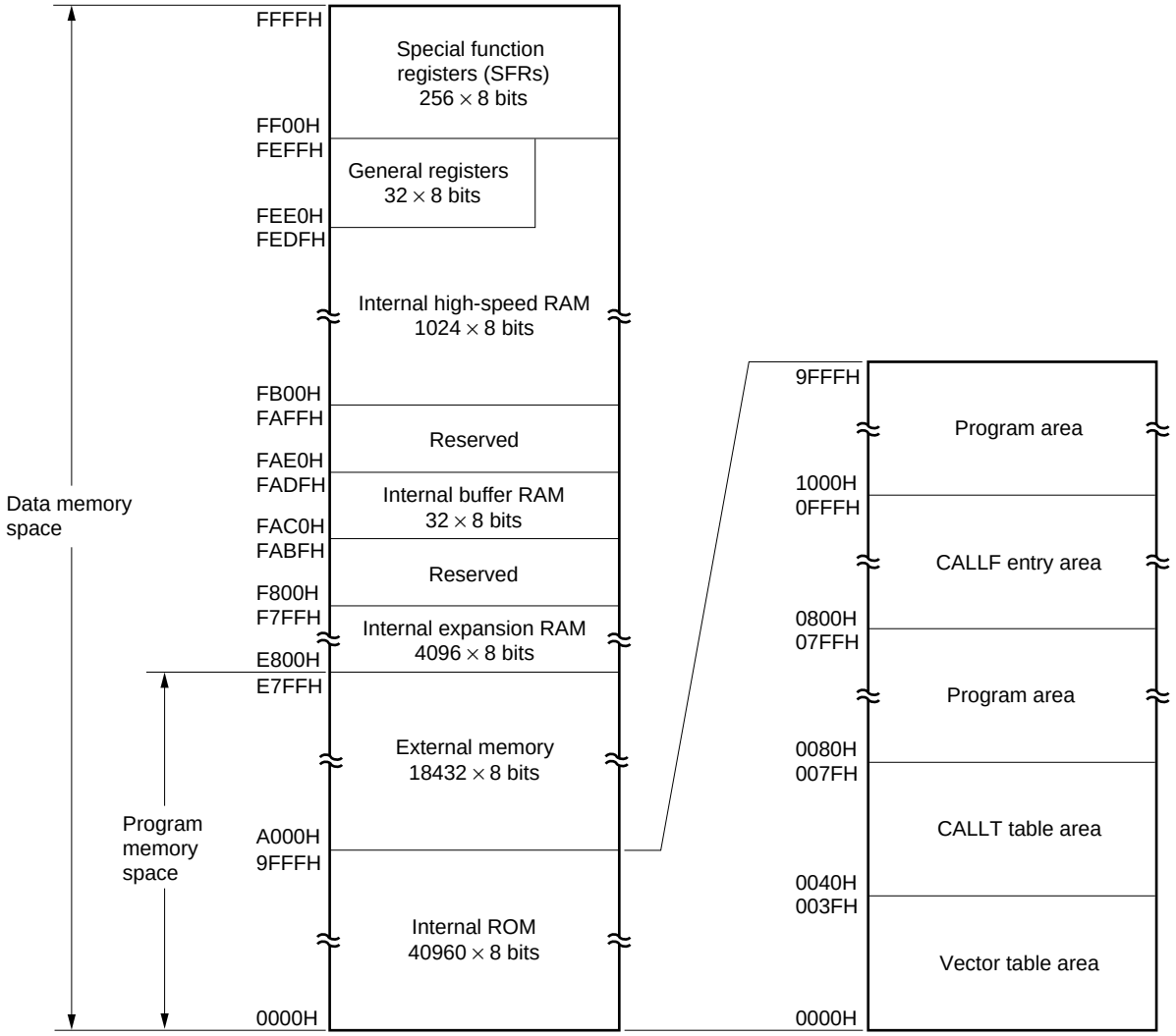
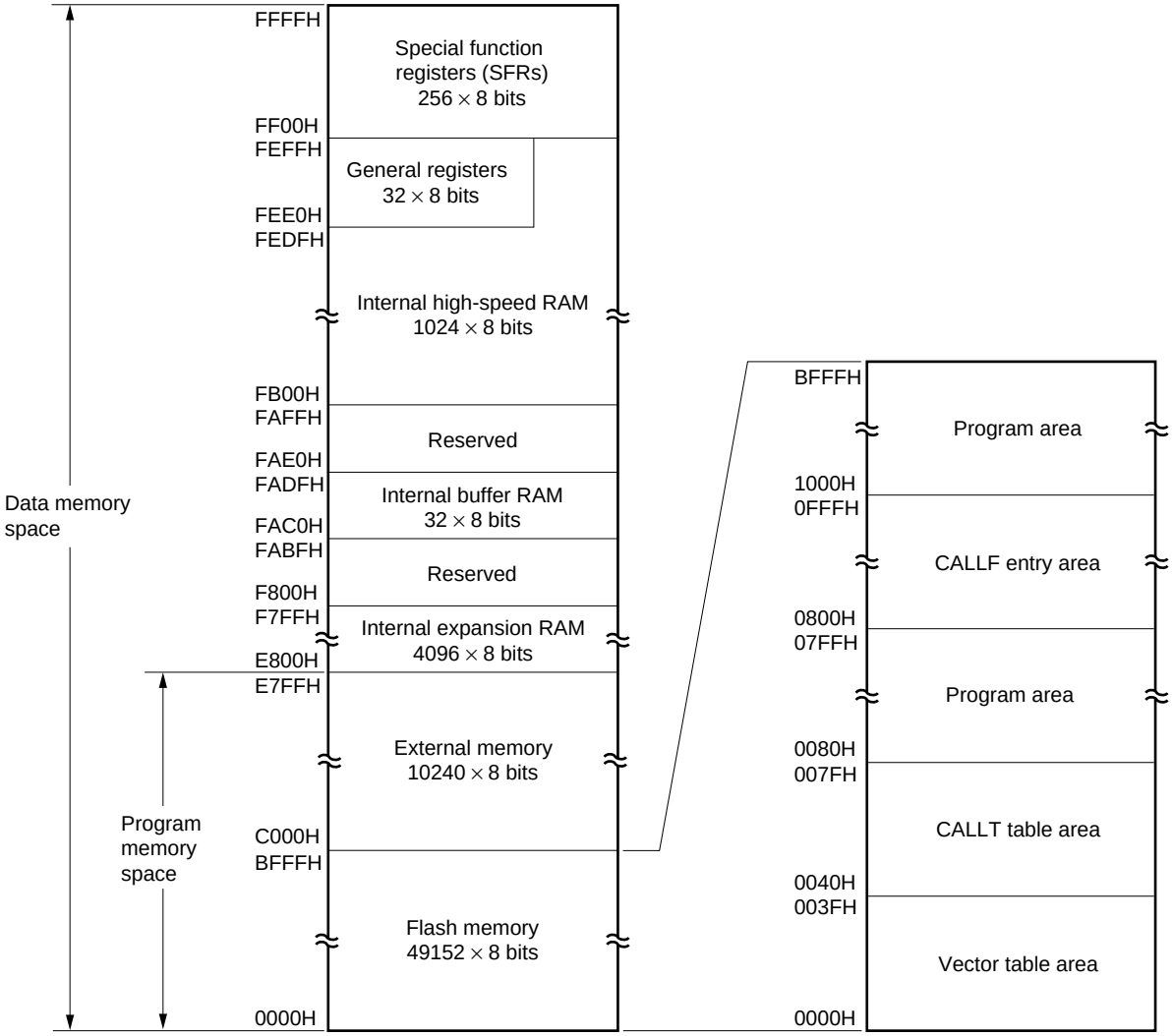


Figure 3-2. Memory Map (μPD78F0066)



3.1.1 Internal program memory space

The internal program memory space contains the program and table data. Normally, it is addressed with the program counter (PC).

The μPD780065 Subseries products incorporate an internal ROM (or flash memory), as listed below.

Table 3-1. Internal ROM Capacity

Part Number	Internal ROM	
	Type	Capacity
μPD780065	Mask ROM	40960 × 8 bits (0000H to 9FFFH)
μPD78F0066	Flash Memory	49152 × 8 bits (0000H to BFFFH)

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

(1) Vector table area

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

Table 3-2. Vector Table

Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input
0004H	INTWDT
0006H	INTP0
0008H	INTP1
000AH	INTP2
000CH	INTP3
000EH	INTSER0
0010H	INTSR0
0012H	INTST0
0014H	INTCSI30
0016H	INTCSI31
0018H	INTCSI1
001AH	INTTM00
001CH	INTTM01
001EH	INTTM50
0020H	INTTM51
0022H	INTWTI
0024H	INTWT
0026H	INTAD0
003EH	BRK

(2) CALLT instruction table area

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

(3) CALLF instruction entry area

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

3.1.2 Internal data memory space

The μPD780065 Subseries incorporates the following RAM.

(1) Internal high-speed RAM

The internal high-speed RAM has an 1024 × 8-bit (FB00H to FEFFH) configuration. Four general-purpose register banks composed of eight 8-bit registers are allocated to the 32-byte area of FEE0H to FEFFH.

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

(2) Internal buffer RAM

An internal buffer RAM is allocated to the 32-byte area of FAC0H to FADFH. The internal buffer RAM is used for storing transmitted/received data to/from the serial interface (SIO1) (a 3-wire serial I/O mode with automatic transmission/reception function). When the internal buffer RAM is not used in the 3-wire serial I/O mode with automatic transmission/reception function, it is used as a conventional RAM.

3.1.3 Special Function Register (SFR) area

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

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

3.1.4 External memory space

The external memory space is accessible via the memory expansion mode register (MEM). The external memory space can store programs, table data, etc., and allocate peripheral devices.

3.1.5 Data memory addressing

Addressing refers to the method of specifying the address of the instruction to be executed next or the address of the register or memory relevant to the execution of instructions.

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

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the μPD780065 Subseries, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFR) and general-purpose registers are available for use. Data memory addressing is illustrated in Figures 3-3 and 3-4. For the details of each addressing mode, see **3.4 Operand Address Addressing**.

Figure 3-3. Data Memory Addressing (μPD780065)

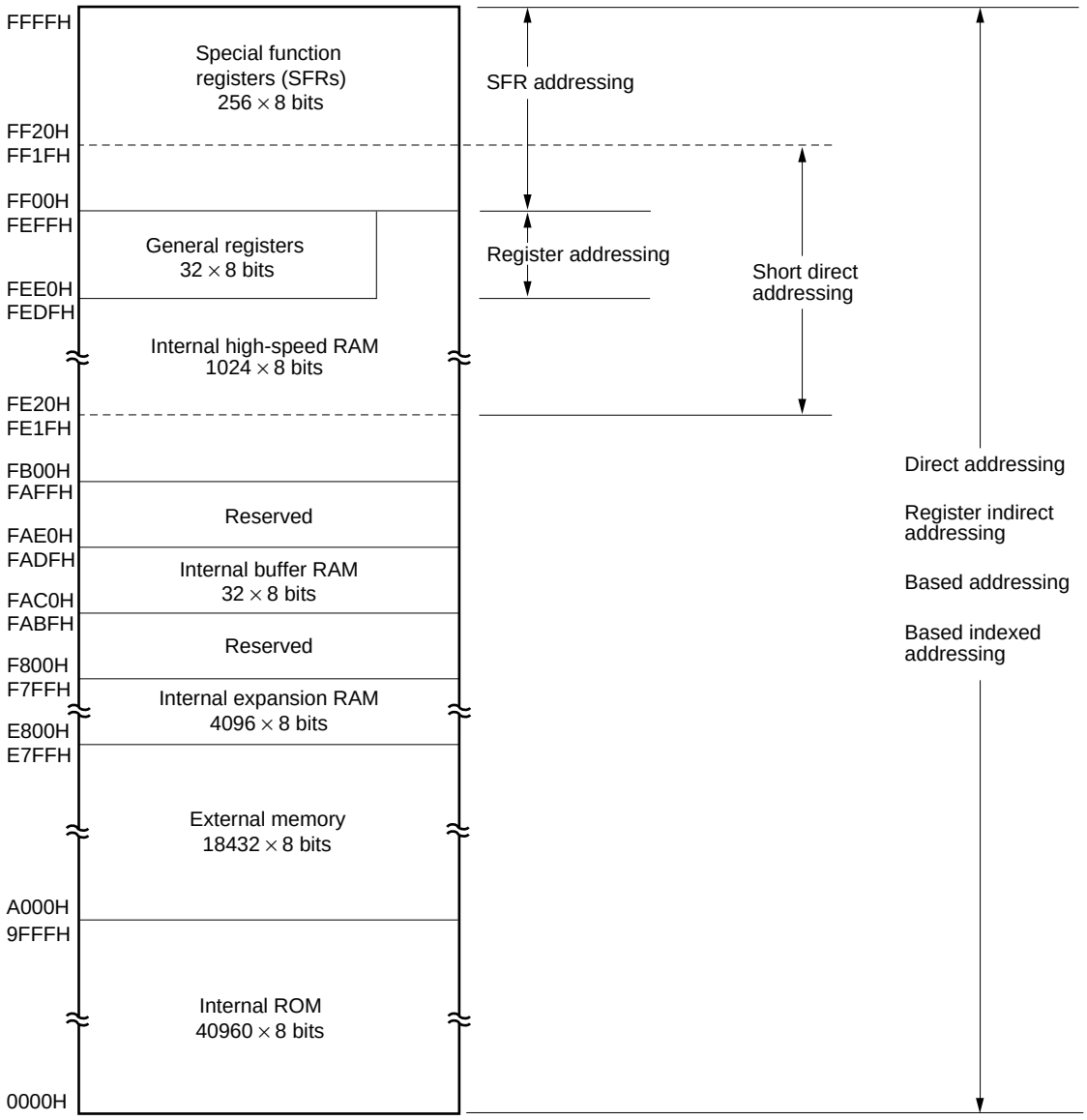
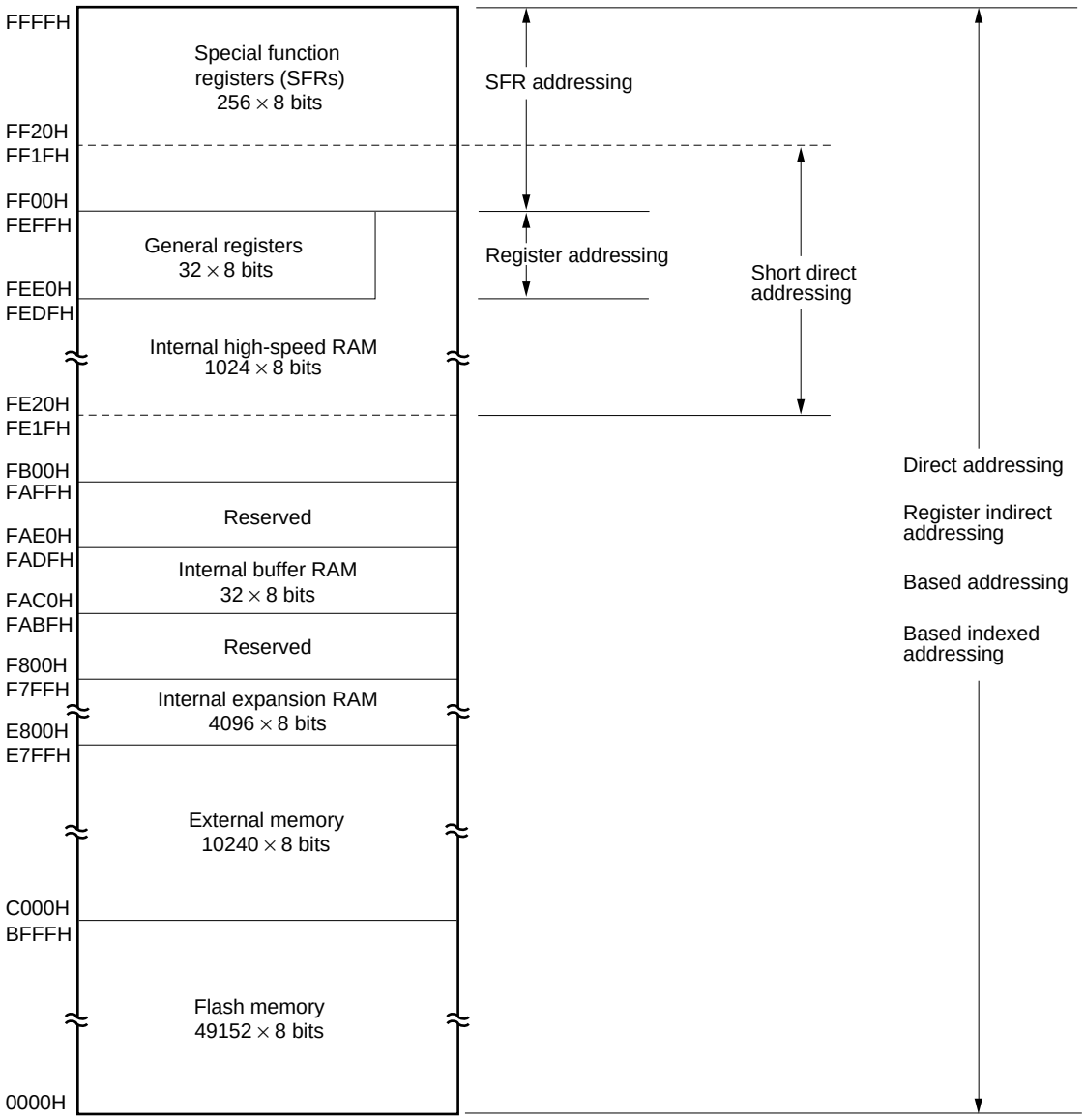


Figure 3-4. Data Memory Addressing (μPD78F0066)



3.2 Processor Registers

The μPD780065 Subseries products incorporate the following processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

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

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

Figure 3-5. 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 3-6. Program Status Word Format

	7							0
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY

(a) Interrupt enable flag (IE)

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

When 0, the IE is set to the disable interrupt (DI) state, and only non-maskable interrupt request becomes acknowledgeable. Other interrupt requests are all disabled.

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

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

(b) Zero flag (Z)

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

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

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

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

(d) Auxiliary carry flag (AC)

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

(e) In-service priority flag (ISP)

This flag manages the priority of acknowledgeable maskable vectored interrupts. When this flag is 0, low-level vectored interrupts request specified with a priority specify flag register (PR0L, PR0H, PR1L) (refer to **17.3 (3) Priority Specify Flag Register (PR0L, PR0H, PR1L)**) are disabled for acknowledgement. When it is 1, all interrupts are acknowledgeable. Actual request acknowledgement is controlled with the interrupt enable flag (IE).

(f) Carry flag (CY)

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

(3) Stack pointer (SP)

This is a 16-bit register to hold the start address of the memory stack area. Only the internal high-speed RAM area FB00H to FEFFH can be set as the stack area.

Figure 3-7. 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 3-8 and 3-9.

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

Figure 3-8. Data to be Saved to Stack Memory

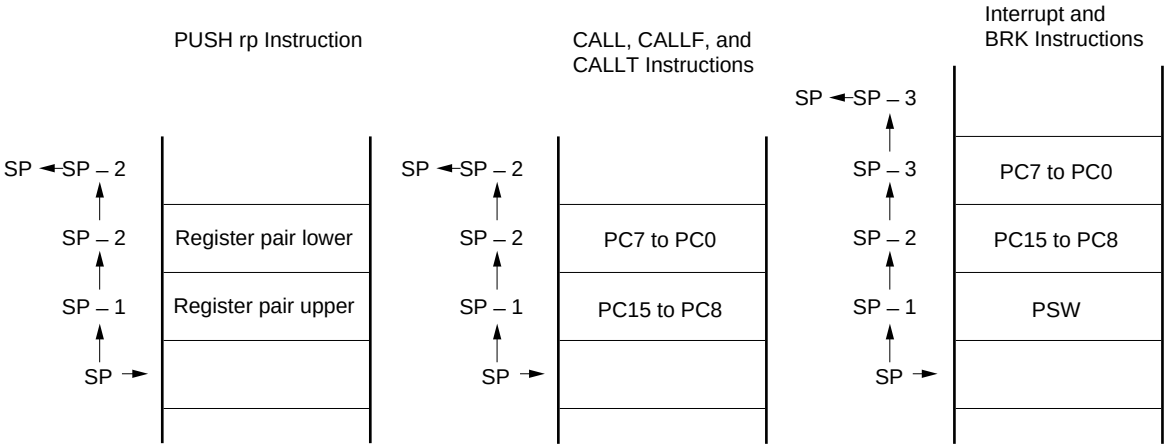
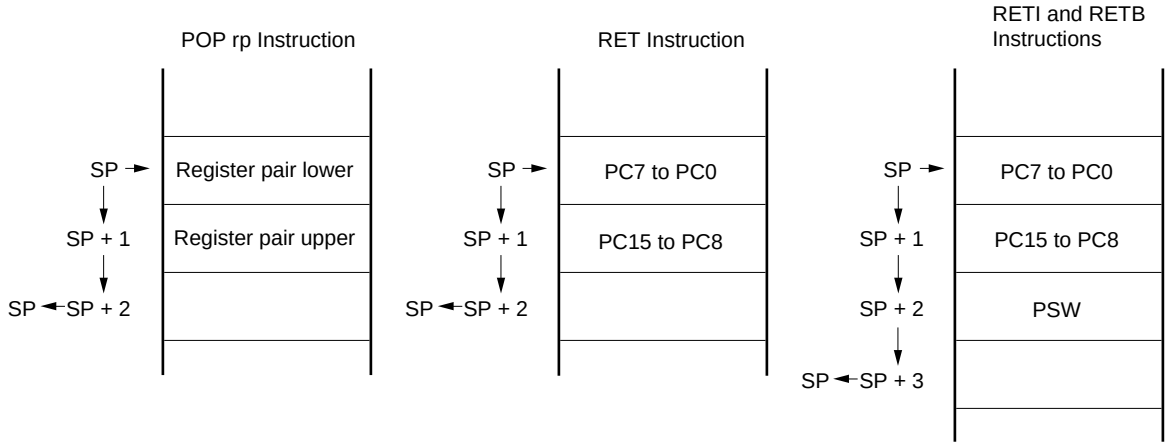


Figure 3-9. Data to be Restored from Stack Memory



3.2.2 General registers

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

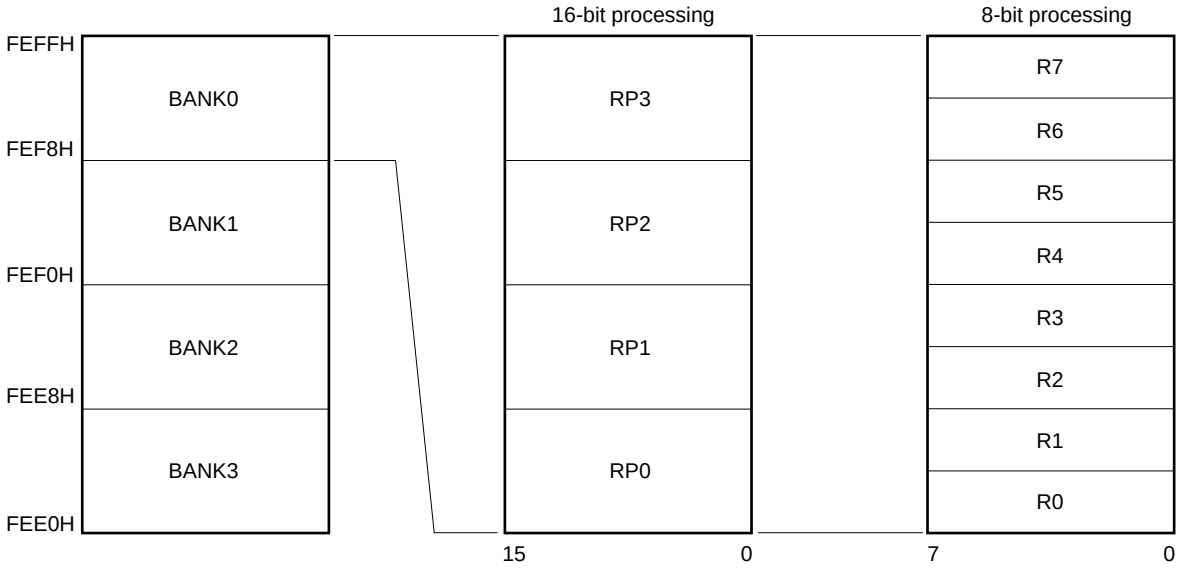
Each register can also be used as an 8-bit register. Two 8-bit registers can be used in pairs as a 16-bit register (AX, BC, DE, and HL).

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

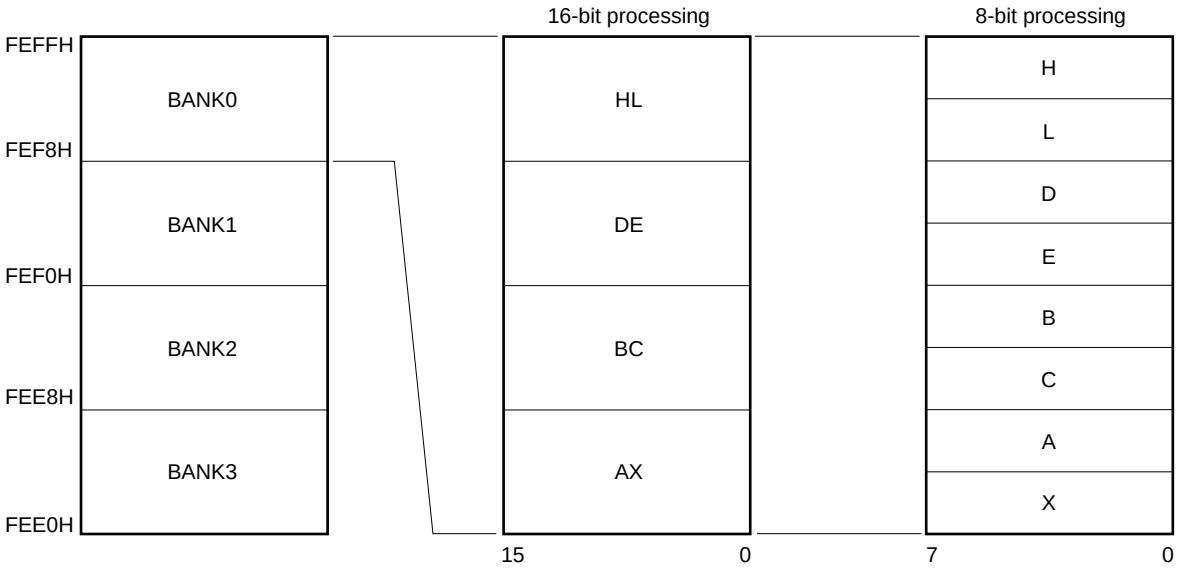
Register banks to be used for instruction execution are set with the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

Figure 3-10. General Register Configuration

(a) Absolute name



(b) Function name



3.2.3 Special Function Register (SFR)

Unlike a general register, each special-function register has special functions.

It is allocated in the FF00H to FFFFH area.

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

Each manipulation bit unit can be specified as follows.

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

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

- Symbol
  Symbol indicating the address of a special function register. It is a reserved word in the RA78K/0, and is defined via the header file “sfrbit.h” in the CC78K/0. When using the RA78K/0, ID78K0-NS, ID78K0, or SM78K0, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding special-function register can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulatable bit units
  Indicates the manipulatable bit unit (1, 8, or 16). “-” indicates a bit unit for which manipulation is not possible.
- 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	Port0	P0	R/W	√	√	—	00H
FF02H	Port2	P2		√	√	—	
FF03H	Port3	P3		√	√	—	
FF04H	Port4	P4		√	√	—	Undefined
FF05H	Port5	P5		√	√	—	
FF06H	Port6	P6		√	√	—	
FF07H	Port7	P7		√	√	—	00H
FF08H	Port8	P8		√	√	—	
FF09H	Port9	P9		√	√	—	
FF0AH	Capture/compare register 00	CR00		—	—	√	Undefined
FF0BH							
FF0CH	Capture/compare register 01	CR01		—	—	√	
FF0DH							
FF0EH	16-bit timer/counter 0	TM0	R	—	—	√	0000H
FF0FH							
FF10H	8-bit compare register 50	CR50	R/W	—	√	—	Undefined
FF11H	8-bit compare register 51	CR51		—	√	—	
FF12H	8-bit counter 50	TM5 TM50	R	—	√	√	00H
FF13H	8-bit counter 51	TM51		—	√		
FF15H	A/D conversion result register	ADCR0		—	√	—	
FF17H	Serial I/O shift register 1	SIO1	R/W	—	√	—	Undefined
FF18H	Serial I/O shift register 30	SIO30		—	√	—	
FF19H	Serial I/O shift register 31	SIO31		—	√	—	
FF1AH	Transmit shift register	TXS0	W	—	√	—	FFH
	Receive buffer register	RXB0	R	—	√	—	
FF20H	Port mode register 0	PM0	R/W	√	√	—	
FF22H	Port mode register 2	PM2		√	√	—	
FF23H	Port mode register 3	PM3		√	√	—	
FF24H	Port mode register 4	PM4		√	√	—	
FF25H	Port mode register 5	PM5		√	√	—	
FF26H	Port mode register 6	PM6		√	√	—	
FF27H	Port mode register 7	PM7		√	√	—	
FF28H	Port mode register 8	PM8		√	√	—	
FF29H	Port mode register 9	PM9		√	√	—	

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	
	FF30H	Pull-up resistor option register 0	PU0	R/W	√	√	—	00H
	FF32H	Pull-up resistor option register 2	PU2		√	√	—	
	FF33H	Pull-up resistor option register 3	PU3		√	√	—	
	FF34H	Pull-up resistor option register 4	PU4		√	√	—	
	FF35H	Pull-up resistor option register 5	PU5		√	√	—	
	FF36H	Pull-up resistor option register 6	PU6		√	√	—	
	FF37H	Pull-up resistor option register 7	PU7		√	√	—	
	FF38H	Pull-up resistor option register 8	PU8		√	√	—	
	FF39H	Pull-up resistor option register 9	PU9		√	√	—	
	FF40H	Clock output selection register	CKS		√	√	—	
	FF41H	Watch timer mode control register	WTM		√	√	—	
	FF42H	Watchdog timer clock selection register	WDCS		—	√	—	
	FF47H	Memory expansion mode register	MEM		√	√	—	
	FF48H	External interrupt rising edge enable register	EGP		√	√	—	
	FF49H	External interrupt falling edge enable register	EGN		√	√	—	
	FF60H	16-bit timer mode control register	TMC0		√	√	—	
	FF61H	Prescaler mode register	PRM0		—	√	—	
	FF62H	Capture/compare control register 0	CRC0		√	√	—	
	FF63H	16-bit timer output control register 0	TOC0		√	√	—	
	FF68H	Serial operation mode register 1	CSIM1		√	√	—	
★	FF69H	Automatic data transmission/reception control register	ADTC0		√	√	—	
★	FF6AH	Automatic data transmission/reception address pointer	ADTP0		—	√	—	
★	FF6BH	Automatic data transmission/reception interval specification register	ADTI0		√	√	—	
	FF70H	8-bit timer mode control register 50	TMC50		√	√	—	04H
	FF71H	Timer clock selection register 50	TCL50		—	√	—	00H
	FF74H	8-bit timer mode control register 51	TMC51		√	√	—	04H
	FF75H	Timer clock selection register 51	TCL51		—	√	—	00H
	FF80H	A/D converter mode register	ADM0		√	√	—	
	FF81H	Analog input channel specification register	ADS0		—	√	—	
	FFA0H	Asynchronous serial interface mode register	ASIM0		√	√	—	
	FFA1H	Asynchronous serial interface status register	ASIS0	R	—	√	—	
	FFA2H	Baud rate generator control register	BRGC0	R/W	—	√	—	

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	
FFB0H	Serial operation mode register 30	CSIM30		R/W	√	√	—	00H
FFB1H	Serial operation mode register 31	CSIM31			√	√	—	
FFD0H ↓ FFDFH	External access area[Note 1]				√	√	—	Undefined
FFE0H	Interrupt request flag register 0L	IF0	IF0L		√	√	√	00H
FFE1H	Interrupt request flag register 0H		IF0H		√	√		
FFE2H	Interrupt request flag register 1L	IF1L			√	√	—	
FFE4H	Interrupt mask flag register 0L	MK0	MK0L		√	√	√	FFH
FFE5H	Interrupt mask flag register 0H		MK0H		√	√		
FFE6H	Interrupt mask flag register 1L	MK1L			√	√	—	
FFE8H	Priority level specification flag register 0L	PR0	PR0L		√	√	√	
FFE9H	Priority level specification flag register 0H		PR0H		√	√		
FFEAH	Priority level specification flag register 1L	PR1L			√	√	—	
FFF0H	Memory size switching register	IMS			—	√	—	CFH[Note 2]
FFF4H	Internal expansion RAM size switching register	IXS			√	√	—	0CH[Note 3]
FFF8H	Memory expansion wait setting register	MM			√	√	—	10H
FFF9H	Watchdog timer mode register	WDTM			√	√	—	00H
FFFAH	Oscillation stabilization time selection register	OSTS			—	√	—	04H
FFFBH	Processor clock control register	PCC			√	√	—	

Notes 1. The external access area cannot be accessed by SFR addressing. Access it with the direct addressing method.

2. The default is CFH, but be sure to set the value corresponding to each respective product as indicated below.

μPD780065: CAH

μPD78F0066: CCH or value for mask ROM version

3. The default is 0CH, but always set 04H.

3.3 Instruction Address Addressing

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

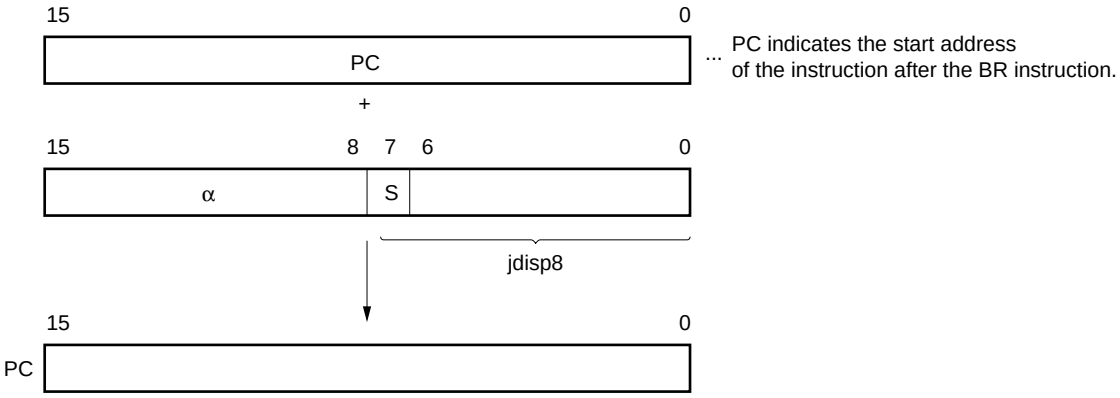
3.3.1 Relative addressing

[Function]

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

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

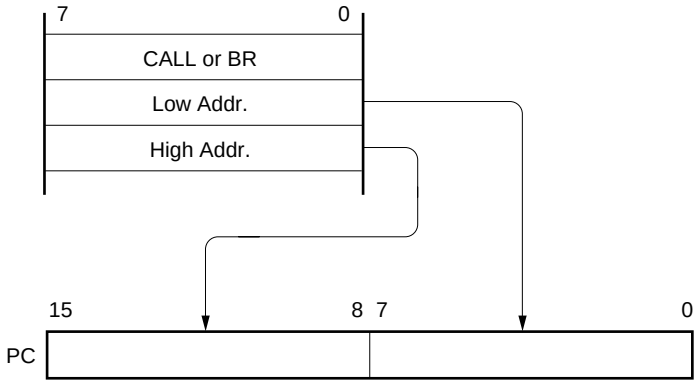
[Function]

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

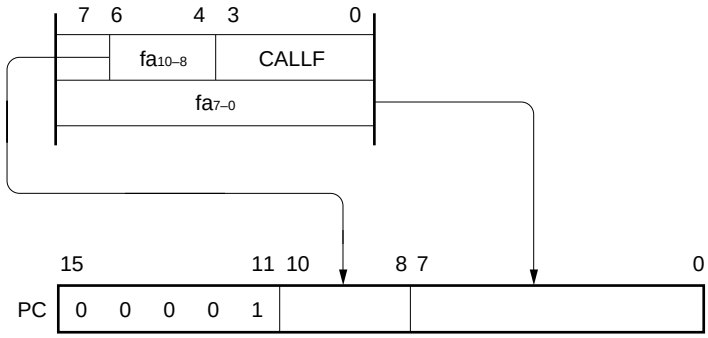
This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed. CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space. The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

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



In the case of CALLF !addr11 instruction



3.3.3 Table indirect addressing

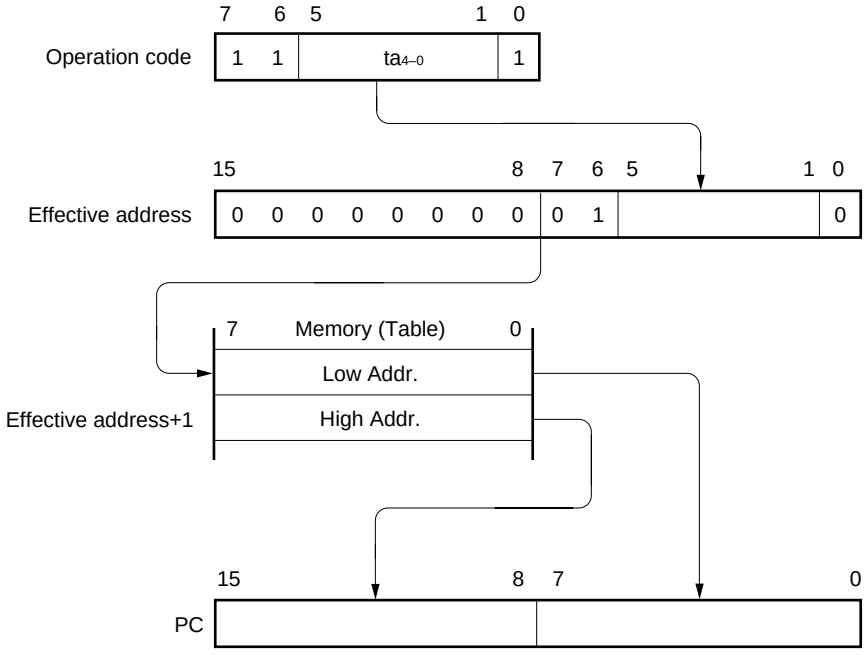
[Function]

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

This function is carried out when the CALLT [addr5] instruction is executed.

This instruction references the address stored in the memory table from 40H to 7FH, and allows branching to the entire memory space.

[Illustration]



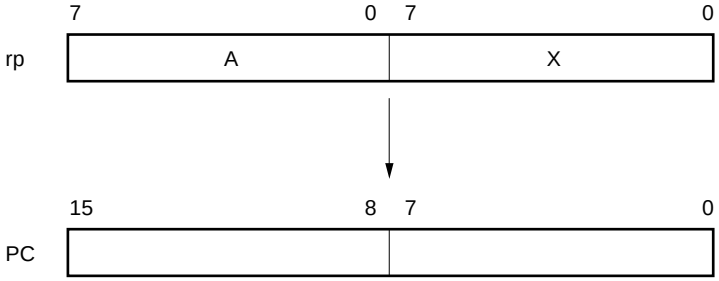
3.3.4 Register addressing

[Function]

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

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

[Illustration]



3.4 Operand Address Addressing

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

3.4.1 Implied addressing

[Function]

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

Of the μPD780065 Subseries instruction words, the following instructions employ implied addressing.

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

[Operand format]

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

[Description example]

In the case of MULU X

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

3.4.2 Register addressing

[Function]

The general register to be specified is accessed as an operand with the register specify code (Rn and RPn) of an instruction word in the registered bank specified with the register bank select flag (RBS0 to RBS1). Register addressing is carried out when an instruction with the following operand format is executed. When an 8-bit register is specified, one of the eight registers is specified with 3 bits in the operation code.

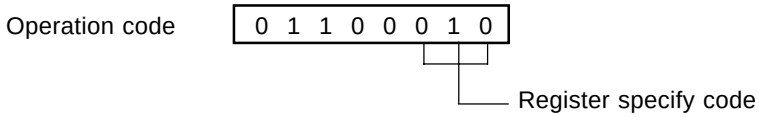
[Operand format]

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

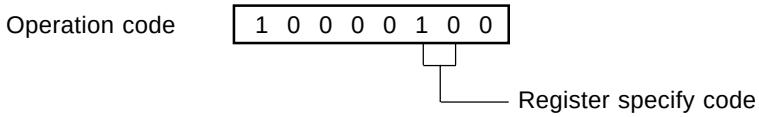
'r' and 'rp' can be described with absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[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 to be manipulated is addressed with immediate data in an instruction word becoming an operand address.

[Operand format]

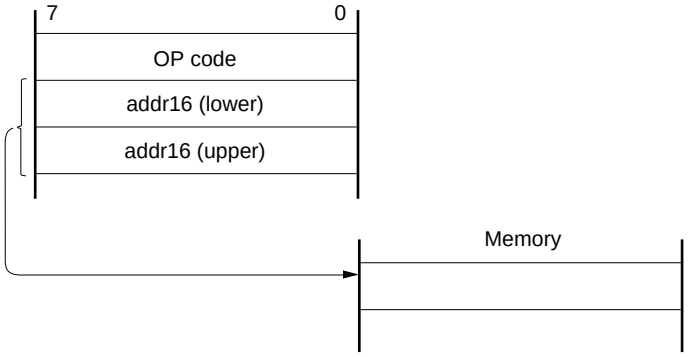
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

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

[Illustration]



3.4.4 Short direct addressing

[Function]

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

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

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

[Operand format]

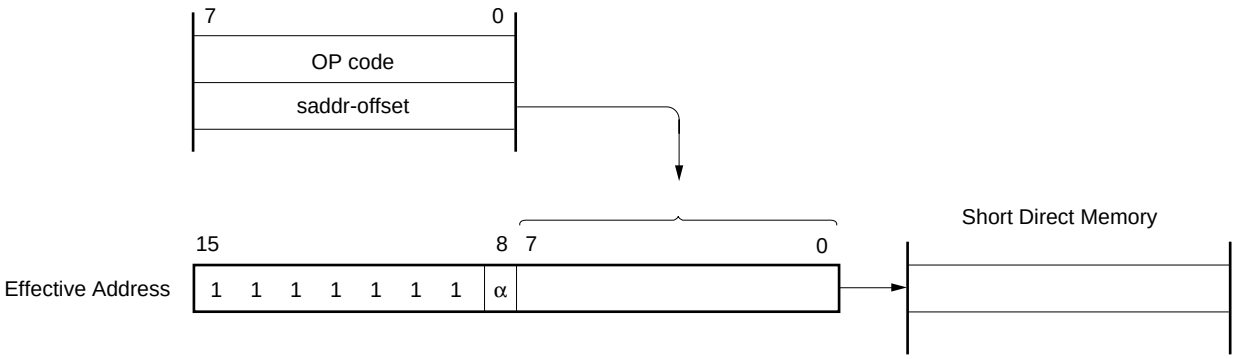
Identifier	Description
saddr	Label of FE20H to FF1FH immediate data
saddrp	Label of FE20H to FF1FH immediate data (even address only)

[Description example]

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

Operation code	0 0 0 1 0 0 0 1	OP code
	0 0 1 1 0 0 0 0	30H (saddr-offset)
	0 1 0 1 0 0 0 0	50H (immediate data)

[Illustration]



When 8-bit immediate data is 20H to FFH, α = 0
When 8-bit immediate data is 00H to 1FH, α = 1

3.4.5 Special-function register (SFR) addressing

[Function]

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

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

[Operand format]

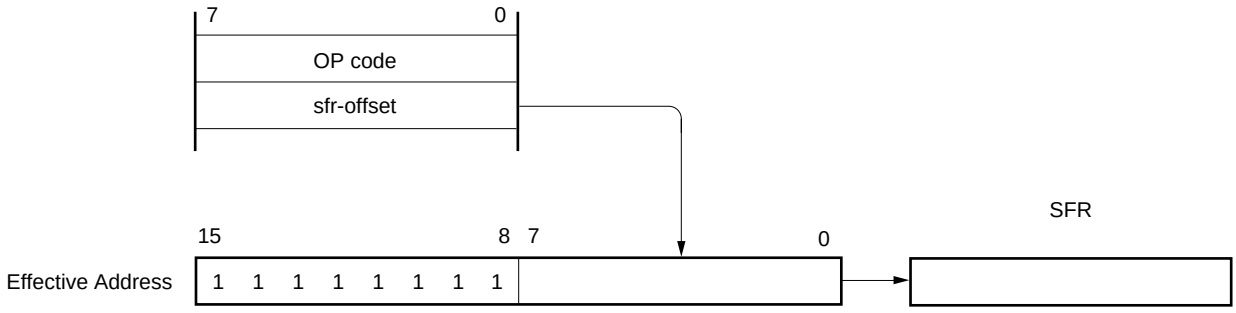
Identifier	Description
sfr	Special-function register name
sfrp	16-bit manipulatable special-function register name (even address only)

[Description example]

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

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

[Illustration]



3.4.6 Register indirect addressing

[Function]

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

[Operand format]

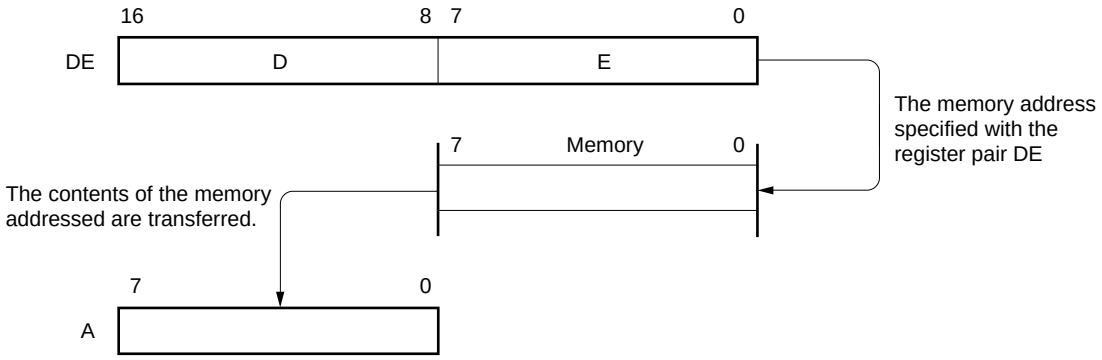
Identifier	Description
—	[DE], [HL]

[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 as offset data to the contents of the base register, that is, the HL register pair in an instruction word of the register bank specified with the register bank select flag (RBS0 and RBS1) and the sum is used to address the memory. 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 are added to the contents of the base register, that is, the HL register pair in an instruction word of the register bank specified with the register bank select flag (RBS0 and RBS1) and the sum is used to address the memory.

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 + B], [HL + C]

[Description example]

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

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

3.4.9 Stack addressing

[Function]

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

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

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

[Description example]

In the case of PUSH DE

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

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The μPD780065 Subseries products incorporate sixty input/output ports. Figure 4-1 shows the port configuration. Every port is capable of 1-bit and 8-bit manipulations and can carry out considerably varied control operations. Besides port functions, the ports can also serve as on-chip hardware input/output pins.

Figure 4-1. Port Types

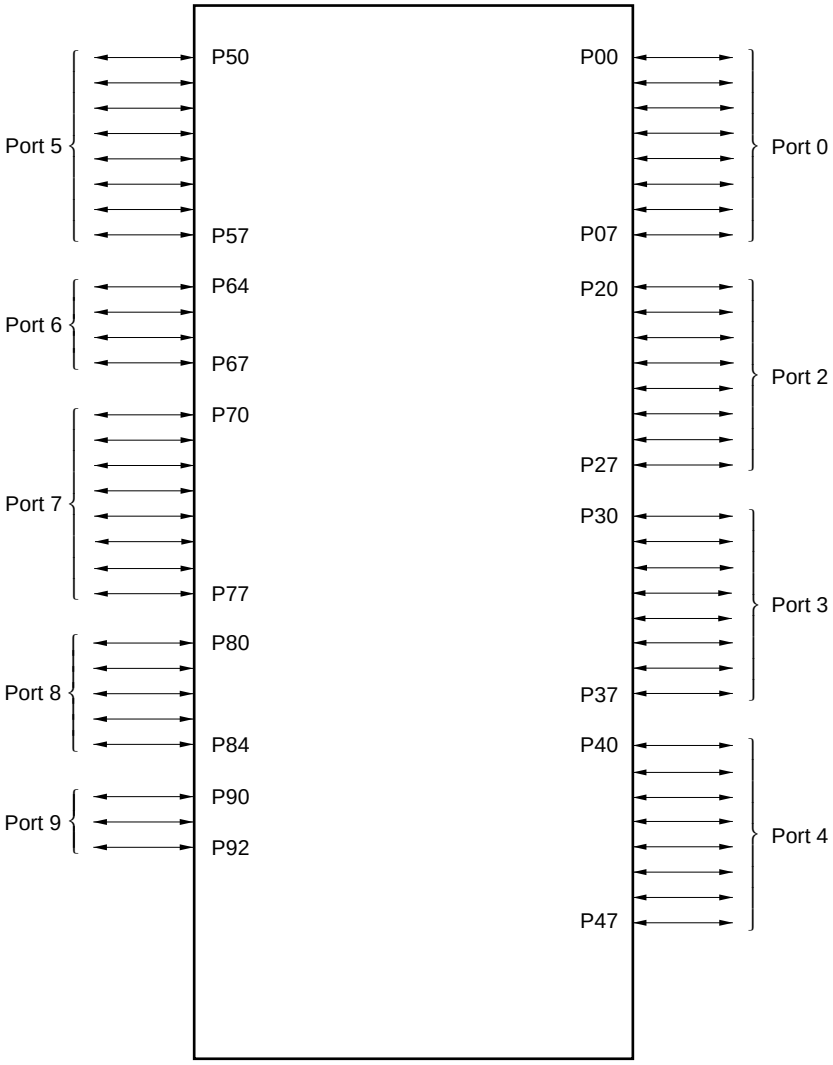


Table 4-1. Port Functions

Pin Name	Function	Alternate Function
P00 to P03	Port 0 8-bit input/output port. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by means of software.	INTP0 to INTP3
P04 to P07		—
P20	Port 2 8-bit input/output port. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by means of software.	TI00/TO0
P21		TI01
P22		TI50/TO50
P23		TI51/TO51
P24 to P27		—
P30 to P37	Port 3 8-bit input/output port. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by means of software.	—
P40 to P47	Port 4 8-bit input/output port. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by means of software.	AD0 to AD7
P50 to P57	Port 5 8-bit input/output port. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by means of software.	A8 to A15
P64	Port 6 4-bit input/output port. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by means of software.	$\overline{RD}$
P65		$\overline{WR}$
P66		$\overline{WAIT}$
P67		ASTB
P70	Port 7 8-bit input/output port. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by means of software.	PCL
P71		ASCK0
P72		TxD0
P73		RxD0
P74		$\overline{SCK30}$
P75		SDIO30
P76, P77		—
P80	Port 8 5-bit input/output port. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by means of software.	STB
P81		BUSY
P82		$\overline{SCK1}$
P83		SO1
P84		SI1
P90	Port 9 3-bit input/output port. Input/output mode can be specified in 1-bit units. An on-chip pull-up resistor can be specified by means of software.	$\overline{SCK31}$
P91		SO31
P92		SI31

4.2 Port Configuration

A port consists of the following hardware:

Table 4-2. Port Configuration

Item	Configuration
Control register	Port mode register (PMm: m = 0, 2 to 9) Pull-up resistor option register (PUm: m = 0, 2 to 9)
Port	Input/output: 60
Pull-up resistor	Software control: 60

4.2.1 Port 0

Port 0 is an 8-bit input/output port with output latch. Input mode/output mode can be specified for pins P00 to P07 in 1-bit units with port mode register 0 (PM0). For pins P00 to P07, an on-chip pull-up resistor can be specified in 6-bit units with pull-up resistor option register 0 (PU0).

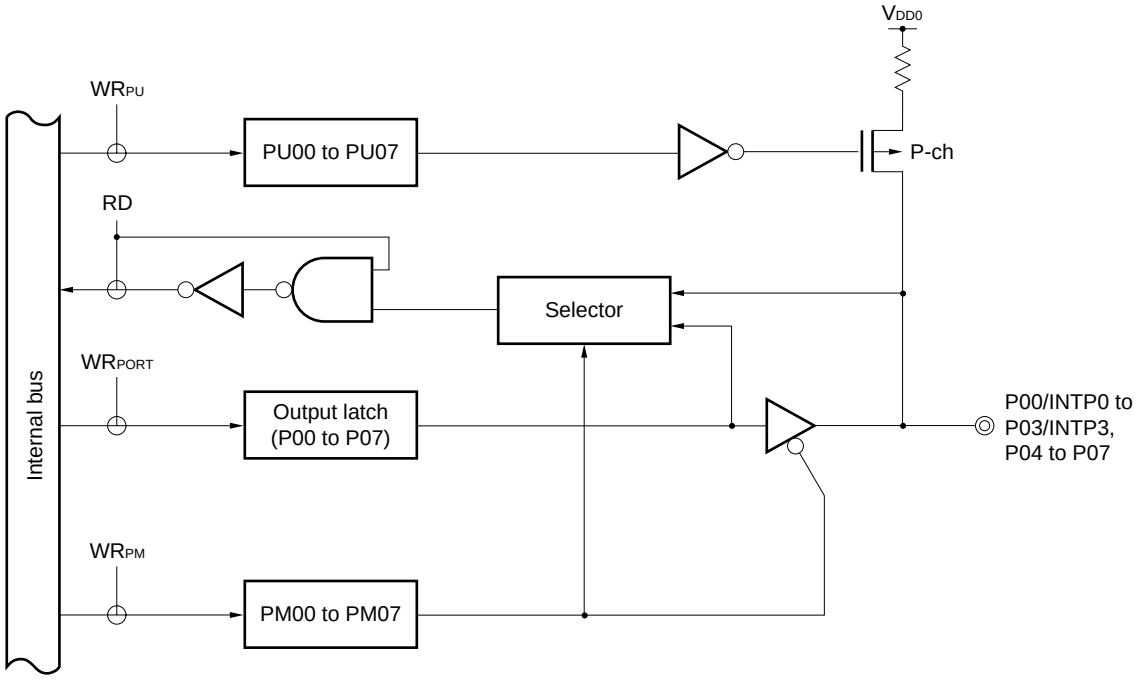
This port can also be used as an external interrupt request input.

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

Figure 4-2 shows a block diagram of port 0.

Caution Because port 0 also serves as an 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 to P07**



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

4.2.2 Port 2

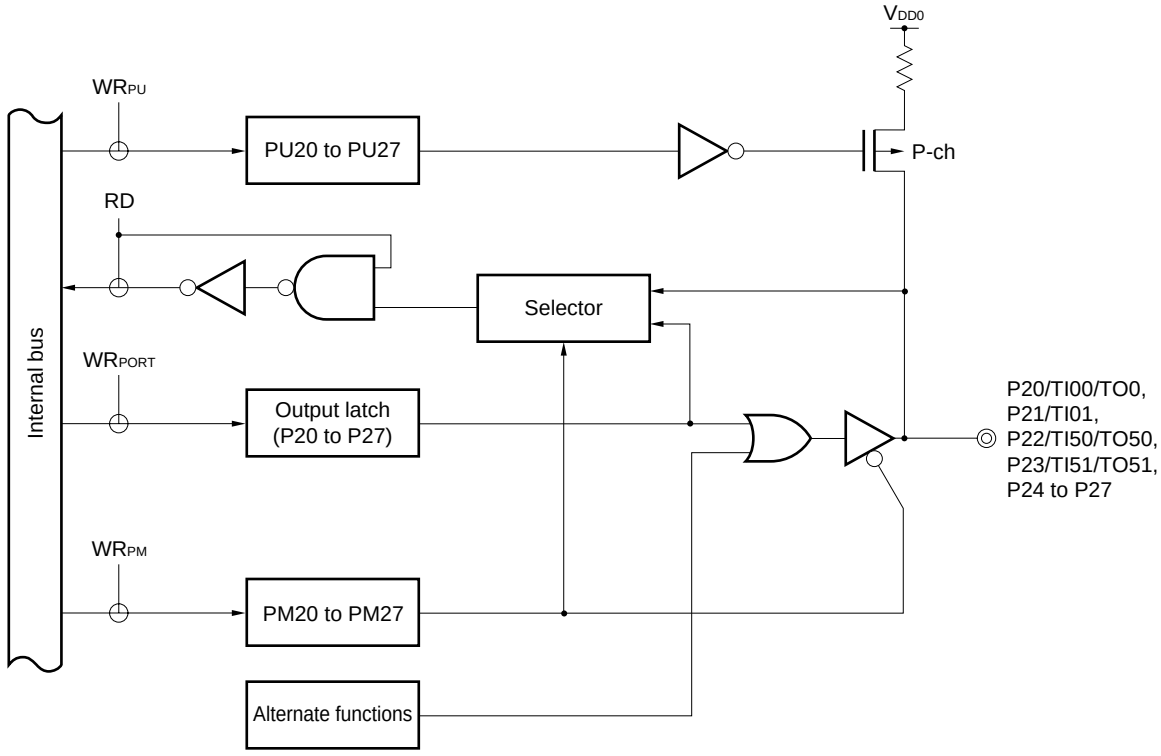
Port 2 is an 8-bit input/output port with output latch. Input mode/output mode can be specified for pins P20 to P27 in 1-bit units with port mode register 2 (PM2). For pins P20 to P27, an on-chip pull-up resistor can be specified in 1-bit units with pull-up resistor option register 2 (PU2).

This port can also be used as the timer input/output.

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

Figure 4-3 shows a block diagram of port 2.

★ **Figure 4-3. Block Diagram of P20 to P27**



PU: Pull-up resistor option register

PM: Port mode register

RD: Port 2 read signal

WR: Port 2 write signal

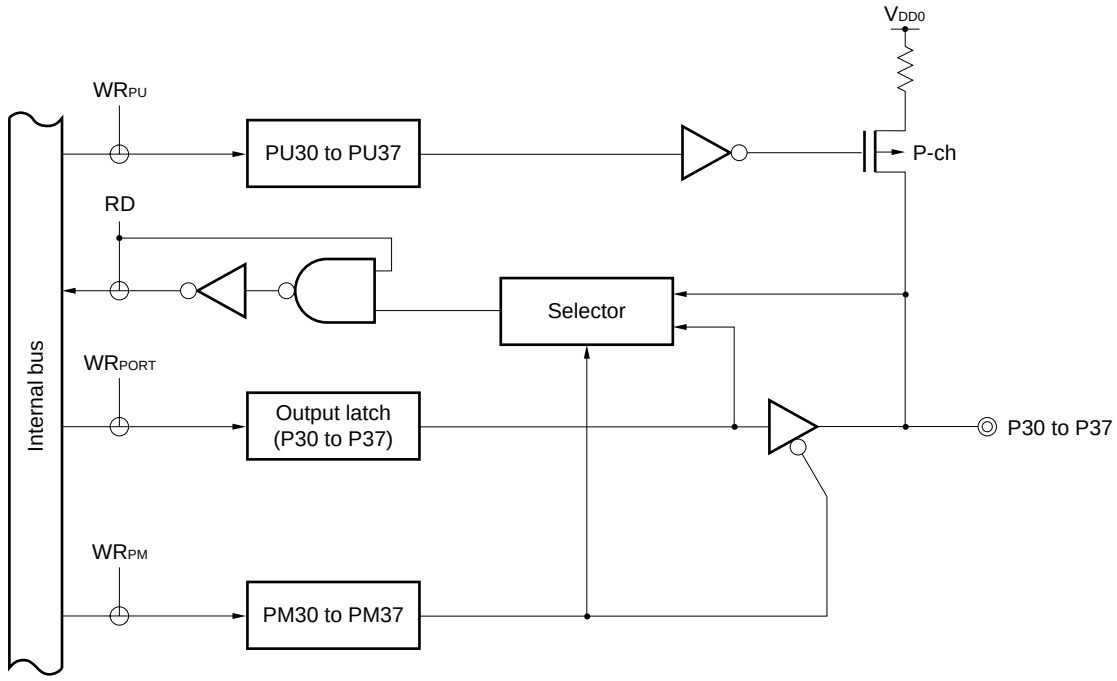
4.2.3 Port 3

Port 3 is an 8-bit input/output port with output latch. Input mode/output mode can be specified for pins P30 to P37 in 1-bit units with port mode register 3 (PM3). For pins P30 to P37, an on-chip pull-up resistor can be specified in 1-bit units with pull-up resistor option register 3 (PU3).

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

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

★ **Figure 4-4. Block Diagram of P30 to P37**



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

4.2.4 Port 4

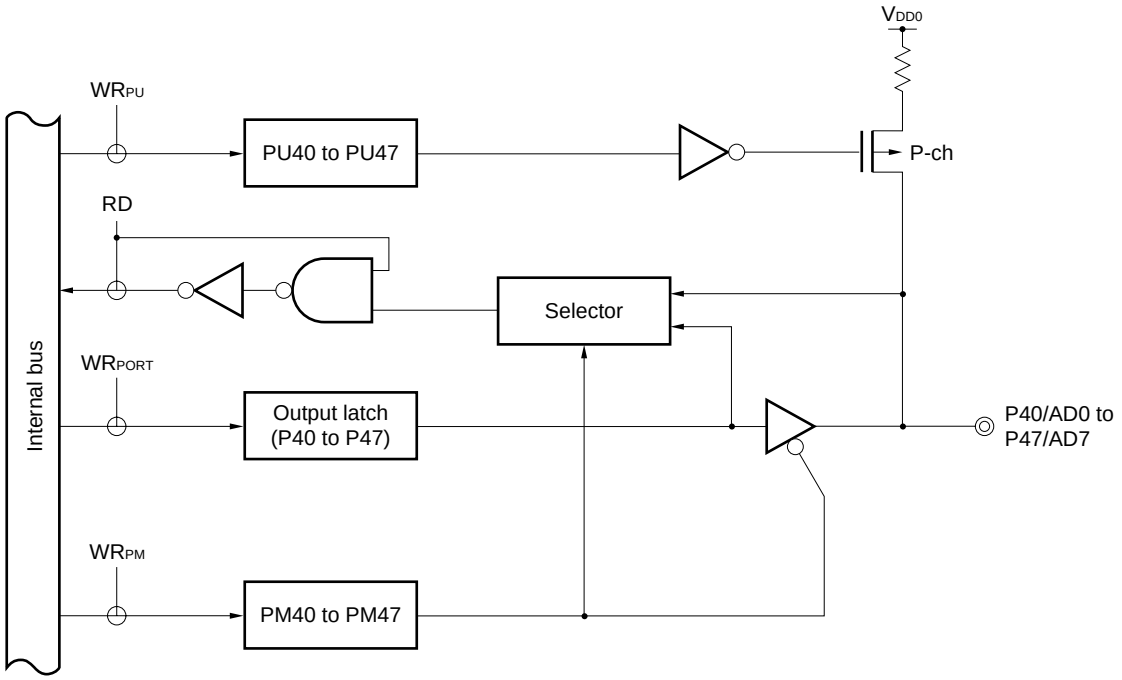
Port 4 is an 8-bit input/output port with output latch. Input mode/output mode can be specified for pins P40 to P47 in 1-bit units with port mode register 4 (PM4). For pins P40 to P47, a pull-up resistor can be specified in 1-bit units with pull-up resistor option register 4 (PU4).

This port can also be used as an address/data bus in external memory expansion mode.

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

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

★ **Figure 4-5. Block Diagram of P40 to P47**



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

4.2.5 Port 5

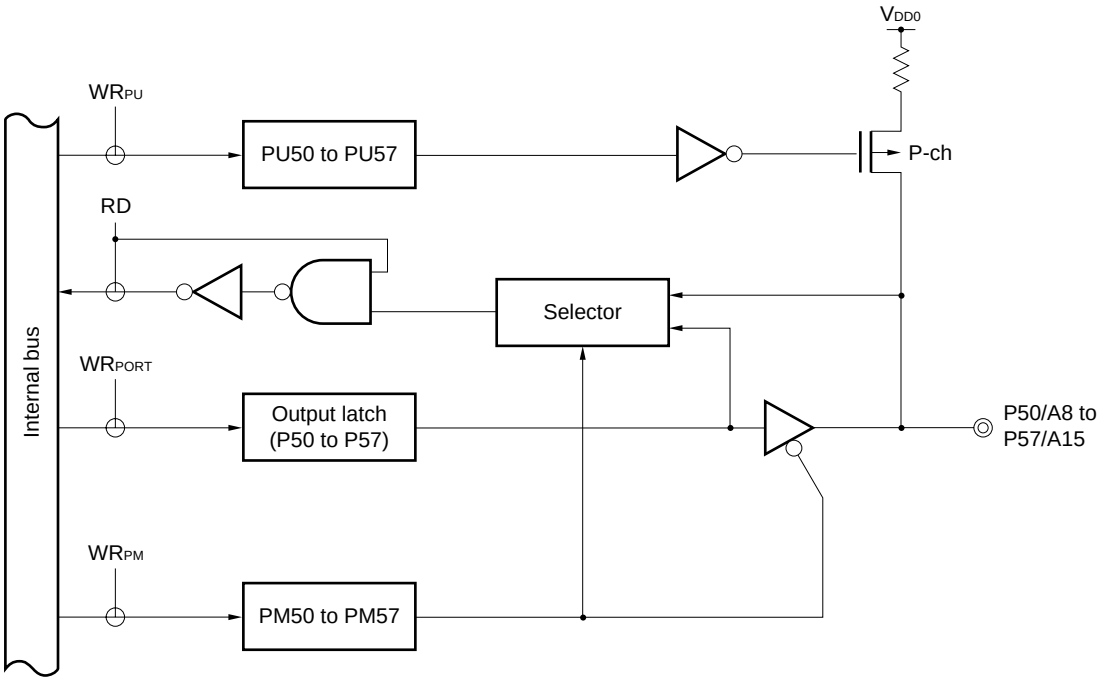
Port 5 is an 8-bit input/output port with output latch. Input mode/output mode can be specified for pins P50 to P57 in 1-bit units with port mode register 5 (PM5). For pins P50 to P57, an on-chip pull-up resistor can be specified in 1-bit units with pull-up resistor option register 5 (PU5).

This port can also be used as an address bus in external memory expansion mode.

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

Figure 4-6 shows a block diagram of port 5.

★ **Figure 4-6. Block Diagram of P50 to P57**



PU: Pull-up resistor option register
PM: Port mode register
RD: Port 5 read signal
WR: Port 5 write signal

4.2.6 Port 6

Port 6 is a 4-bit input/output port with output latch. Input mode/output mode can be specified for pins P64 to P67 in 1-bit units with port mode register 6 (PM6). For pins P64 to P67, an on-chip pull-up resistor can be specified in 1-bit units with pull-up resistor option register 6 (PU6).

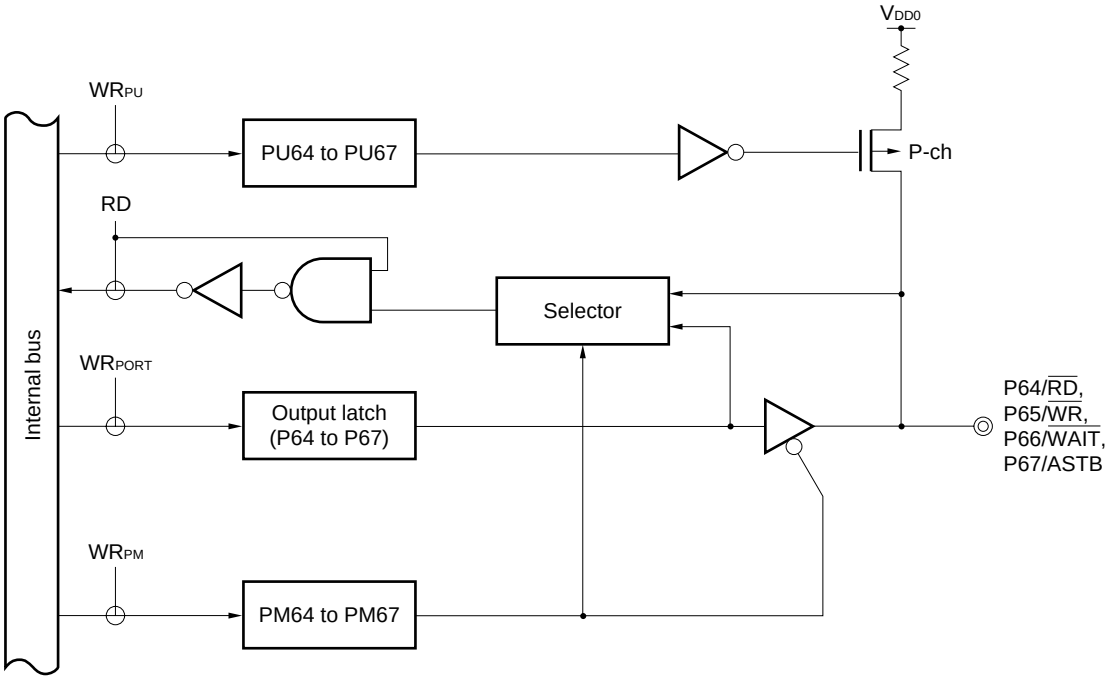
This port can also be used as a control signal output in external memory expansion mode.

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

Figure 4-7 shows a block diagram of port 6.

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

★ **Figure 4-7. Block Diagram of P64 to P67**



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

4.2.7 Port 7

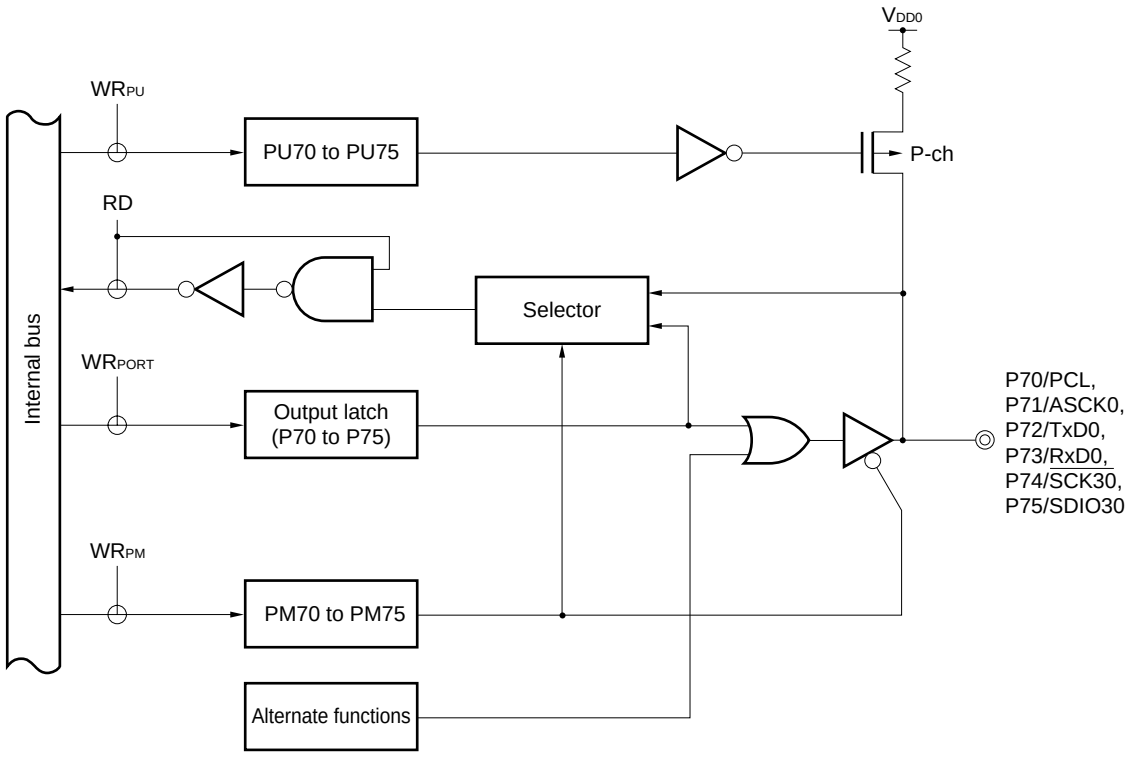
This is an 8-bit input/output port with output latch. Input mode/output mode can be specified for pins P70 to P77 in 1-bit units by means of port mode register 7 (PM7). For pins P70 to P77, an on-chip pull-up resistor can be specified in 1-bit units with pull-up resistor option register 7 (PU7).

This port can also be used as the serial interface serial data input/output, serial clock input/output, and clock output.

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

Figures 4-8 and 4-9 show block diagrams of port 7.

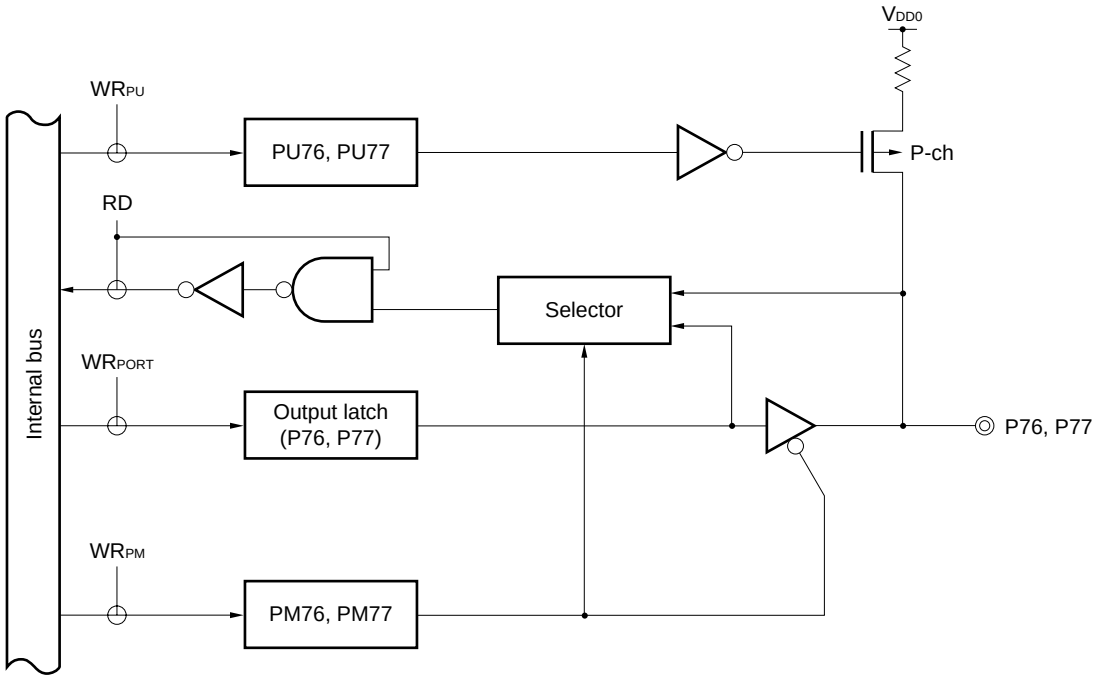
★ **Figure 4-8. Block Diagram of P70 to P75**



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

★

Figure 4-9. Block Diagram of P76 and P77



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

4.2.8 Port 8

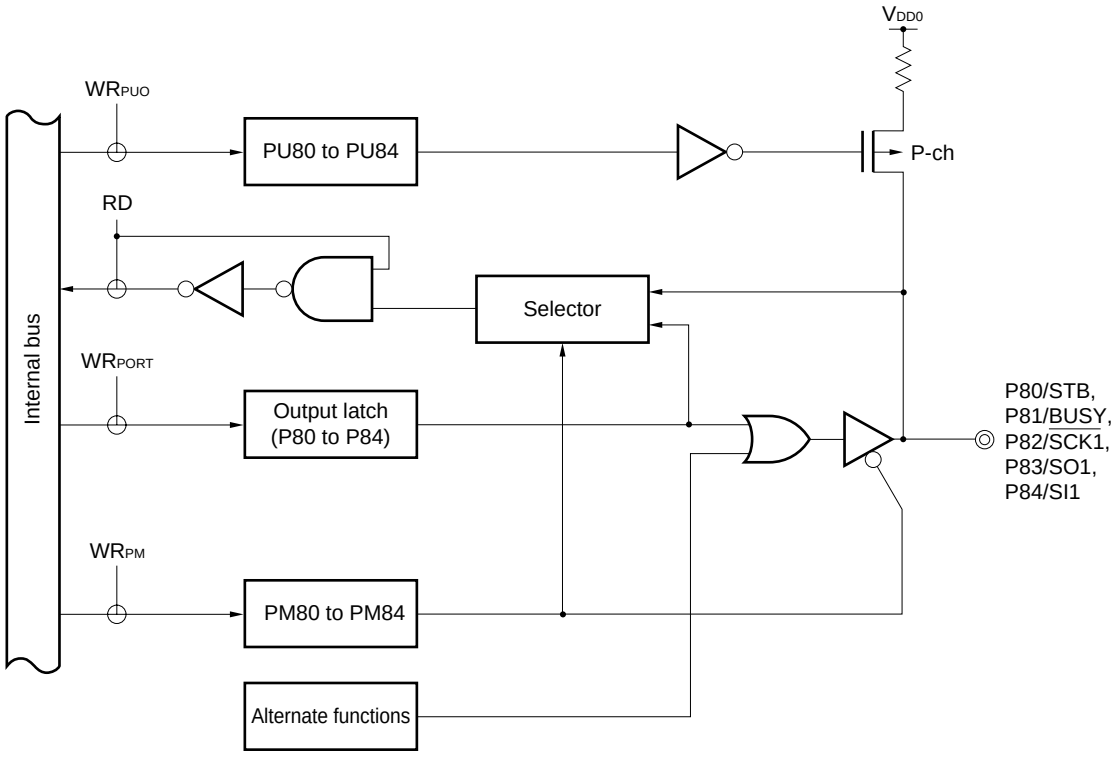
This is a 5-bit input/output port with output latch. Input mode/output mode can be specified for pins P80 to P84 in 1-bit units with port mode register 8 (PM8). For pins P80 to P84, an on-chip pull-up resistor can be specified in 1-bit units by means of pull-up resistor option register 8 (PU8).

This port can also be used as the serial interface serial data input/output, serial clock input/output, automatic transmission/reception busy input, and strobe output.

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

Figure 4-10 shows a block diagram of port 8.

★ **Figure 4-10. Block Diagram of P80 to P84**



PU: Pull-up resistor option register
PM: Port mode register
RD: Port 8 read signal
WR: Port 8 write signal

4.2.9 Port 9

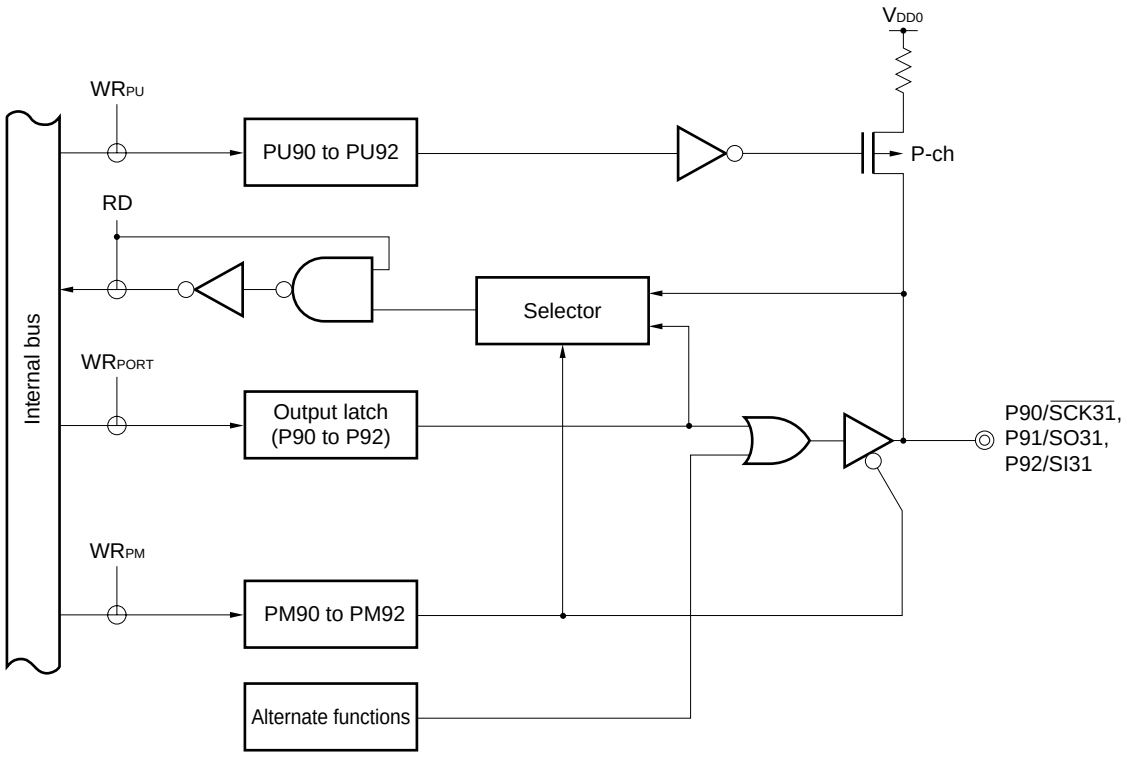
This is a 3-bit input/output port with output latches. Input mode/output mode can be specified for pins P90 to P92 in 1-bit units with port mode register 9 (PM9). For pins P90 to P92, an on-chip pull-up resistor can be specified in 1-bit units with pull-up resistor option register 9 (PU9).

This port can also be used as the serial interface serial data input/output, and serial clock input/output.

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

Figure 4-11 shows a block diagram of port 9.

★ **Figure 4-11. Block Diagram of P90 to P92**



PU: Pull-up resistor option register

PM: Port mode register

RD: Port 9 read signal

WR: Port 9 write signal

4.3 Port Function Control Registers

The following two types of registers control the ports.

- Port mode registers (PM0, PM2 to PM9)
- Pull-up resistor option register (PU0, PU2 to PU9)

(1) Port mode registers (PM0, PM2 to PM9)

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

PM0 and PM2 to PM9 are independently set with a 1-bit or 8-bit memory manipulation instruction.

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

Caution As port 0 has an alternate function as an external interrupt 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.

Figure 4-12. Format of Port Mode Registers (PM0, PM2 to PM9)

Address: FF20H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00

Address: FF22H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

Address: FF23H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

Address: FF24H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

Address: FF25H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

Address: FF26H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	PM67	PM66	PM65	PM64	1	1	1	1

Address: FF27H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

Address: FF28H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	1	1	1	PM84	PM83	PM82	PM81	PM80

Address: FF29H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM9	1	1	1	1	1	PM92	PM91	PM90

PMmn	Pmn pin input/output mode select (m = 0, 2 to 9: n = 0 to 7)
0	Output Mode (Output buffer on)
1	Input Mode (Output buffer off)

★ **(2) Pull-up resistor option registers (PU0, PU2 to PU9)**

This register is used to set whether to use an internal pull-up resistor at each port or not. An on-chip pull-up resistor for each port pin can be specified by setting PU0, and PU2 to PU9.

PU0 and PU2 to PU9 are set with a 1-bit or 8-bit memory manipulation instruction.

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

Caution When an on-chip pull-up resistor is used, even if output mode is set, the pull-up resistor will not be cut off.

To use in output mode, set the corresponding pull-up resistor option registers to 0.

Figure 4-13. Format of Pull-Up Resistor Option Registers (PU0, PU2 to PU9)

Address: FF30H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00

Address: FF32H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU2	PU27	PU26	PU25	PU24	PU23	PU22	PU21	PU20

Address: FF33H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU3	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30

Address: FF34H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40

Address: FF35H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU5	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50

Address: FF36H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU6	PU67	PU66	PU65	PU64	0	0	0	0

Address: FF37H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70

Address: FF38H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU8	0	0	0	PU84	PU83	PU82	PU81	PU80

Address: FF39H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU9	0	0	0	0	0	PU92	PU91	PU90

PUmn	Pmn pin internal pull-up resistor select (m = 0, 2 to 9: n = 0 to 7)
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 input/output port

(1) Output mode

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

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

(2) Input mode

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

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

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

4.4.2 Reading from input/output port

(1) Output mode

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

(2) Input mode

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

4.4.3 Operations on input/output port

(1) Output mode

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

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

(2) Input mode

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

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

CHAPTER 5 CLOCK GENERATOR

5.1 Clock Generator Functions

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

(1) Main system clock oscillator

This circuit oscillates at frequencies of 1 to 8.38 MHz. Oscillation can be stopped by executing the STOP instruction or setting the processor clock control register (PCC).

(2) Subsystem clock oscillator

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

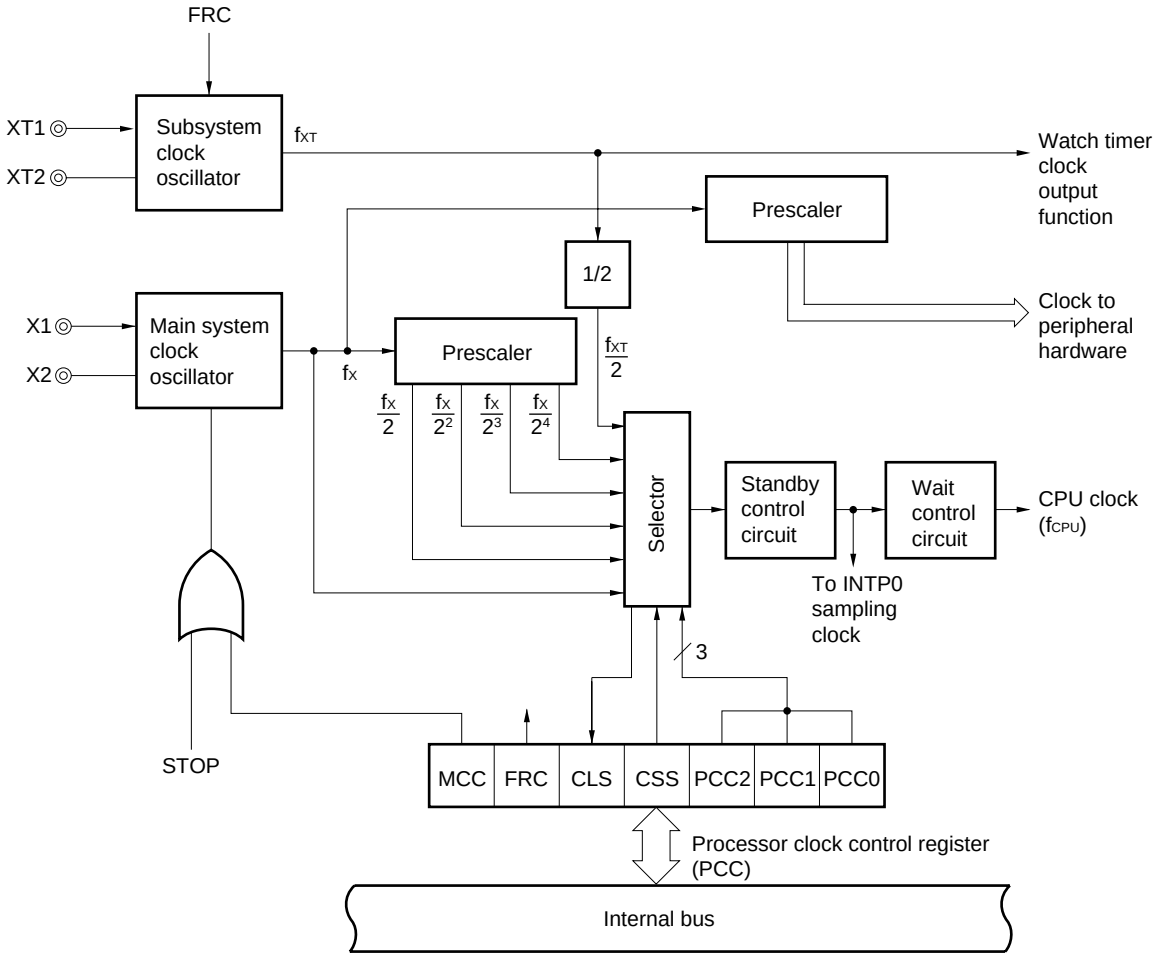
5.2 Clock Generator Configuration

The clock generator consists of the following hardware.

Table 5-1. Clock Generator Configuration

Item	Configuration
Control register	Processor clock control register (PCC)
Oscillator	Main system clock oscillator Subsystem clock oscillator

Figure 5-1. Clock Generator Block Diagram



5.3 Clock Generator Control Register

The clock generator is controlled by the processor clock control register (PCC).

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

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

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

Figure 5-2. Subsystem Clock Feedback Resistor

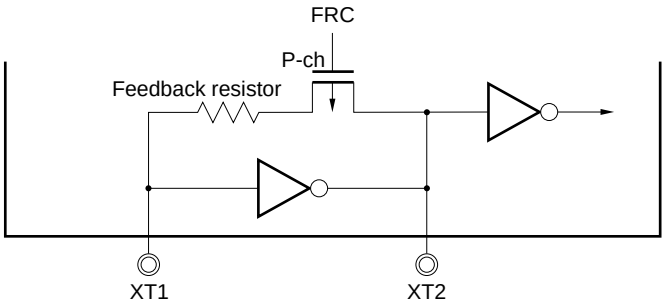


Figure 5-3. Format of Processor Clock Control Register (PCC)

Address: FFFBH After Reset: 04H R/W[Note 1]

Symbol	7	6	5	4	3	2	1	0
PCC	MCC	FRC	CLS	CSS	0	PCC2	PCC1	PCC0

MCC	Main system clock oscillation control [Note 2]
0	Oscillation possible
1	Oscillation stopped

FRC	Subsystem clock feedback resistor select
0	Internal feedback resistor used
1	Internal feedback resistor not used

CLS	CPU clock status
0	Main system clock
1	Subsystem clock

CSS	PCC2	PCC1	PCC0	CPU clock (f_{CPU}) select
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

Notes 1. Bit 5 is Read Only.

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

Cautions 1. Bit 3 must be set to 0.

2. When the external clock is input, MCC should not be set.
This is because the X2 pin is connected to V_{DD1} via a pull-up resistor.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

The fastest instructions of μPD780065 Subseries are carried out in 2 CPU clocks. The relationship of CPU clock (f_{CPU}) and minimum instruction execution time is shown in Table 5-2.

Table 5-2. Relationship of CPU Clock and Min. Instruction Execution Time

CPU Clock (f_{CPU})	Min. Instruction Execution Time: 2/(f_{CPU})
f_X	0.24 μs
$f_X/2$	0.48 μs
$f_X/2^2$	0.95 μs
$f_X/2^3$	1.91 μs
$f_X/2^4$	3.81 μs
f_{XT}	122 μs

f_X = 8.38 MHz, f_{XT} = 32.768 kHz

f_X: Main system clock oscillation frequency

f_{XT}: Subsystem clock oscillation frequency

5.4 System Clock Oscillator

5.4.1 Main system clock oscillator

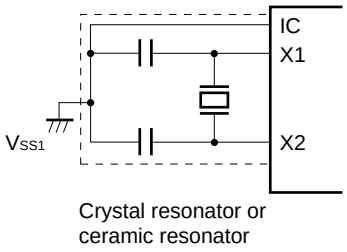
The main system clock oscillator oscillates with a crystal resonator or a ceramic resonator (standard: 8.38 MHz) connected to the X1 and X2 pins.

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

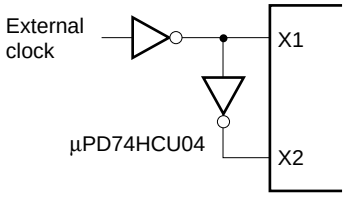
Figure 5-4 shows an external circuit of the main system clock oscillator.

Figure 5-4. External Circuit of Main System Clock Oscillator

(a) Crystal and ceramic oscillation



(b) External clock



Caution Do not execute the STOP instruction and do not set MCC [bit 7 of processor clock control register (PCC)] to 1 if an external clock is input. This is because when the STOP instruction or MCC is set to 1, the main system clock operation stops and the X2 pin is connected to V_{DD1} via a pull-up resistor.

5.4.2 Subsystem clock oscillator

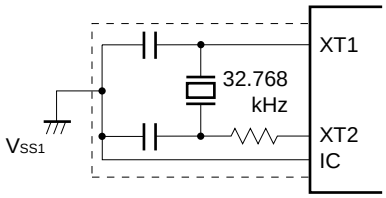
The subsystem clock oscillator oscillates with a crystal resonator (standard: 32.768 kHz) connected to the XT1 and XT2 pins.

External clocks can be input to the main system clock oscillator. In this case, input a clock signal to the XT1 pin and an inverted-phase clock signal to the XT2 pin.

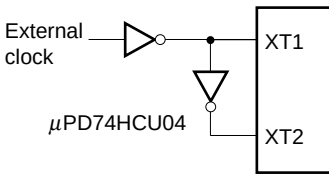
Figure 5-5 shows an external circuit of the subsystem clock oscillator.

Figure 5-5. External Circuit of Subsystem Clock Oscillator

(a) Crystal oscillation



(b) External clock



Cautions are listed on the next page.

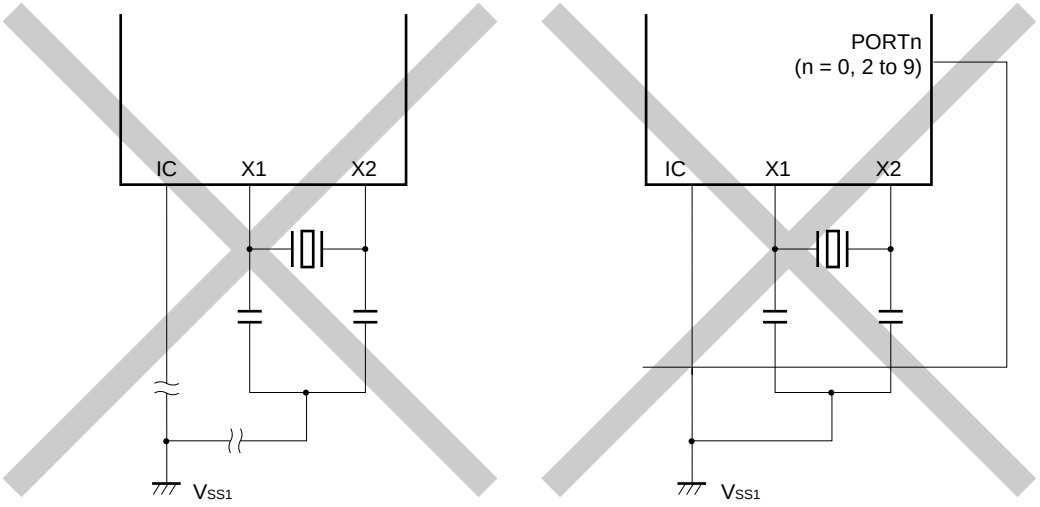
Cautions 1. When using the main system clock oscillator and a subsystem clock oscillator, carry out wiring in the broken line area in Figures 5-4 and 5-5 to prevent any effects from wiring capacitance.

- **Minimize the wiring length.**
- **Do not allow wiring to intersect with other signal lines. Do not route the wiring in the vicinity of a line through which a high-fluctuating current flows.**
- **Always keep the ground of the capacitor of the oscillation circuit at the same potential as V_{SS1}. Do not ground a capacitor to a ground pattern where high-current flows.**
- **Do not fetch signals from the oscillator.**

Take special note of the fact that the subsystem clock oscillator is a circuit with low-level amplification so that current consumption is maintained at low levels.
Figure 5-6 shows examples of incorrect oscillator connection.

Figure 5-6. Examples of Incorrect Oscillator Connection (1/2)

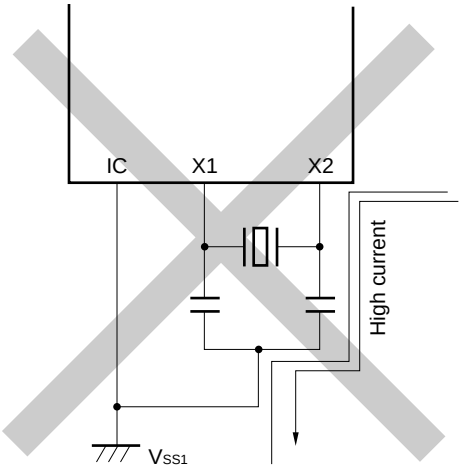
(a) Too long wiring **(b) Crossed signal line**



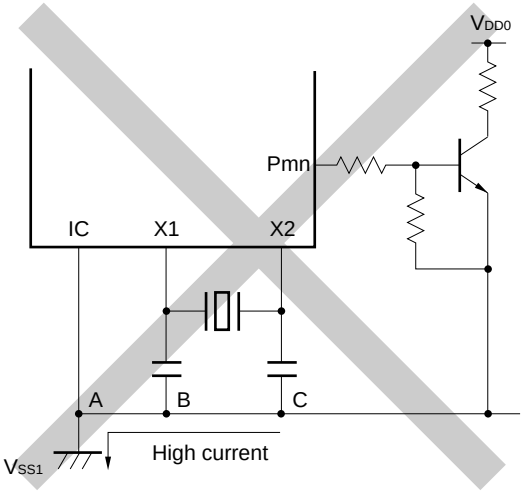
Remark When using a subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Further, insert resistors in series on the side of XT2.

Figure 5-6. Examples of Incorrect Oscillator Connection (2/2)

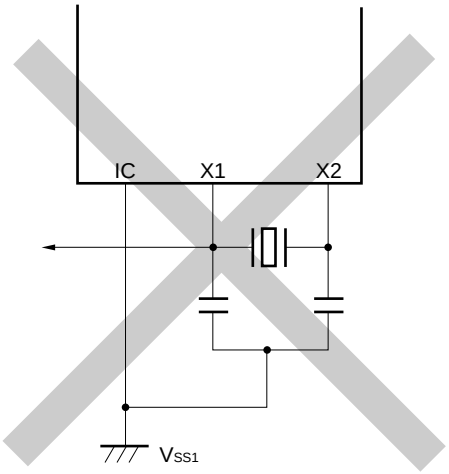
(c) Wiring near high alternating current



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



(e) Signals are fetched



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

Cautions 2. When X2 and XT1 are wired in parallel, the cross-talk noise of X2 may increase with XT1, resulting in malfunctioning.
To prevent that from occurring, it is recommended to wire X2 and XT1 so that they are not in parallel, and to connect the IC pin between X2 and XT1 directly to V_{SS1}.

5.4.3 Scaler

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

5.4.4 When no subsystem clocks are used

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

XT1: Connect to V_{DD0}
XT2: Open

In this state, however, some current may leak via the internal feedback resistor of the subsystem clock oscillator when the main system clock stops. To minimize leakage current, the above internal feedback resistor can be removed with bit 6 (FRC) of the processor clock control register (PCC). 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 following clock generator functions and operations are determined with the processor clock control register (PCC).

(a) Upon generation of $\overline{\text{RESET}}$ signal, the lowest speed mode of the main system clock (3.81 μs @ 8.38-MHz oscillation) is selected (PCC = 04H). Main system clock oscillation stops while low level is applied to $\overline{\text{RESET}}$ pin.

(b) With the main system clock selected, one of the five CPU clock types (0.24 μs. 0.48 μs, 0.95 μs, 1.91 μs, 3.81 μs, @ 8.38-MHz operation) can be selected by setting the PCC.

(c) With the main system clock selected, two standby modes, the STOP and HALT modes, are available. To reduce current consumption in the STOP mode, the subsystem clock feedback resistor can be disconnected to stop the subsystem clock.

(d) The PCC can be used to select the subsystem clock and to operate the system with low-current consumption (122 μs @ 32.768-kHz operation).

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

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

5.5.1 Main system clock operations

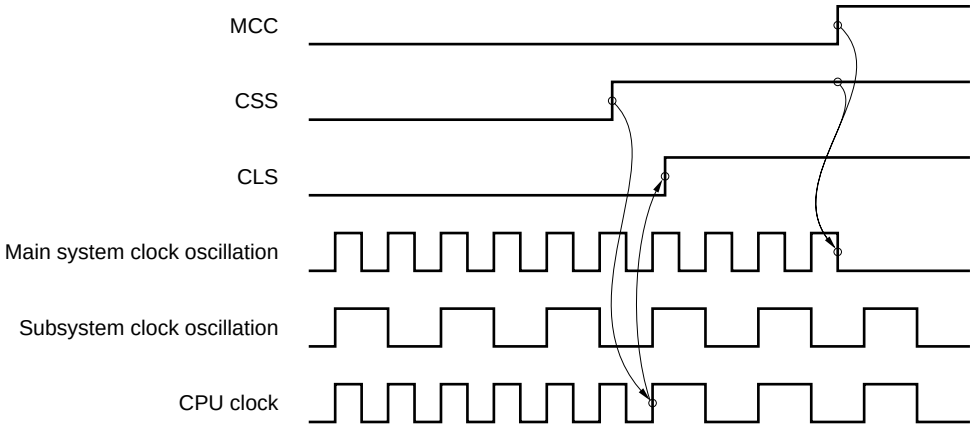
When operated with the main system clock (with bit 5 (CLS) of the processor clock control register (PCC) set to 0), the following operations are carried out by PCC setting.

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

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

Figure 5-7. Main System Clock Stop Function (1/2)

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



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

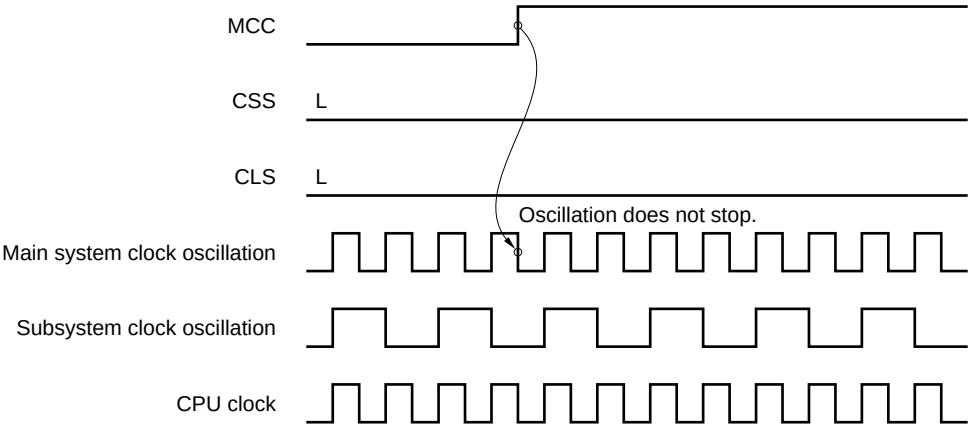
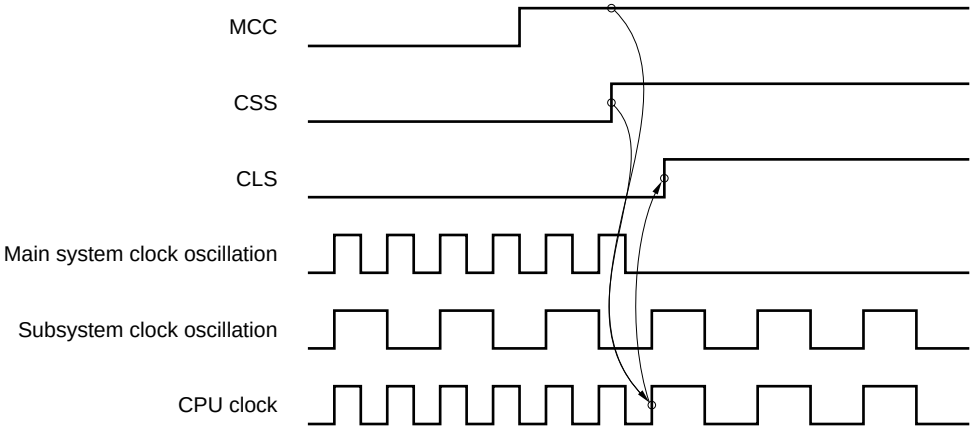


Figure 5-7. Main System Clock Stop Function (2/2)

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



5.5.2 Subsystem clock operations

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

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

(b) Watchdog timer counting stops.

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

5.6 Changing System Clock and CPU Clock Settings

5.6.1 Time required for switchover between system clock and CPU clock

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

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

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

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

Set Value before Switchover				Set Value 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	×	×	×
0	0	0	0					16 instructions				16 instructions				16 instructions				16 instructions				$f_X/2f_{XT}$ instruction (77 instructions)			
	0	0	1	8 instructions								8 instructions				8 instructions				8 instructions				$f_X/4f_{XT}$ instruction (39 instructions)			
	0	1	0	4 instructions				4 instructions								4 instructions				4 instructions				$f_X/8f_{XT}$ instruction (20 instructions)			
	0	1	1	2 instructions				2 instructions				2 instructions								2 instructions				$f_X/16f_{XT}$ instruction (10 instructions)			
	1	0	0	1 instruction				1 instruction				1 instruction				1 instruction								$f_X/32f_{XT}$ instruction (5 instructions)			
1	×	×	×	1 instruction				1 instruction				1 instruction				1 instruction				1 instruction							

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

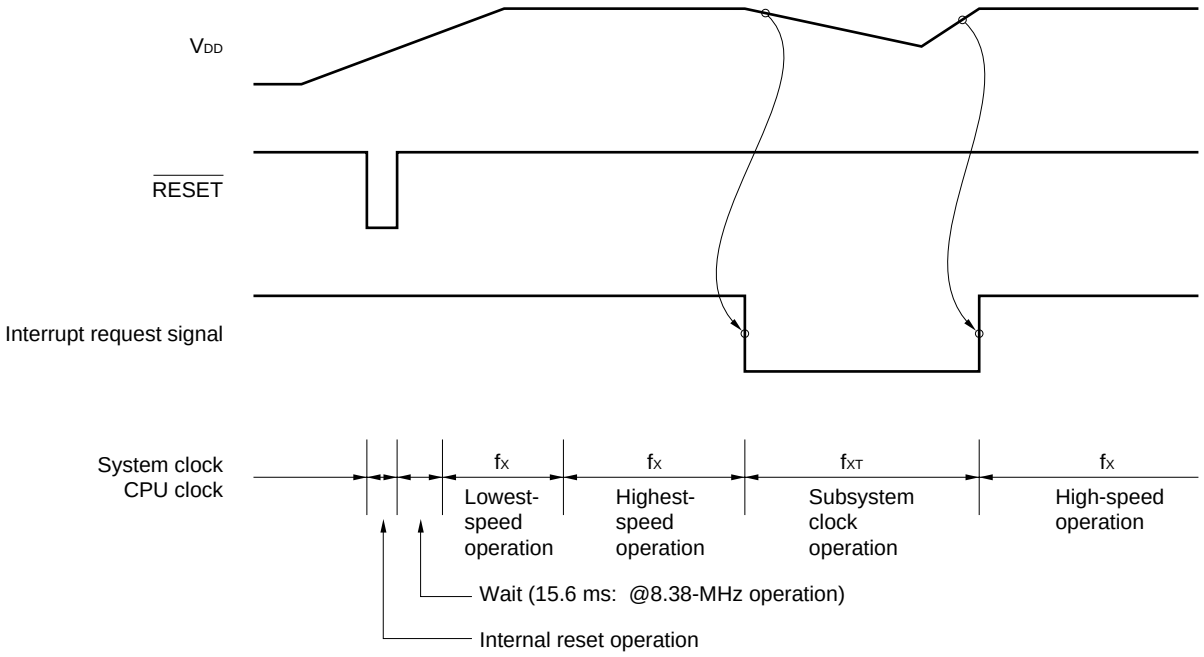
2. Figures in parentheses are for operation with f_X = 8.38 MHz and f_{XT} = 32.768 kHz.

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

5.6.2 System clock and CPU clock switching procedure

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

Figure 5-8. System Clock and CPU Clock Switching



<1> The CPU is reset by setting the $\overline{\text{RESET}}$ signal to low level after power-on. After that, when reset is released by setting the $\overline{\text{RESET}}$ signal to high level, main system clock starts oscillation. At this time, oscillation stabilization time ($2^{17}/f_X$) is secured automatically.
After that, the CPU starts executing the instruction at the minimum speed of the main system clock (3.81 μs @ 8.38-MHz operation).

<2> After the lapse of a sufficient time for the V_{DD} voltage to increase to enable operation at maximum speeds, the PCC is rewritten and 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, 0 is set to the MCC and oscillation of the main system clock is started. After the lapse of time required for stabilization of oscillation, the PCC is rewritten and the maximum-speed operation is resumed.

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

CHAPTER 6 16-BIT TIMER/EVENT COUNTER

6.1 Outline of Timer Integrated in μPD780065 Subseries

In this chapter, the 16-bit timer/event counter is described. The timers integrated in the μPD780065 Subseries are outlined below.

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

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

(2) 8-bit timer/event counter (TM5)

The TM5 can be used to serve as an interval timer, an external event counter, to output square wave output with any selected frequency, and PWM output. Two 8-bit timer/event counters can be used as one 16-bit timer/event counter (See **CHAPTER 7 8-BIT TIMER/EVENT COUNTER**).

(3) Watch timer (WT)

This timer can set a flag every 0.5 sec. or 0.25 sec. and simultaneously generate an interrupt request at the preset time intervals (See **CHAPTER 8 WATCH TIMER**).

(4) Watchdog timer (WDT)

The watchdog timer can also be used to generate a non-maskable interrupt request, maskable interrupt request, or $\overline{\text{RESET}}$ signal at the preset time intervals (See **CHAPTER 9 WATCHDOG TIMER**).

(5) Clock output control circuit (CKU)

The clock output circuit supplies other devices with the divided main system clock and the subsystem clock (See **CHAPTER 10 CLOCK OUTPUT CONTROL CIRCUITS**).

★ **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[Note1]	1 channel[Note2]
	External event counter	√	√	—	—
Function	Timer output	√	√	—	—
	PPG output	√	—	—	—
	PWM output	—	√	—	—
	Pulse width measurement	√	—	—	—
	Square-wave output	√	√	—	—
	One-shot pulse output	√	—	—	—
	Interrupt request	√	√	√	√

Notes 1. The watch timer can perform both watch timer and interval timer functions at the same time.

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

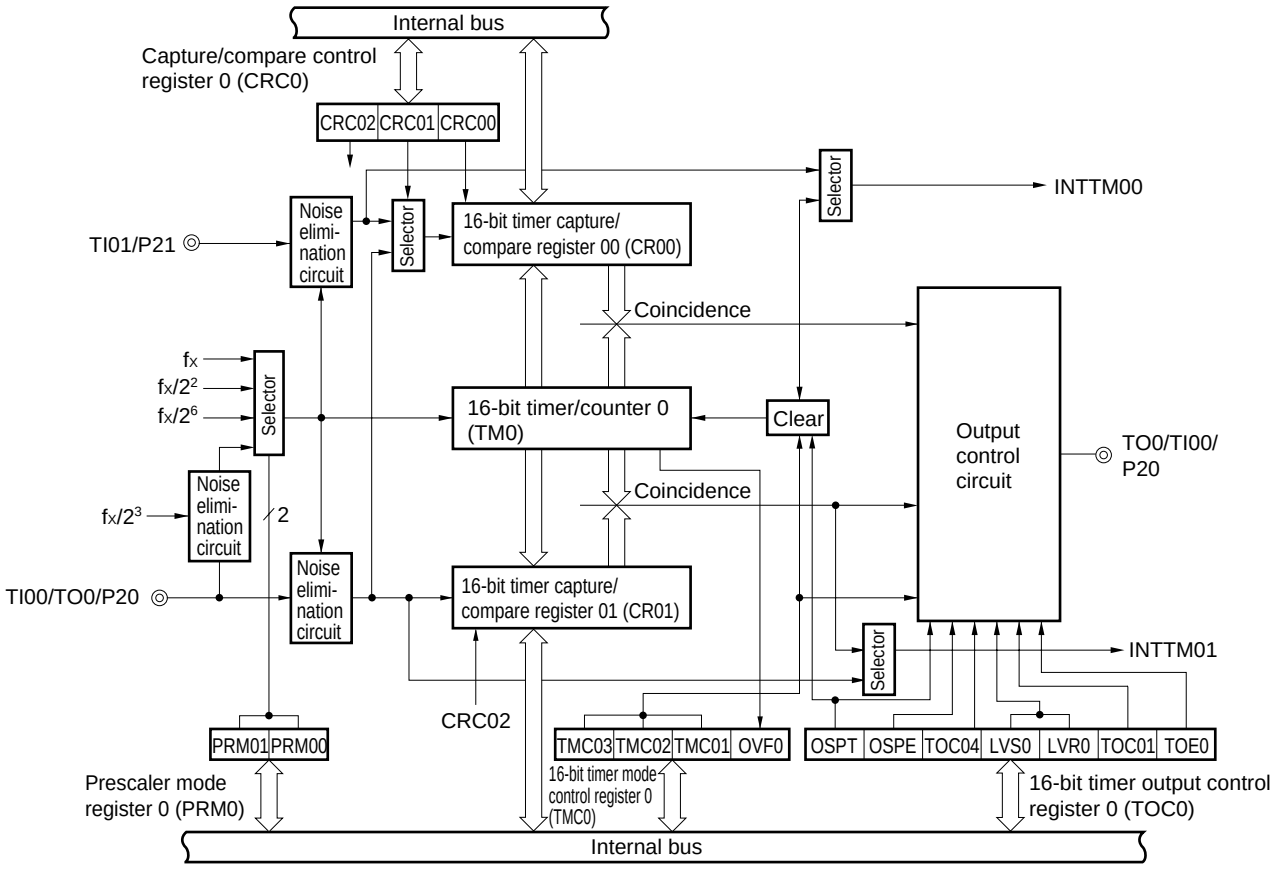
6.2 16-Bit Timer/Event Counter Functions

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

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

Figure 6-1 shows 16-bit timer/event counter block diagram.

★ **Figure 6-1. 16-Bit Timer/Event Counter Block Diagram**



(1) Interval timer

TM0 generates interrupt request at the preset time interval.

(2) PPG output

TM0 can output a square wave whose frequency and output pulse can be set freely.

(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.

(6) One-shot pulse output

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

6.3 16-Bit Timer/Event Counter Configuration

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

Table 6-2. 16-Bit Timer/Event Counter Configuration

Item	Configuration
Timer register	16 bits × 1 (TM0)
Register	Capture/compare register: 16 bits × 2 (CR00, CR01)
Timer output	1 (TO0)
Control register	16-bit timer mode control register 0 (TMC0) Capture/compare control register 0 (CRC0) 16-bit timer output control register 0 (TOC0) Prescaler mode register 0 (PRM0) Port mode register 2 (PM2)**Note**

Note See **Block Diagram of Figure 4-3 P20 to P27.**

(1) 16-bit timer/counter 0 (TM0)

TM0 is a 16-bit read-only register that counts count pulses.

The counter is incremented in synchronization with the rising edge of an input clock. If the count value is read during operation, input of the count clock is temporarily stopped, and the count value at that point is read.

The count value is reset to 0000H in the following cases:

<1> At $\overline{\text{RESET}}$ input
<2> If TMC03 and TMC02 are cleared
<3> If valid edge of TI00 is input in the clear & start mode by inputting valid edge of TI00
<4> If TM0 and CR00 coincide with each other in the clear & start mode on coincidence between TM0 and CR00
<5> OSPT is set in the one-shot pulse output mode

(2) 16-bit timer capture/compare register 00 (CR00)

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

- **When CR00 is used as a compare register**

  The value set in the CR00 is constantly compared with the 16-bit timer/counter 0 (TM0) count value, and an interrupt request (INTTM00) is generated if they match. It can also be used as the register which holds the interval time then TM0 is set to interval timer operation, and as the register which sets the pulse width in the PWM operating mode.

- **When CR00 is used as a capture register**

  It is possible to select the valid edge of the TI00/TO0/P20 pin or the TI01/P21 pin as the capture trigger. Setting of the TI00 or TI01 valid edge is performed by means of prescaler mode register 0 (PRM0).

  If CR00 is specified as a capture register and capture trigger is specified to be the valid edge of the TI00/TO0/P20 pin, the situation is as shown in Table 6-3. On the other hand, when capture trigger is specified to be the valid edge of the TI01/P21 pin, the situation is as shown in Table 6-4.

★ **Table 6-3. TI00/TO0/P20 Pin Valid Edge and Capture/Compare Register Capture Trigger**

ES01	ES00	TI00/TO0/P20 Pin Valid Edge	CR00 Capture Trigger	CR01 Capture Trigger
0	0	Falling edge	Rising edge	Rising edge
0	1	Rising edge	Falling edge	Falling edge
1	0	Setting prohibited	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	No capture operation	Both rising and falling edges

Table 6-4. TI01/P21 Pin Valid Edge and Capture/Compare Register Capture Trigger

ES11	ES10	TI01/P21 Pin Valid Edge	CR00 Capture Trigger
0	0	Falling edge	Falling edge
0	1	Rising edge	Rising edge
1	0	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	Both rising and falling edges

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

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

Cautions 1. Set a value other than 0000H in CR00. This means 1-pulse count operation cannot be performed when CR00 is used as an event counter. However, in the free-running mode and in the clear mode using the valid edge of TI00, if 0000H is set to CR00, an interrupt request (INTTM00) is generated following overflow (FFFFH).

2. When P20 is used as the valid edge of TI00, it cannot be used as timer output (TO0). Moreover, when P20 is used as TO0, it cannot be used as the valid edge of TI00.

(3) 16-bit timer capture/compare register 01 (CR01)

CR01 is a 16-bit register which has the functions of both a capture register and a compare register. Whether it is used as a capture register or a compare register is set by bit 2 (CRC02) of capture/compare control register 0 (CRC0).

- **When CR01 is used as a compare register**
  The value set in the CR01 is constantly compared with the 16-bit timer/counter 0 (TM0) count value, and an interrupt request (INTTM01) is generated if they match.

- **When CR01 is used as a capture register**
  It is possible to select the valid edge of the TI00/TO0/P20 pin as the capture trigger (refer to **Table 6-3**). The TI00/TO0/P20 valid edge is set by means of prescaler mode register 0 (PRM0).

CR01 is set with a 16-bit memory manipulation instruction.
After $\overline{\text{RESET}}$ input, the value of CR01 is undefined.

Caution Set other than 0000H to CR01. This means 1-pulse count operation cannot be performed when CR01 is used as the event counter. However, in the free-running mode and in the clear mode using the valid edge of TI00, if 0000H is set to CR01, an interrupt request (INTTM01) is generated following overflow (FFFFH).

6.4 Registers to Control 16-Bit Timer/Event Counter

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

- 16-bit timer mode control register 0 (TMC0)
- Capture/compare control register 0 (CRC0)
- 16-bit timer output control register 0 (TOC0)
- Prescaler mode register 0 (PRM0)
- Port mode register 2 (PM2)

(1) 16-bit timer mode control register 0 (TMC0)

This register sets the 16-bit timer operating mode, the 16-bit timer/counter 0 (TM0) clear mode, and output timing, and detects an overflow.
TMC0 is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets TMC0 value to 00H.

Caution The 16-bit timer/counter 0 (TM0) starts operation at the moment a value other than 0, 0 (operation stop mode) is set in TMC02 to TMC03, respectively. Set 0, 0 in TMC02 to TMC03 to stop the operation.

Figure 6-2. Format of 16-Bit Timer Mode Control Register 0 (TMC0)

Address FF60H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TMC0	0	0	0	0	TMC03	TMC02	TMC01	OVF0

TMC03	TMC02	TMC01	Operating mode and 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			
0	1	0	Free running mode	Match between TM0 and CR00 or match between TM0 and CR01	Generated on match between TM0 and CR00, or match between TM0 and CR01
0	1	1		Match between TM0 and CR00, match between TM0 and CR01 or TI00 valid edge	
1	0	0	Clear & start on TI00 valid edge	—	
1	0	1			
1	1	0	Clear & start on match between TM0 and CR00	Match between TM0 and CR00 or match between TM0 and CR01	
1	1	1		Match between TM0 and CR00, match between TM0 and CR01 or TI00 valid edge	

OVF0	TM0 overflow detection
0	Overflow not detected
1	Overflow detected

Cautions 1. Timer operation must be stopped before writing to bits other than the OVF0 flag.

2. Set the valid edge of the TI00/TO0/P20 pin with prescaler mode register 0 (PRM0).

3. If clear & start mode on match between TM0 and CR00 is selected, when the set value of CR00 is FFFFH and the TM0 value changes from FFFFH to 0000H, OVF0 flag is set to 1.

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

2. TI00: 16-bit timer/event counter input pin

3. TM0: 16-bit timer/counter 0

4. CR00: Compare register 00

5. CR01: Compare register 01

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

This register controls the operation of the capture/compare registers (CR00, CR01).

CRC0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CRC0 value to 00H.

Figure 6-3. Format of Capture/Compare Control Register 0 (CRC0)

Address: FF62H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC0	0	0	0	0	0	CRC02	CRC01	CRC00

CRC02	CR01 operating mode selection
0	Operates as compare register
1	Operates as capture register

CRC01	CR00 capture trigger selection
0	Captures on valid edge of TI01
1	Captures on valid edge of TI00 by reverse phase

CRC00	CR00n operating mode selection
0	Operates as compare register
1	Operates as capture register

Cautions 1. Timer operation must be stopped before setting CRC0.

2. When clear & start mode on a match between TM0 and CR00 is selected with the 16-bit timer mode control register 0 (TMC0), CR00 should not be specified as a capture register.

3. If both the rising and falling edges have been selected as the valid edges of TI00, capture is not performed.

(3) 16-bit timer output control register 0 (TOC0)

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

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

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

Figure 6-4 shows the TOC0 format.

Figure 6-4. Format of 16-Bit Timer Output Control Register 0 (TOC0)

Address: FF63H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TOC0	0	OSPT	OSPE	TOC04	LVS0	LVR0	TOC01	TOE0

OSPT	Control of one-shot pulse output trigger by software
0	One-shot pulse trigger not used
1	One-shot pulse trigger used

OSPE	One-shot pulse output control
0	Continuous pulse output
1	One-shot pulse output[Note]

TOC04	Timer output F/F control by match of CR01 and TM0
0	Inversion operation disabled
1	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

TOC01	Timer output F/F control by match of CR00 and TM0
0	Inversion operation disabled
1	Inversion operation enabled

TOE0	16-bit timer/event counter output control
0	Output disabled (Output set to level 0)
1	Output enabled

Note One-shot pulse output operates normally only in the free-running mode.

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.

3. OSPT is cleared automatically after data setting, and will therefore be 0 if read.

★ **4. Be sure not to set (1) OSPT other than in one-shot pulse output mode.**

(4) Prescaler mode register 0 (PRM0)

This register is used to set 16-bit timer/counter 0 (TM0) count clock and TI00, TI01 input valid edges.

PRM0 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PRM0 value to 00H.

Figure 6-5. Format of Prescaler Mode Register 0 (PRM0)

Address: FF61H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM0	ES11	ES10	ES01	ES00	0	0	PRM01	PRM00

ES11	ES10	TI01 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES01	ES00	TI00 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM01	PRM00	Count clock selection
0	0	f_X (8.38 MHz)
0	1	$f_X/2^2$ (2.09 MHz)
1	0	$f_X/2^6$ (131 kHz)
1	1	TI00 valid edge**Note**

★ **Note** The external clock needs a pulse more than twice the length of the internal clock ($f_X/2^3$).

Cautions 1. If the valid edge of TI00 is to be set to the count clock, do not set the clear/start mode and the capture trigger at the valid edge of TI00.
Moreover, do not use the P20/TI00/TO0 pins as timer outputs (TO0).

2. Always set data to PRM0 after stopping the timer operation.

★ **3. To secure capture of the capture trigger, a pulse more than twice the length of the count clock to be selected is required. When TI00 is high level just after system reset, the falling edge is detected just after the TM0 operation is permitted. Be aware of this if using a pull-up resistor.**

Remarks 1. f_X: Main system clock oscillation frequency

2. TI00, TI01: 16-bit timer/event counter input pin

3. Figures in parentheses are for operation with f_X = 8.38 MHz.

(5) Port mode register 2 (PM2)

This register sets port 2 input/output in 1-bit units.

When using the P20/TO0/TI00 pin for timer output, set PM20 and the output latch of P20 to 0.

PM2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM7 value to FFH.

Figure 6-6. Format of Port Mode Register 2 (PM2)

Address: FF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

PM2n	P2n pin input/output mode selection (n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

6.5 16-Bit Timer/Event Counter Operations

6.5.1 Interval timer operations

Setting 16-bit timer mode control register 0 (TMC0) and capture/compare control register 0 (CRC0) as shown in Figure 6-7 allows operation as an interval timer. Interrupt request is generated repeatedly using the count value set in 16-bit timer capture/compare register 00 (CR00) beforehand as the interval.

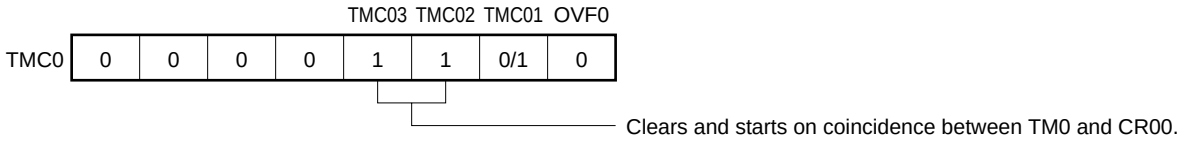
When the count value of 16-bit timer/counter 0 (TM0) matches the value set to CR00, counting continues with the TM0 value cleared to 0 and the interrupt request signal (INTTM00) is generated.

Count clock of the 16-bit timer/event counter can be selected with bits 0 to 1 (PRM00, PRM01) of prescaler mode register 0 (PRM0).

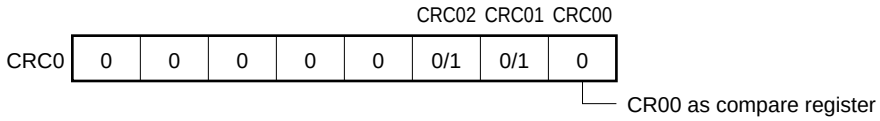
See **6.6 16-Bit Timer/Event Counter Operating Precautions (2) 16-bit Compare-Register Setting** about the operation when the compare register value is changed during timer count operation.

Figure 6-7. Control Register Settings for Interval Timer Operation

(a) 16-bit timer mode control register 0 (TMC0)



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



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the interval timer. See Figures 6-2 and 6-3.

★ **Figure 6-8. Interval Timer Configuration Diagram**

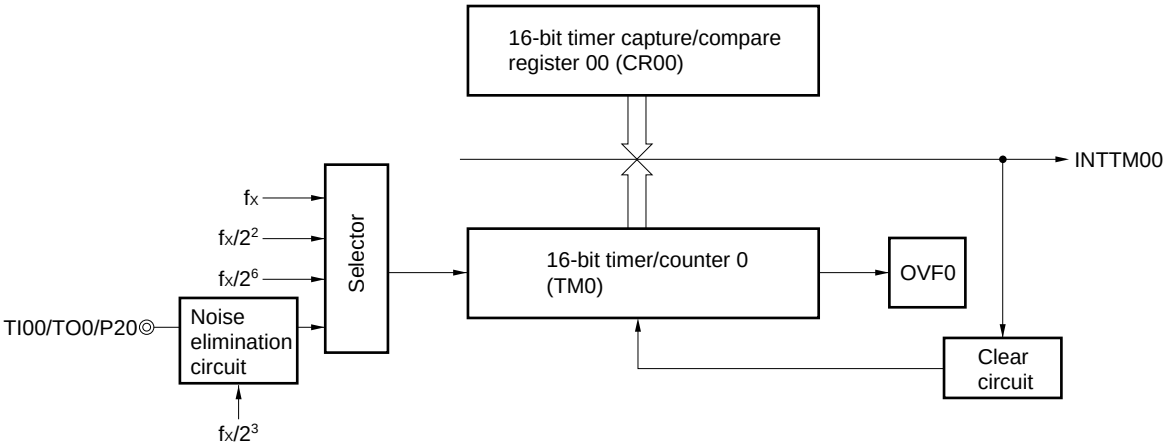
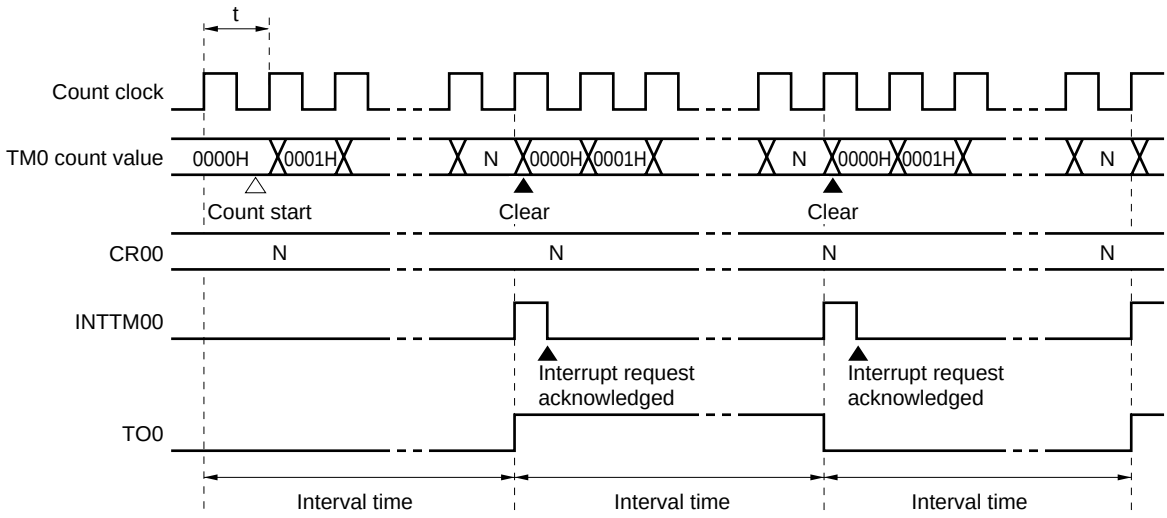


Figure 6-9. Timing of Interval Timer Operation



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

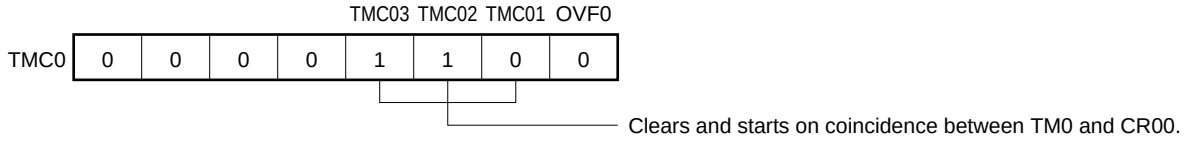
6.5.2 PPG output operations

Setting 16-bit timer mode control register 0 (TMC0) and capture/compare control register 0 (CRC0) as shown in Figure 6-10 allows operation as PPG (Programmable Pulse Generator) output.

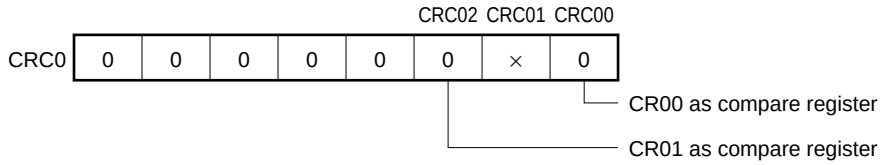
In the PPG output operation, square waves are output from the TO0/TI00/P20 pin with the pulse width and the cycle that corresponds to the count values set beforehand in 16-bit timer capture/compare register 01 (CR01) and in 16-bit timer capture/compare register 00 (CR00), respectively.

Figure 6-10. Control Register Settings for PPG Output Operation

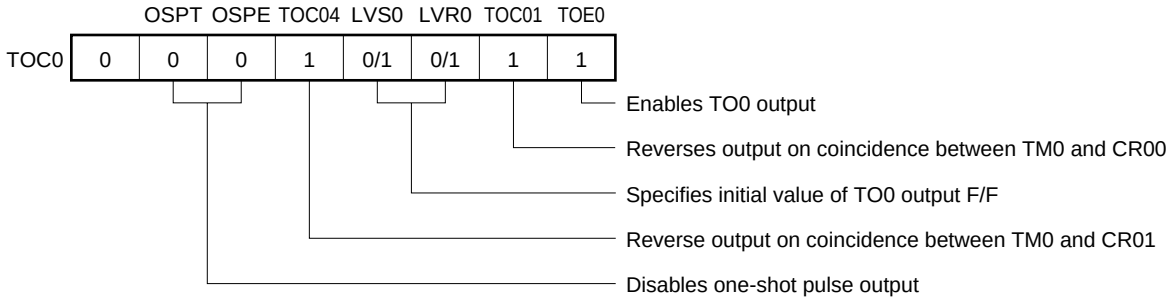
(a) 16-bit timer mode control register 0 (TMC0)



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



(c) 16-bit timer output control register 0 (TOC0)



Cautions 1. Values in the following range should be set in CR00 and CR01:
0000H < CR01 < CR00 ≤ FFFFH

2. The cycle of the pulse generated through PPG output (CR00 setting value + 1) has a duty of (CR01 setting value + 1)/(CR00 setting value + 1).

Remark × : don't care

6.5.3 Pulse width measurement operations

It is possible to measure the pulse width of the signals input to the TI00/TO0/P20 pin and TI01/P21 pin using 16-bit timer/counter 0 (TM0).

There are two measurement methods: measuring with TM0 used in free-running mode, and measuring by restarting the timer in synchronization with the edge of the signal input to the TI00/TO0/P20 pin.

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

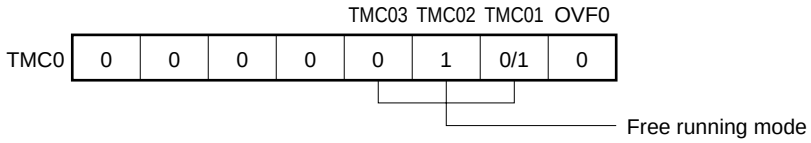
When 16-bit timer/counter 0 (TM0) is operated in free-running mode (see register settings in Figure 6-11), and the edge specified by prescaler mode register 0 (PRM0) is input to the TI00/TO0/P20 pin, the value of TM0 is taken into 16-bit timer capture/compare register 01 (CR01) and an external interrupt request signal (INTTM01) is set.

Any of three edges can be selected—rising, falling, or both edges—specified by means of bits 6 and 7 (ES10 and ES11) of PRM0.

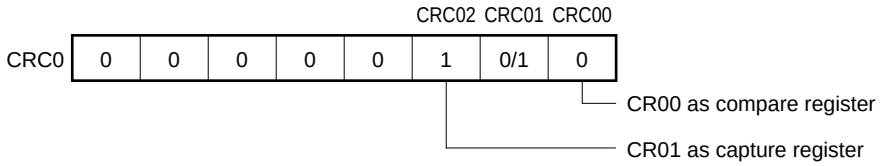
For valid edge detection, sampling is performed at the count clock selected by PRM0, and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Figure 6-11. Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register

(a) 16-bit timer mode control register 0 (TMC0)



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



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See Figures 6-2 and 6-3.

Figure 6-12. Configuration Diagram for Pulse Width Measurement by Free-Running Counter

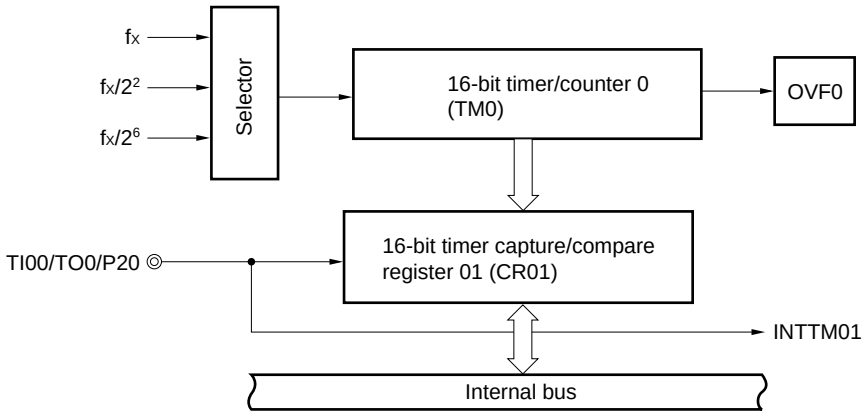
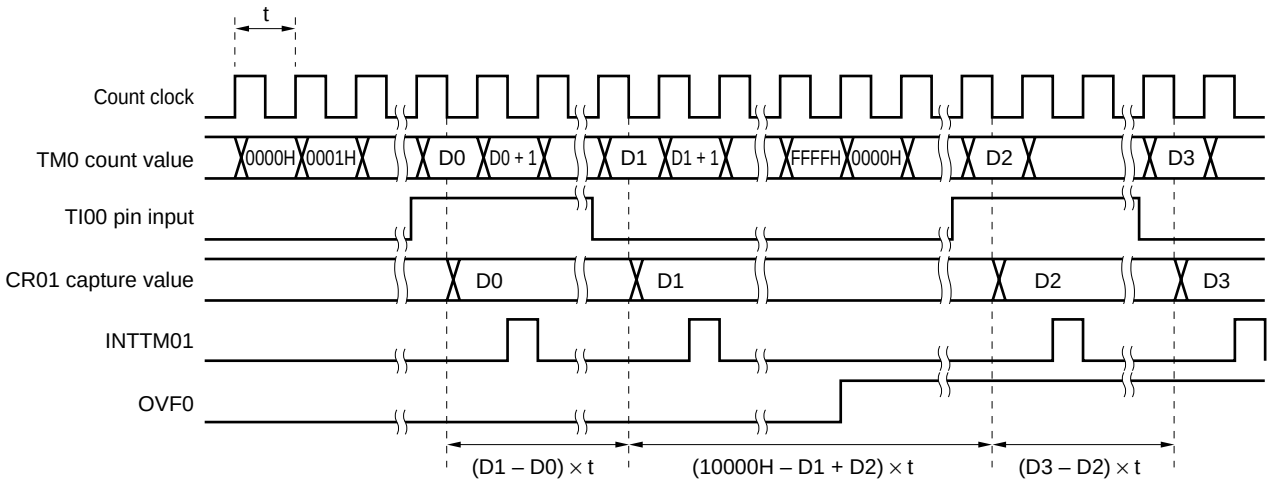


Figure 6-13. Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)



(2) Measurement of two pulse widths with free-running counter

When 16-bit timer/counter 0 (TM0) is operated in free-running mode (see register settings in **Figure 6-14**), it is possible to simultaneously measure the pulse widths of the two signals input to the TI00/TO0/P20 pin and the TI01/P21 pin.

When the edge specified by bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0) is input to the TI00/TO0/P20 pin, the value of TM0 is taken into 16-bit capture/compare register 01 (CR01) and an external interrupt request signal (INTTM01) is set.

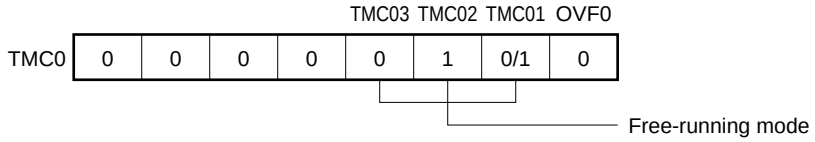
Also, when the edge specified by bits 6 and 7 (ES10 and ES11) of PRM0 is input to the TI01/P21 pin, the value of TM0 is taken into 16-bit capture/compare register 00 (CR00) and an external interrupt request signal (INTTM00) is set.

Any of three edges can be selected—rising, falling, or both edges—as the valid edges for the TI00/TO0/P20 pin and the TI01/P21 pin specified by means of bits 4 and 5 (ES00 and ES01) and bits 6 and 7 (ES10 and ES11) of INTM0, respectively.

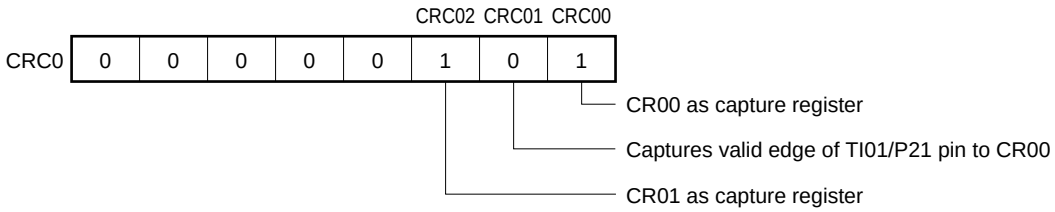
For TI00/TO0/P20 pin valid edges detection, sampling is performed at the interval selected by means of the prescaler mode register 0 (PRM0), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Figure 6-14. Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter

(a) 16-bit timer mode control register 0 (TMC0)



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



Remark 0/1: When these bits are reset to 0 or set to 1, the other functions can be used along with the pulse width measurement function. For details, see Figures 6-2 and 6-3.

- **Capture operation (Free-Running mode)**
  Capture register operation in capture trigger input is shown.

Figure 6-15. Capture Operation with Rising Edge Specified

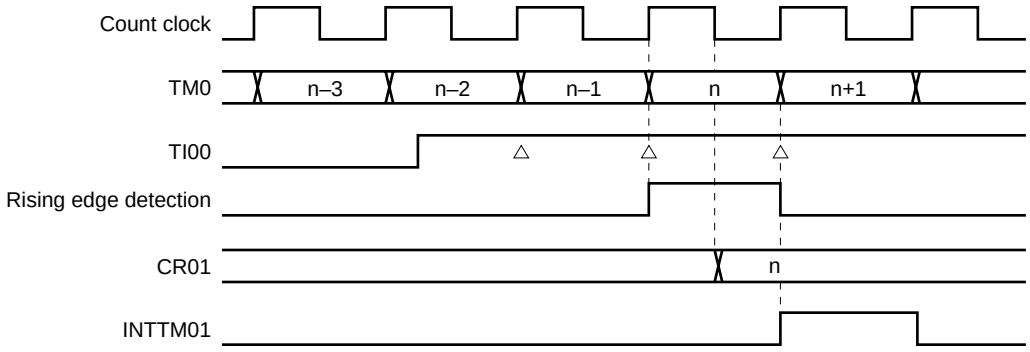
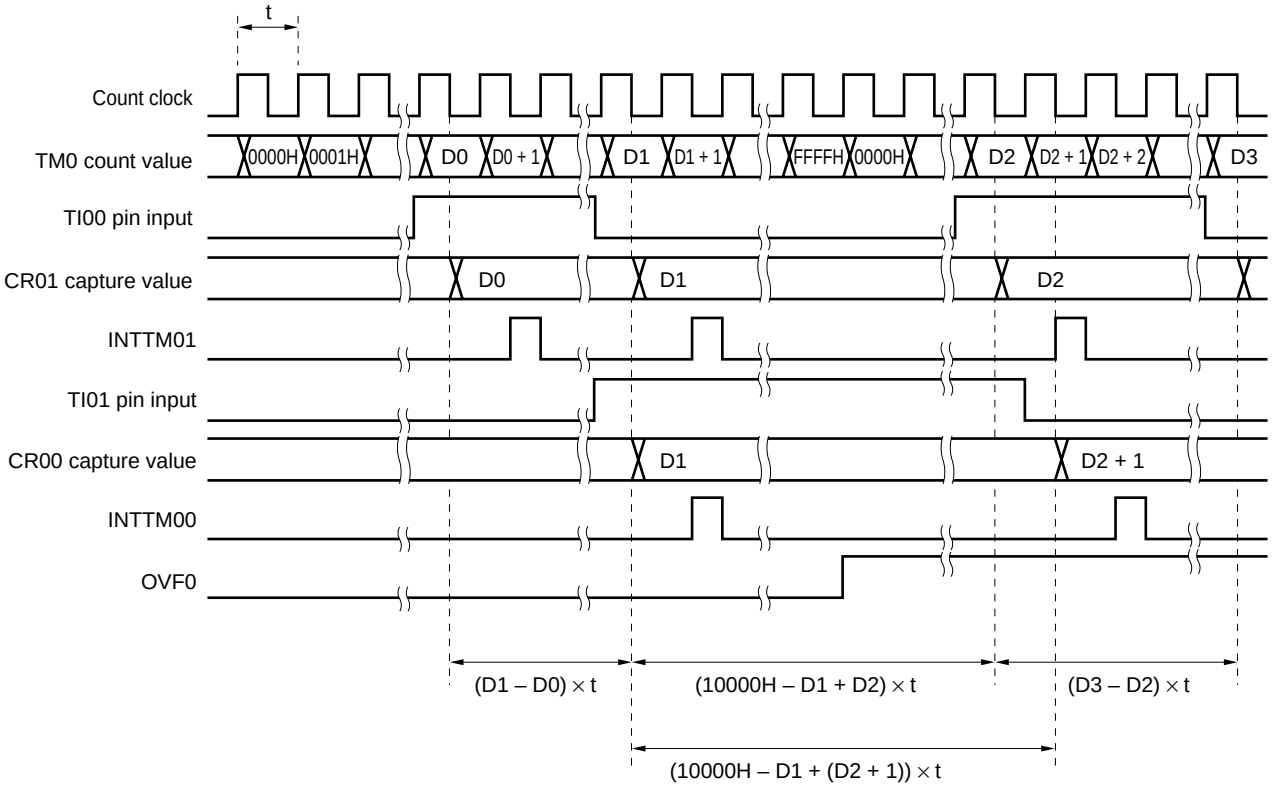


Figure 6-16. Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified)



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

When 16-bit timer/counter 0 (TM0) is operated in free-running mode (see register settings in **Figure 6-17**), it is possible to measure the pulse width of the signal input to the TI00//TO0/P20 pin.

When the edge specified by bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0) is input to the TI00/TO0/P20 pin, the value of TM0 is taken into 16-bit timer capture/compare register 01 (CR01) and an external interrupt request signal (INTTM01) is set.

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

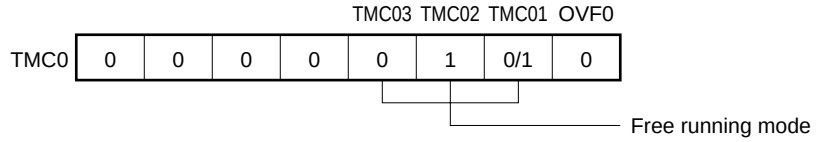
Either of two edges can be selected—rising or falling—as the valid edges for the TI00/TO0/P20 pin specified by means of bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0).

For TI00/TO0/P20 pin valid edge detection, sampling is performed at the interval selected by means of the prescaler mode register (PRM0), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

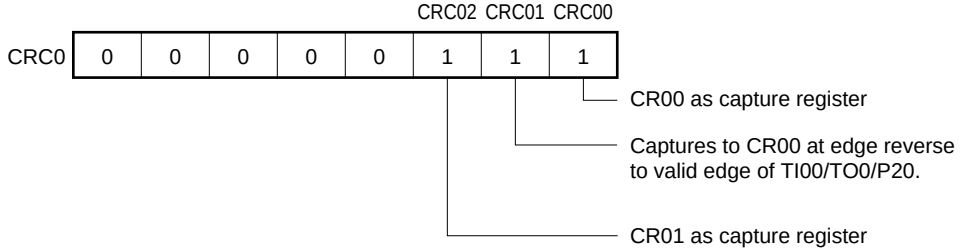
Caution If the valid edge of TI00/TO0/P20 is specified to be both rising and falling edge, capture/compare register 00 (CR00) cannot perform the capture operation.

Figure 6-17. Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers

(a) 16-bit timer mode control register 0 (TMC0)

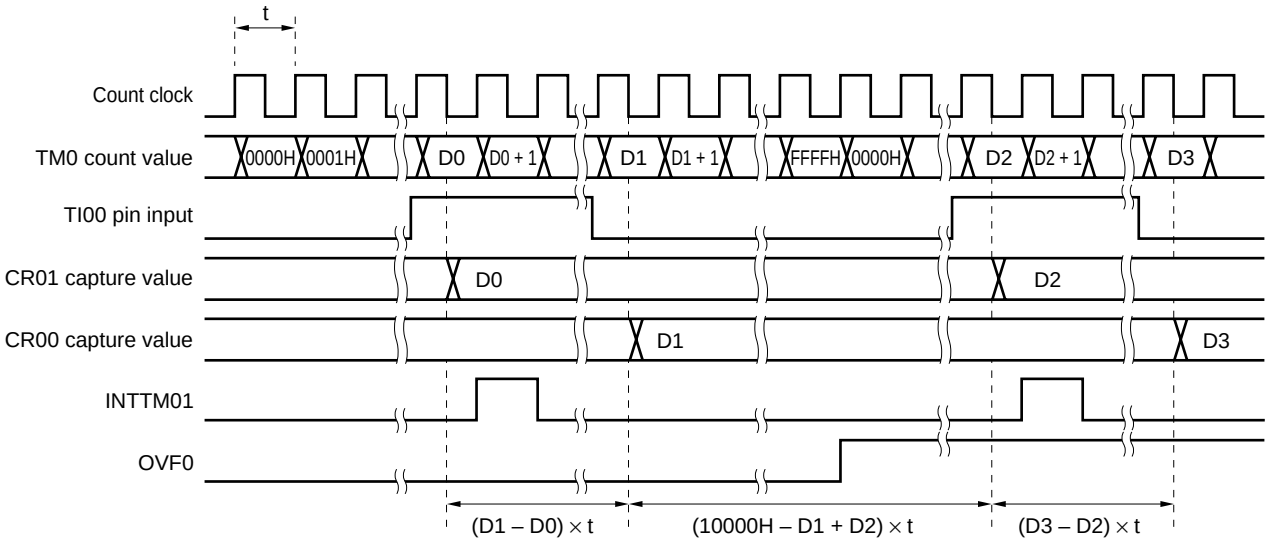


(b) Capture/compare control register 0 (CRC0)



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

Figure 6-18. Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified)



(4) Pulse width measurement by means of restart

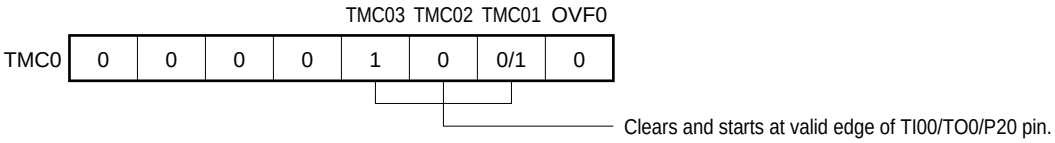
When input of a valid edge to the TI00/TO0/P20 pin is detected, the count value of 16-bit timer/counter 0 (TM0) is taken into 16-bit timer capture/compare register 01 (CR01), and then the pulse width of the signal input to the TI00/TO0/P20 pin is measured by clearing TM0 and restarting the count (see register settings in Figure 6-19). The edge specification can be selected from two types, rising and falling edges by bits 4 and 5 (ES00 and ES01) of the prescaler mode resister 0 (PRM0).

In a valid edge detection, the sampling is performed by a cycle selected by the prescaler mode resistor 0 (PRM0) and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

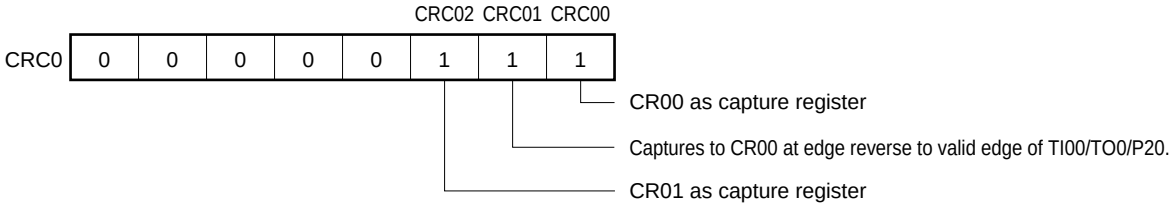
Caution If the valid edge of TI00/TO0/P20 pin is specified to be both rising and falling edges, 16-bit capture/compare register 00 (CR00) cannot perform the capture operation.

Figure 6-19. Control Register Settings for Pulse Width Measurement by Means of Restart

(a) 16-bit timer mode control register 0 (TMC0)

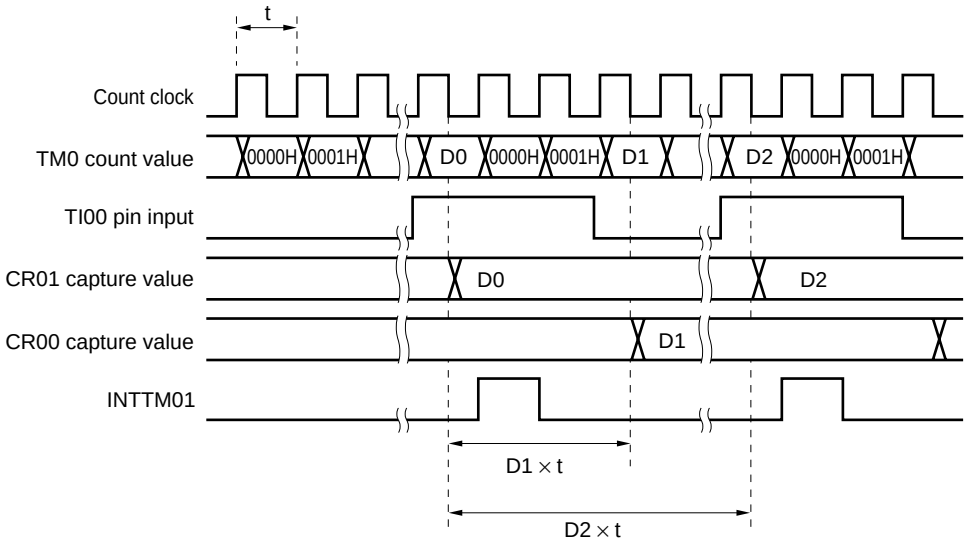


(b) Capture/compare control register 0 (CRC0)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See Figures 6-2 and 6-3.

Figure 6-20. Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)



6.5.4 External event counter operation

The external event counter counts the number of external clock pulses to be input to the TI00/TO0/P20 pin with 16-bit timer/counter 0 (TM0).

TM0 is incremented each time the valid edge specified with the prescaler mode register 0 (PRM0) is input.

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

Input the value except 0000H to CR00. (Count operation with a pulse cannot be carried out.)

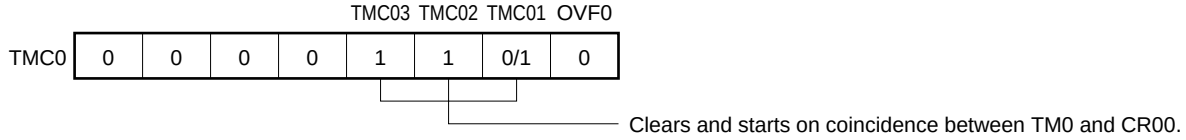
The rising edge, the falling edge, or both edges can be selected with bits 4 and 5 (ES00 and ES01) of prescaler mode register 0 (PRM0).

Because operation is carried out only after the valid edge is detected twice by sampling at internal clock ($f_X/2^3$), noise with short pulse widths can be removed.

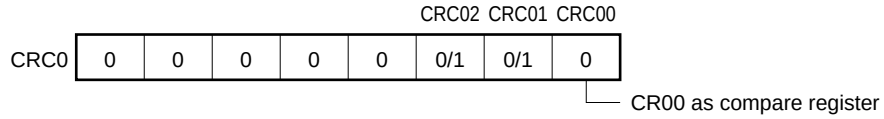
Caution When used as an external event counter, the P20/TI00/TO0 pin cannot be used as timer output (TO0).

Figure 6-21. Control Register Settings in External Event Counter Mode

(a) 16-bit timer mode control register 0 (TMC0)



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



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the external event counter. See Figures 6-2 and 6-3.

Figure 6-22. External Event Counter Configuration Diagram

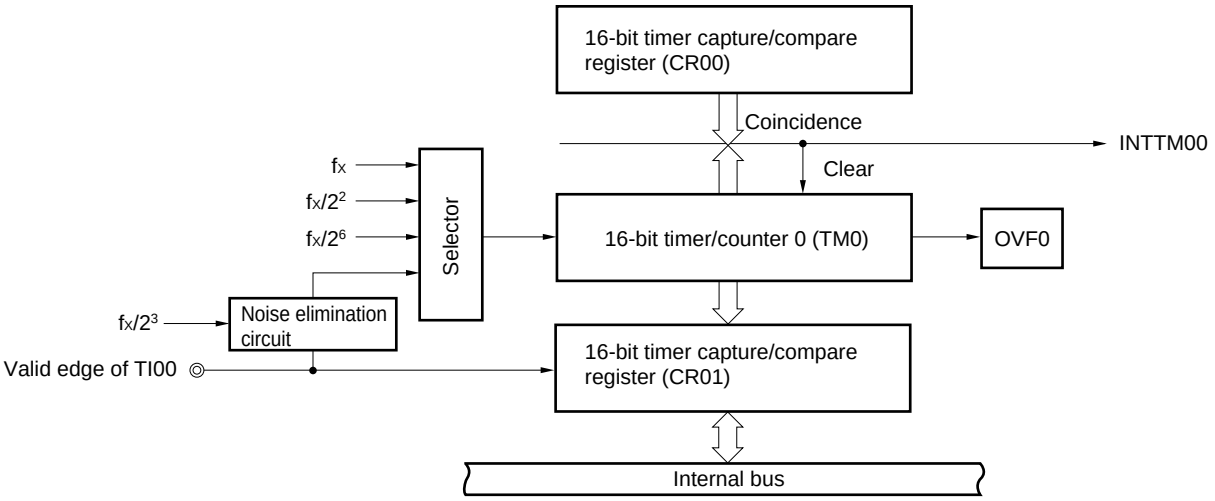
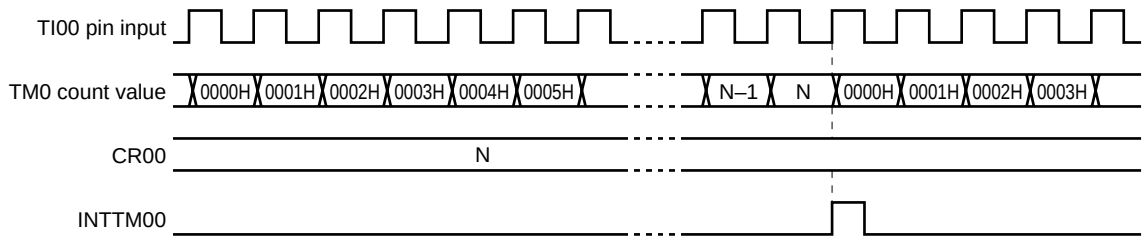


Figure 6-23. External Event Counter Operation Timings (with Rising Edge Specified)



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

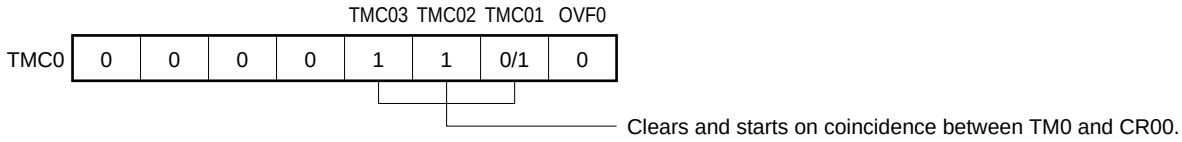
6.5.5 Square-wave output operation

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

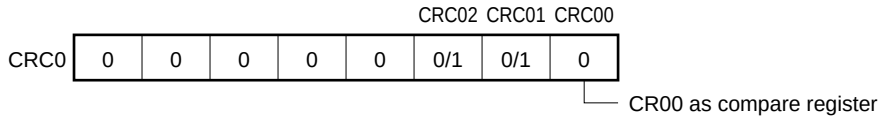
The TO0 pin output status is reversed at intervals of the count value preset to CR00 by setting bit 0 (TOE0) and bit 1 (TOC01) of the 16-bit timer output control register 0 (TOC0) to 1. This enables a square wave with any selected frequency to be output.

Figure 6-24. Control Register Settings in Square-Wave Output Mode

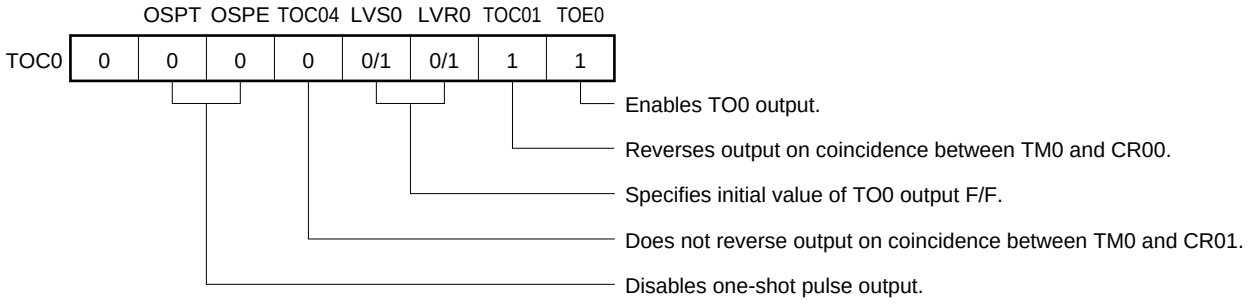
(a) 16-bit timer mode control register 0 (TMC0)



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

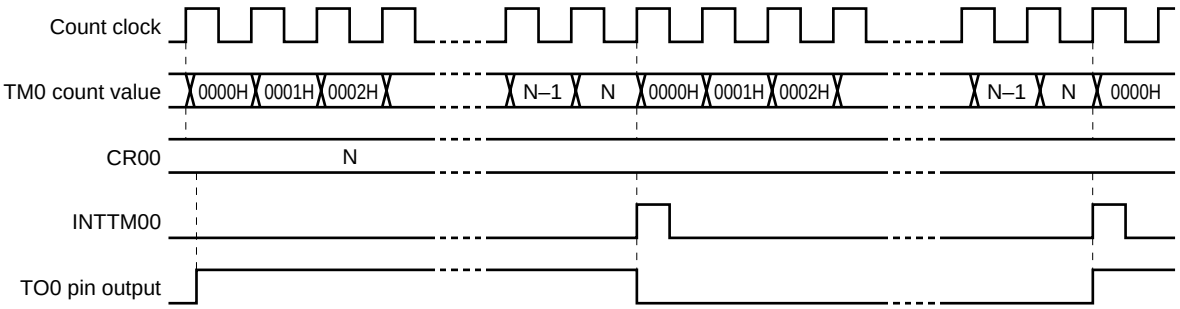


(c) 16-bit timer output control register 0 (TOC0)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with square-wave output. See Figures 6-2, 6-3, and 6-4.

Figure 6-25. Square-Wave Output Operation Timing



6.5.6 One-shot pulse output operation

It is possible to output one-shot pulses by software trigger.

If 16-bit timer mode control register 0 (TMC0), capture/compare control register 0 (CRC0), and 16-bit timer output control register 0 (TOC0) are set as shown in Figure 6-26, and 1 is set in bit 6 (OSPT) of TOC0 by software, a one-shot pulse is output from the TO0/TI00/P20 pin.

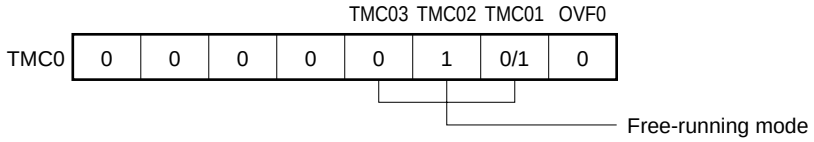
By setting 1 in OSPT, 16-bit timer/counter 0 (TM0) is cleared and started, and output is activated by the count value set beforehand in 16-bit timer capture/compare register 01 (CR01). Thereafter, output is inactivated by the count value set beforehand in 16-bit timer capture/compare register 00 (CR00).

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

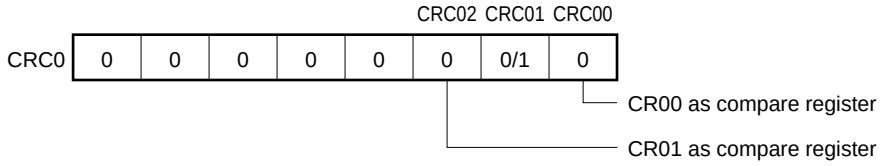
Caution When outputting one-shot pulse, do not set 1 in OSPT. When outputting one-shot pulse again, do so after the INTTM00, or interrupt match signal with CR00, is generated.

Figure 6-26. Control Register Settings for One-Shot Pulse Output Operation Using Software Trigger

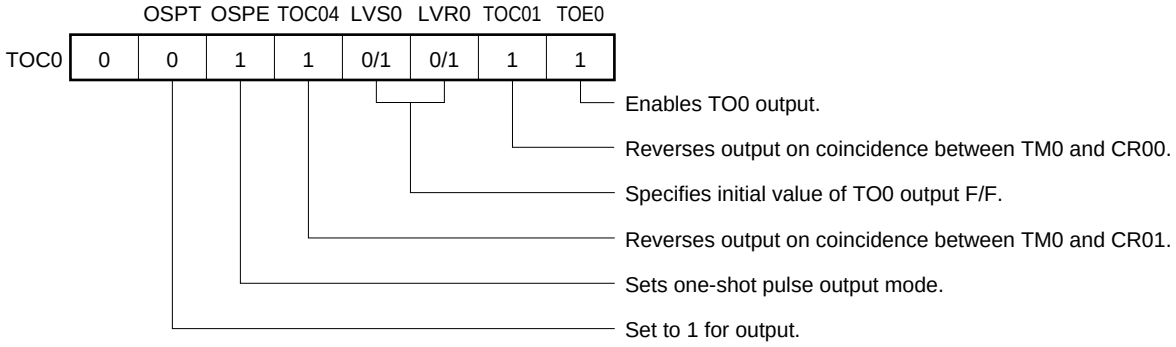
(a) 16-bit timer mode control register 0 (TMC0)



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



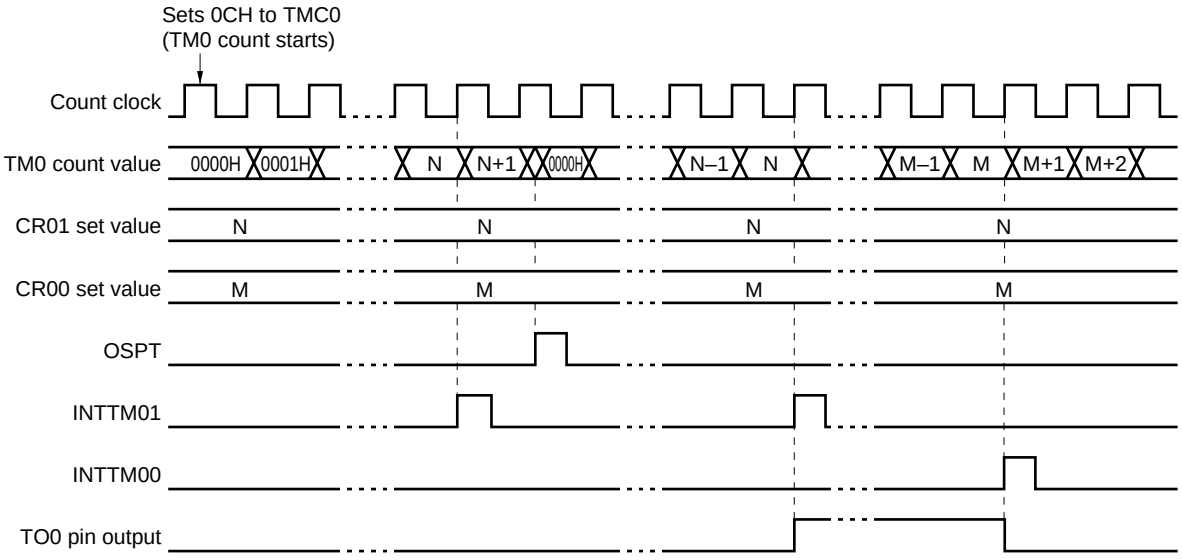
(c) 16-bit timer output control register 0 (TOC0)



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

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with one-shot pulse output. See Figures 6-2, 6-3, and 6-4.

Figure 6-27. Timing of One-Shot Pulse Output Operation Using Software Trigger



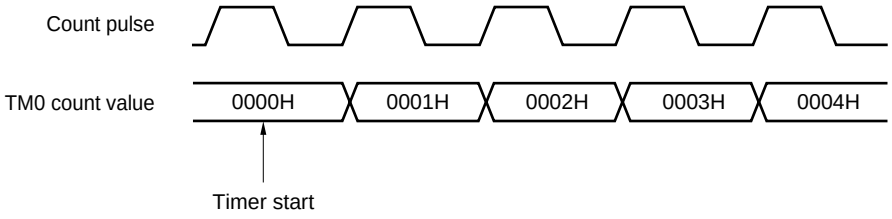
Caution The 16-bit timer/counter 0 (TM0) starts operation at the moment a value other than 0, 0 (operation stop mode) is set to TMC02 and TMC03, respectively.

6.6 16-Bit Timer/Event Counter Operating Precautions

(1) Timer start errors

An error with a maximum of one clock may occur concerning the time required for a match signal to be generated after timer start. This is because 16-bit timer/counter 0 (TM0) is started asynchronously with the count pulse.

Figure 6-28. 16-Bit Timer/Counter 0 (TM0) Start Timing



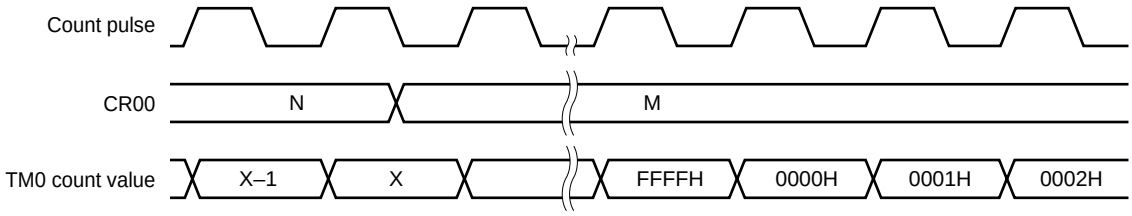
(2) 16-Bit Compare Register Setting

Set other than 0000H to 16-bit timer capture/compare registers 00, 01 (CR00, CR01). This means 1-pulse count operation cannot be performed when it is used as the event counter.

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

If the value after 16-bit timer capture/compare register 00 (CR00) is changed is smaller than that of 16-bit timer/counter 0 (TM0), TM0 continues counting, overflows and then restarts counting from 0. Thus, if the value (M) after CR00 change is smaller than that (N) before change, it is necessary to restart the timer after changing CR00.

Figure 6-29. Timings after Change of Compare Register during Timer Count Operation

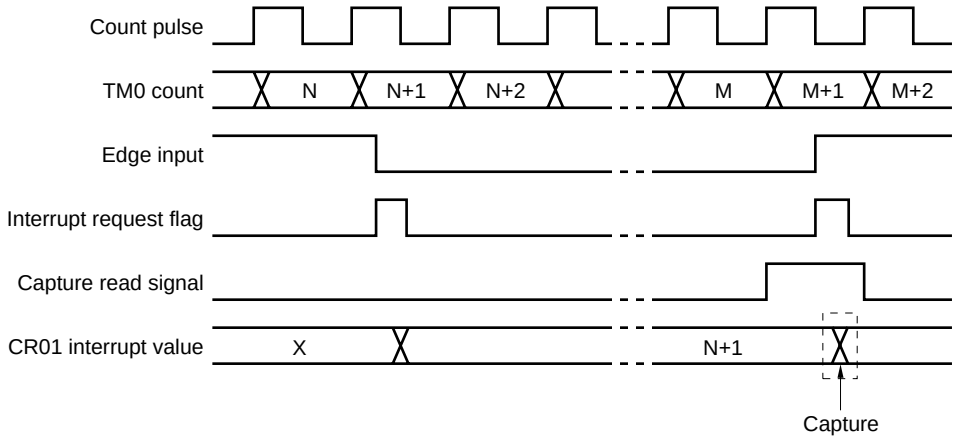


Remark N > X > M

(4) Capture register data retention timings

If the valid edge of the TI00/TO0/P20 pin is input during 16-bit timer capture/compare register 01 (CR01) read, CR01 holds data without carrying out capture operation. However, the interrupt request flag (TMIF01) is set upon detection of the valid edge.

Figure 6-30. Capture Register Data Retention Timing



(5) Valid edge setting

Set the valid edge of the TI00/TO0/P20 pin after setting bits 2 and 3 (TMC02 and TMC03) of 16-bit timer mode control register 0 (TMC0) to 0, 0, respectively, and then stopping timer operation. Valid edge is set with bits 4 and 5 (ES00 and ES01) of the prescaler mode register 0 (PRM0).

(6) Re-trigger of one-shot pulse

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

(7) Operation of OVF0 flag

<1> OFV0 flag is set to 1 in the following case.

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

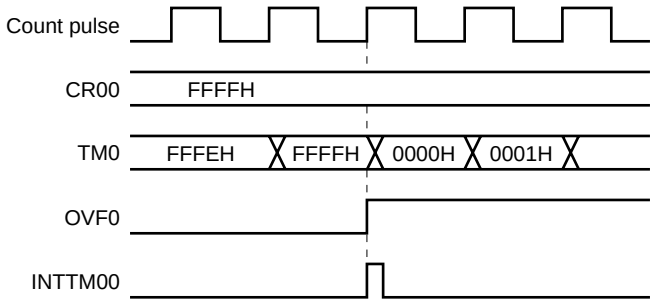
↓

CR00 is set to FFFFH.

↓

When TM0 is counted up from FFFFH to 0000H.

Figure 6-31. Operation Timing of OVF0 Flag



<2> Even if the OVF0 flag is cleared before the next count clock (before TM0 becomes 0001H) after the occurrence of TM0 overflow, the OVF0 flag is reset newly and clear is disabled.

(8) Contending Operations

<1> **The contending operation between the read time of 16-bit timer capture/compare register (CR00/CR01) and capture trigger input (CR00/CR01 used as capture/register)**
Capture/trigger input is prior to the other. The data read from CR00/CR01 is not defined.

<2> **The coincidence timing of contending operation between the write period of 16-bit timer capture/compare register (CR00/CR01) and 16-bit timer/counter 0 (TM0) (CR00/CR01 used as a compare register)**
The coincidence discriminant is not performed normally. Do not write any data to CR00/CR01 near the coincidence timing.

(9) Timer operation

<1> Even if 16-bit timer/counter 0 (TM0) is read, the value is not captured by 16-bit timer capture/compare register 01 (CR01).

<2> Regardless of the CPU's operation mode, when the timer stops, the external interrupt request input noise is not eliminated.

<3> One-shot pulse output operates normally only the free-running mode. In the clear & start mode by TM0 and CR00 match , no overflow occurs, and therefore one-shot pulse output is not possible.

(10) Capture operation

If TI00 is specified as the valid edge of the count clock, capture operation by the capture register specified as the trigger for TI00 is not possible.

(11) Compare operation

<1> When 16-bit timer capture/compare register (CR00/CR01) is overwritten during timer operation, match interrupt may be generated or clear operation may not be performed normally if that value is close to the timer value and larger than the timer value.

<2> Capture operation may not be performed for CR00/CR01 set in compare mode even if a capture trigger has been input.

★ **(12) Edge detection**

<1> To secure capture of the capture trigger, a pulse more than twice the length of the count clock to be selected is required. When TI00 is high level just after system reset, the falling edge is detected just after the TM0 operation is permitted. Be aware of this if using pull-up resistors.

<2> The sampling clock for noise elimination differs according to whether the TI00 valid edge is used as a count clock or as a capture trigger. Sampling is performed by the count clock selected with $f_X/2^3$ for the former, or with prescaler mode register 0 (PRM0) for the latter. Because this counter does not operate until the valid level is detected twice by sampling the valid edge, short-pulse width noise can be eliminated.

[MEMO]

CHAPTER 7 8-BIT TIMER/EVENT COUNTER

7.1 8-Bit Timer/Event Counter Functions

8-bit timer/event counter (TM50, TM51) has the following two modes.

- Mode using 8-bit timer/event counters alone (individual mode)
- Mode using the cascade connection (16-bit resolution: cascade connection mode)

These two modes are described next.

(1) Mode using 8-bit timer/event counters alone (individual mode)

The timer operates as an 8-bit timer/event counter.
It has the following functions.

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

(2) Mode using the cascade connection (16-bit resolution: cascade connection mode)

The timer operates as a 16-bit timer/event counter by connecting in cascade.
It has the following functions.

- Interval timer with 16-bit resolution
- External event counter with 16-bit resolution
- Square wave output with 16-bit resolution

Figures 7-1 and 7-2 show 8-bit timer/event counter block diagrams.

Figure 7-1. 8-Bit Timer/Event Counter 50 Block Diagram

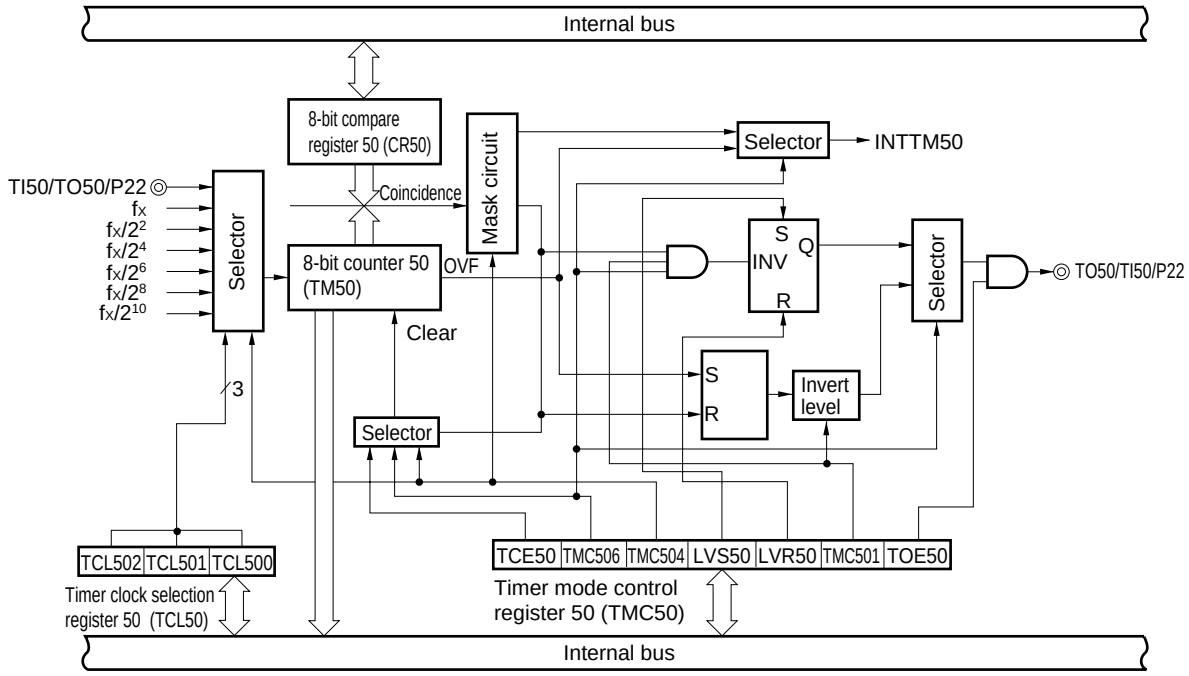
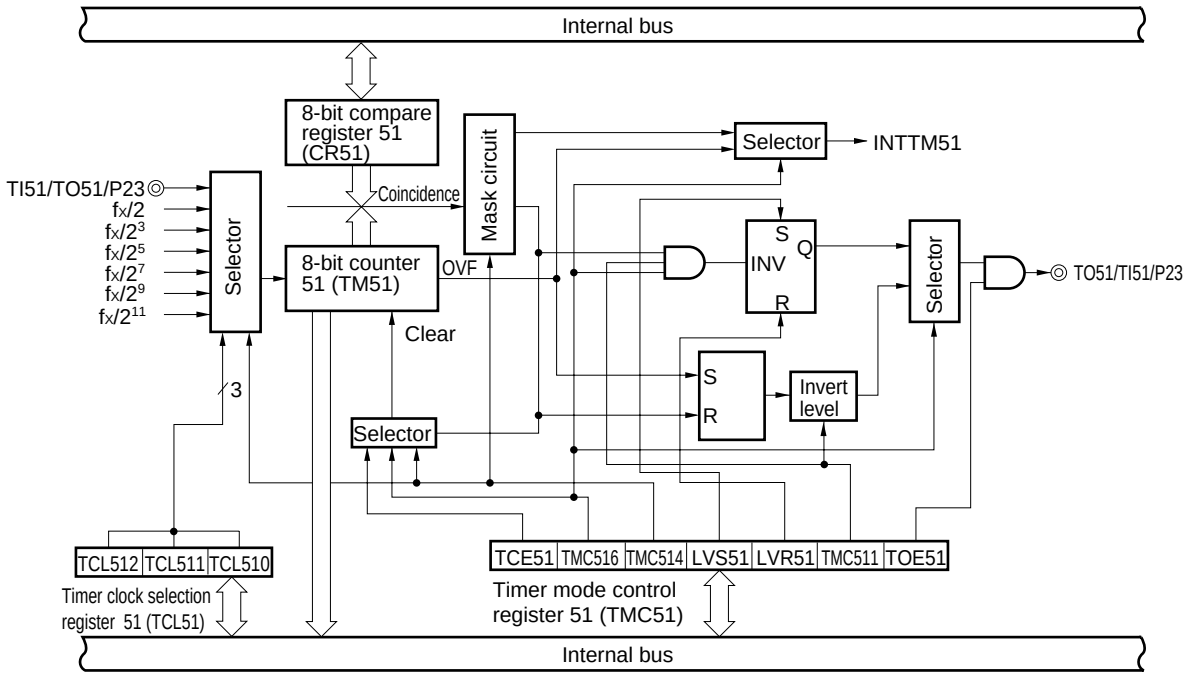


Figure 7-2. 8-Bit Timer/Event Counter 51 Block Diagram



7.2 8-Bit Timer/Event Counter Configurations

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

Table 7-1. 8-Bit Timer/Event Counter Configurations

Item	Configuration
Timer register	8-bit counter 5n (TM5n)
Register	8-bit compare register 5n (CR5n)
Timer output	2 (TO5n)
Control register	Timer clock select register 5n (TCL5n) 8-bit timer mode control register 5n (TMC5n) Port mode register 2 (PM2)**Note**

Note See **Block Diagram of Figure 4-3 P20 to P27.**

Remark n = 0, 1

(1) 8-bit counter 5n (TM5n: n = 0,1)

TM5n is an 8-bit read-only register which counts the count pulses.

When count clock starts, a counter is incremented. TM50 and TM51 can be connected in cascade and used as a 16-bit timer.

When TM50 and TM51 can be connected in cascade and used as a 16-bit timer, they can be read by a 16-bit memory operation instruction. However, since they are connected by an internal 8-bit bus, TM50 and TM51 are read separately in two times. Thus, take read during count change into consideration and compare them in two times reading. When count value is read during operation, count clock input is temporary stopped, and then the count value is read. In the following situations, count value is set to 00H.

<1> $\overline{\text{RESET}}$ input

<2> When TCE5n is cleared

<3> When TM5n and CR5n match in clear & start mode if this mode was entered upon match of TM5n and CR5n values.

Caution In cascade connection mode, the count value is reset to 00H when the lowest timer TCE5n is cleared.

Remark n = 0, 1

(2) 8-bit compare register 5n (CR5n: n = 0, 1)

When CR5n is used as a compare resistor, the value set in CR5n is constantly compared with the 8-bit counter (TM5n) count value, and an interrupt request (INTTM5n) is generated if they match. (Except PWM mode).

It is possible to rewrite the value of CR5n within 00H to FFH during count operation.

When TM50 and TM51 can be connected in cascade and used as a 16-bit timer, CR50 and CR51 operate as the 16-bit compare register. It compares count value with register value, and if the values are matched, interrupt request (INTTM50) are generated. INTTM51 interrupt request is also generated at this time. Thus, when TM50 and TM51 are used as cascade connection, mask INTTM51 interrupt request.

Caution In cascade connection mode, stop the timer operation before setting the data.

Remark n = 0, 1

7.3 Registers to Control 8-Bit Timer/Event Counter

The following three types of registers are used to control 8-bit timer/event counters.

- Timer clock select register 5n (TCL5n)
- 8-bit timer mode control register 5n (TMC5n)
- Port mode register 2 (PM2)

n = 0, 1

(1) Timer clock select register 5n (TCL5n: n = 0, 1)

This register sets count clocks of 8-bit timer/event counter 5n and the valid edge of TI50, TI51 input.

TCL5n is set with an 8-bit memory manipulation instruction.

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

Figure 7-3. Format of Timer Clock Select Register 50 (TCL50)

Address: FF71H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL50	0	0	0	0	0	TCL502	TCL501	TCL500

TCL502	TCL501	TCL500	Count clock selection
0	0	0	TI50 Falling edge
0	0	1	TI50 Rising edge
0	1	0	f_X (8.38 MHz)
0	1	1	$f_X/2^2$ (2.09 MHz)
1	0	0	$f_X/2^4$ (523 kHz)
1	0	1	$f_X/2^6$ (131 kHz)
1	1	0	$f_X/2^8$ (32.7 kHz)
1	1	1	$f_X/2^{10}$ (8.18 kHz)

Cautions 1. When rewriting TCL50 to other data, stop the timer operation beforehand.

2. Set bits 3 to 7 to 0.

Remarks 1. When cascade connection is used, the settings of TCL5n0 to TCL5n2 (n = 0, 1) are valid only for the lowermost timer.

2. f_X: Main system clock oscillation frequency

3. Figures in parentheses are for operation with f_X = 8.38 MHz

Figure 7-4. Format of Timer Clock Select Register 51 (TCL51)

Address: FF75H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510

TCL512	TCL511	TCL510	Count clock selection
0	0	0	TI51 Falling edge
0	0	1	TI51 Rising edge
0	1	0	$f_X/2$ (4.19 MHz)
0	1	1	$f_X/2^3$ (1.04 MHz)
1	0	0	$f_X/2^5$ (261 kHz)
1	0	1	$f_X/2^7$ (65.4 kHz)
1	1	0	$f_X/2^9$ (16.3 kHz)
1	1	1	$f_X/2^{11}$ (4.09 kHz)

Cautions 1. When rewriting TCL51 to other data, stop the timer operation beforehand.

2. Set bit 3 to 7 to 0.

Remarks 1. When cascade connection is used, the settings of TCL5n0 to TCL5n2 (n = 0, 1) are valid only for the lowermost timer.

2. f_X: Main system clock oscillation frequency

3. Figures in parentheses are for operation with f_X = 8.38 MHz

(2) 8-bit Timer Mode Control Register 5n (TMC5n: n = 0, 1)

TMC5n is a register which sets up the following six types.

<1> 8-bit counter 5n (TM5n) count operation control
<2> 8-bit counter 5n (TM5n) operating mode selection
<3> Single mode/cascade connection mode selection
<4> Timer output F/F (flip flop) status setting
<5> Active level selection in timer F/F control or PWM (free-running) mode.
<6> Timer output control

TMC5n is set by a 1-bit memory manipulation instruction or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets to 04H.
Figure 7-5 shows the TMC5n format.

Figure 7-5. Format of 8-Bit Timer Mode Control Register 5n (TMC5n)

Address: FF70H (TMC50) FF74H (TMC51) After reset: 04H R/W

Symbol	7	6	5	4	3	2	1	0
TMC5n	TCE5n	TMC5n6	0	TMC5n4	LVS5n	LVR5n	TMC5n1	TOE5n

TCE5n	TM5n count operation control
0	After cleaning to 0, count operation disabled (prescaler disabled)
1	Count operation start

TMC5n6	TM5n operating mode selection
0	Clear and start mode by matching between TM5n and CR5n
1	PWM (Free-running) mode

TMC5n4	Single mode/cascade connection mode selection
0	Single mode (use the lowest timer)
1	Cascade connection mode (connect to lower timer)

LVS5n	LVR5n	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

TMC5n1	In other modes (TMC5n6 = 0)	In PWM mode (TMC5n6 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active high
1	Inversion operation enabled	Active low

TOE5n	Timer output control
0	Output disabled (Port mode)
1	Output enabled

Remarks 1. In PWM mode, PWM output will be inactive because of TCE5n = 0.

2. If LVS5n and LVR5n are read after data is set, 0 is read.

3. n = 0, 1

(3) Port mode register 2 (PM2)

This register sets port 2 input/output in 1-bit units.

When using the P22/TO50/TI50 and P23/TI51/TO51 pins for timer output, set PM22, PM23, and output latches of P22 and P23 to 0.

PM2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM2 to FFH.

Figure 7-6. Format of Port Mode Register 2 (PM2)

Address: FF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

PM2n	P2n pin input/output mode selection (n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

7.4 8-Bit Timer/Event Counter Operations

7.4.1 8-bit interval timer operation

The 8-bit timer/event counters operate as interval timers which generate interrupt requests repeatedly at intervals of the count value preset to 8-bit compare register 5n (CR5n).

When the count values of 8-bit counter 5n (TM5n) match the values set to CR5n, counting continues with the TM5n values cleared to 0 and the interrupt request signals (INTTM5n) are generated.

The count clock of TM5n can be selected with bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock select register 5n (TCL5n).

See **7.5 8-bit Timer/Event Counter Caution (2) Operation after compare register change during timer count operation** about the operation when the compare register value is changed during timer count operation.

[Setting]

<1> Set the registers.
- TCL5n: Select count clock.
- CR5n: Compare value
- TMC5n: Clear and Start mode by match of TM5n and CR5n.
  (TMC5n = 0000×××0B × = don't care)

<2> After TCE5n = 1 is set, count operation starts.

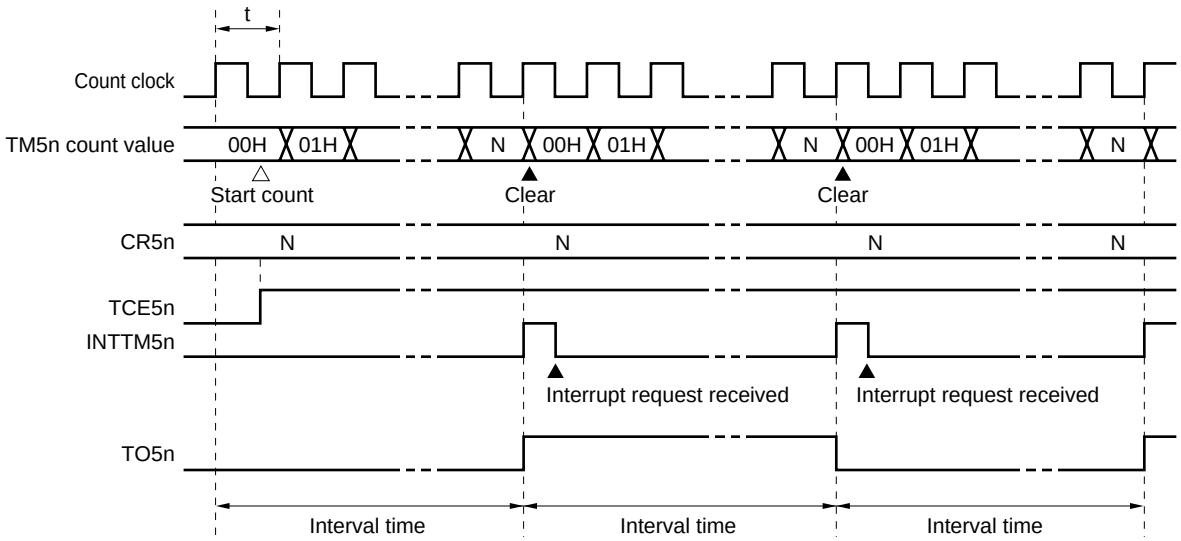
<3> If the values of TM5n and CR5n match, the timer output flip-flop inverts. Also, INTTM5n is generated and TM5n is cleared to 00H.

<4> INTTM5n generates repeatedly at the same interval.
Set TCE5n to 0 to stop count operation.

Remark n = 0, 1

Figure 7-7. Interval Timer Operation Timings (1/3)

(a) Basic operation

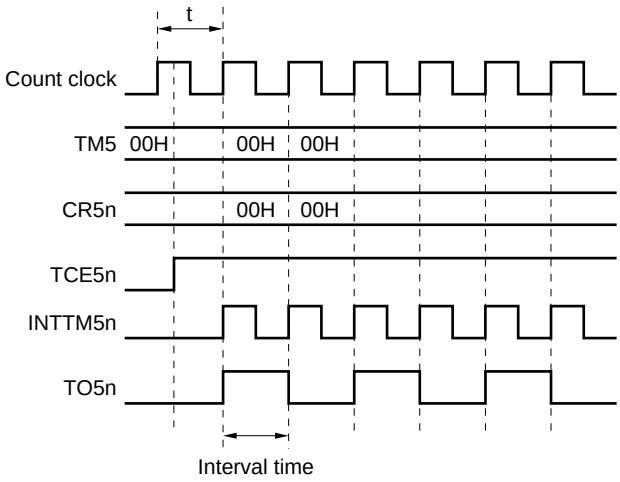


Remarks 1. Interval time = (N + 1) × t: N = 00H to FFH

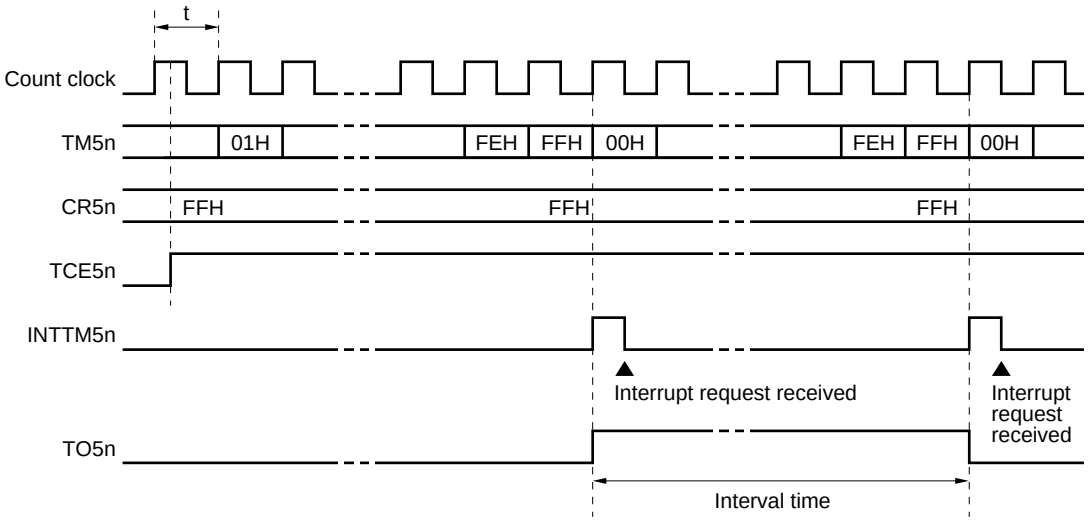
2. n = 0, 1

Figure 7-7. Interval Timer Operation Timings (2/3)

(b) When CR5n = 00H



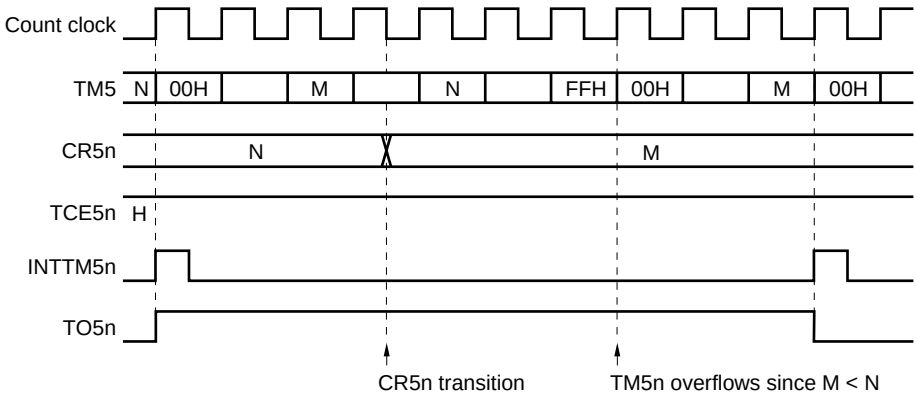
(c) When CR5n = FFH



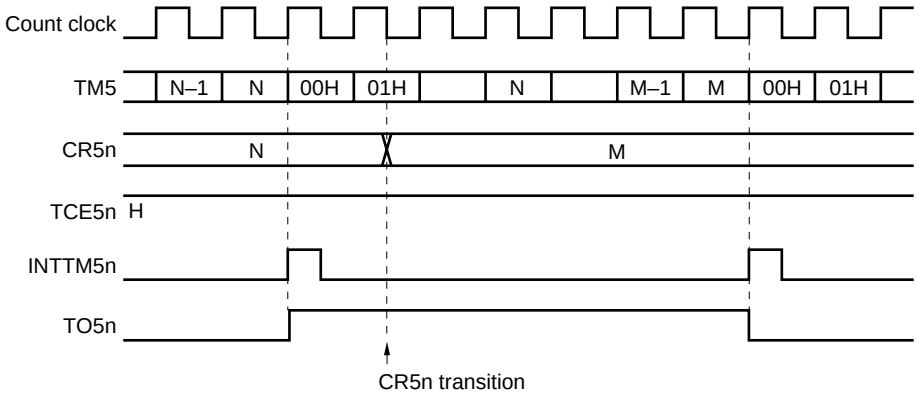
n = 0, 1

Figure 7-7. Interval Timer Operation Timings (3/3)

(d) Operated by CR5n transition (M < N)



(e) Operated by CR5n transition (M > N)



n = 0, 1

7.4.2 External event counter operation

The external event counter counts the number of external clock pulses to be input to TI5n by 8-bit counter 5n (TM5n).

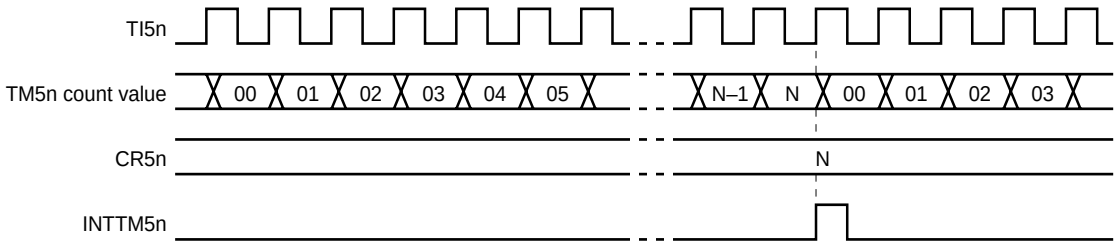
TM5n is incremented each time the valid edge specified with timer clock select register 5n (TCL5n) is input. Either the rising or falling edge can be selected.

When TM5n counted values match the values of 8-bit compare register 5n (CR5n), TM5n is cleared to 0 and the interrupt request signal (INTTM5n) are generated.

Whenever the TM5n counted value matches the value of CR5n, INTTM5n is generated.

Remark n = 0, 1

Figure 7-8. External Event Counter Operation Timings (with Rising Edge Specified)



n = 0, 1

7.4.3. Square-wave output (8-bit resolution) operation

A square wave with any selected frequency is output at intervals of the value preset to the 8-bit compare register 5n (CR5n).

TO5n pin output status is reversed at intervals of the count value preset to CR5n by setting bit 0 (TOE5n) of 8-bit timer mode control register 5n (TMC5n) to 1. This enables a square wave with any selected frequency to be output (duty = 50%).

[Setting]

<1> Set each register

- Set port latch and port mode register to 0.
- TCL5n: Select count clock
- CR5n: Compared value
- TMC5n: Clear and Start mode by match of TM5n and CR5n

LVS5n	LVR5n	Timer output F/F status setting
1	0	High-level output
0	1	Low-level output

Timer output F/F reverse enable
Timer output enable → TOE5n = 1

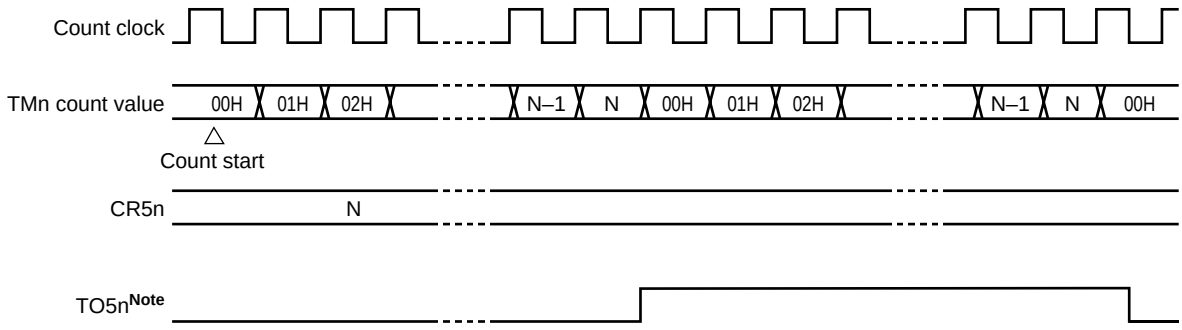
<2> After TCE5n = 1 is set, count operation starts

<3> Timer output F/F is reversed by match of TM5n and CR5n. After INTTM5n is generated, TM5n is cleared to 00H

<4> Timer output F/F is reversed at the same interval and square wave is output from TO5n

Remark n = 0,1

Figure 7-9. Square-Wave Output Operation Timing



Note TO5n output initial value can be set by bits 2 and 3 (LVR5n, LVS5n) of the 8-bit timer mode control register 5n (TMC5n)

Remark n = 0,1

7.4.4 8-bit PWM output operation

The 8-bit timer/event counter operates as PWM output when bit 6 (TMC5n6) of 8-bit timer mode control register 5n (TMC5n) is set to 1.

The duty rate pulse determined by the value set to 8-bit compare register 5n (CR5n).

Set the active level width of PWM pulse to CR5n, and the active level can be selected with bit 1 of TMC5n (TMC5n1).

Count clock can be selected with bit 0 to 2 (TCL5n0 to TCL5n2) of timer clock select register 5n (TCL5n).

Enable/disable for PWM output can be selected with bit 0 of TMC5n (TOE5n).

Caution Rewrite of CR5n in PWM mode is allowed only once in a cycle.

Remark n = 0, 1

(1) PWM output basic operation

[Setting]

<1> Set port latch (P72, 73) and port mode register 7 (PM72, PM73) to 0.
<2> Set active level width with 8-bit compare register (CR5n).
<3> Select count clock with timer clock select register 5n (TCL5n).
<4> Set active level with bit 1 of TMC5n (TMC5n1).
<5> Count operation starts when bit 7 of TMC5n is set to 1.
Set TCE5n to 0 to stop count operation.

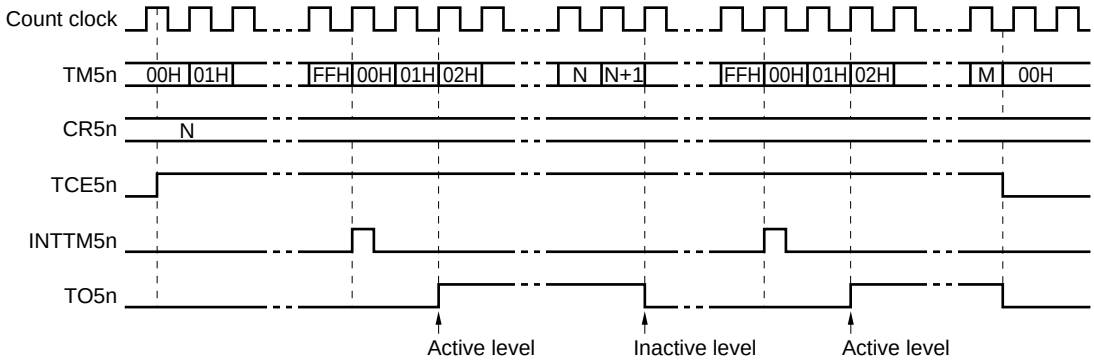
[PWM output operation]

<1> PWM output (output from TO5n) outputs inactive level after count operation starts until overflow is generated.
<2> When overflow is generated, the active level set in <1> of setting is output.
The active level is output until CR5n matches the count value of 8-bit counter 5n (TM5n).
<3> After the CR5n matches the count value, PWM output outputs the inactive level again until overflow is generated.
<4> Operations <2> and <3> are repeated until the count operation stops.
<5> When the count operation is stopped with TCE5n = 0, PWM output comes to inactive level.

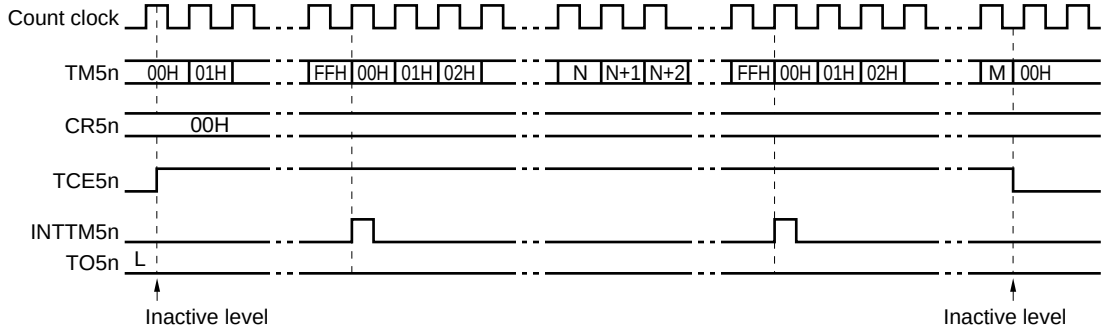
Remark n = 0, 1

Figure 7-10. PWM Output Operation Timing

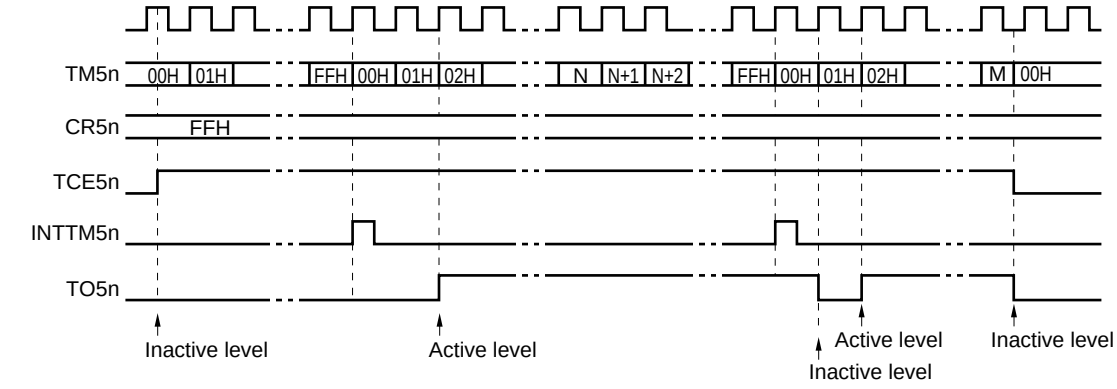
(a) Basic operation (active level = H)



(b) CR5n = 0



(c) CR5n = FFH

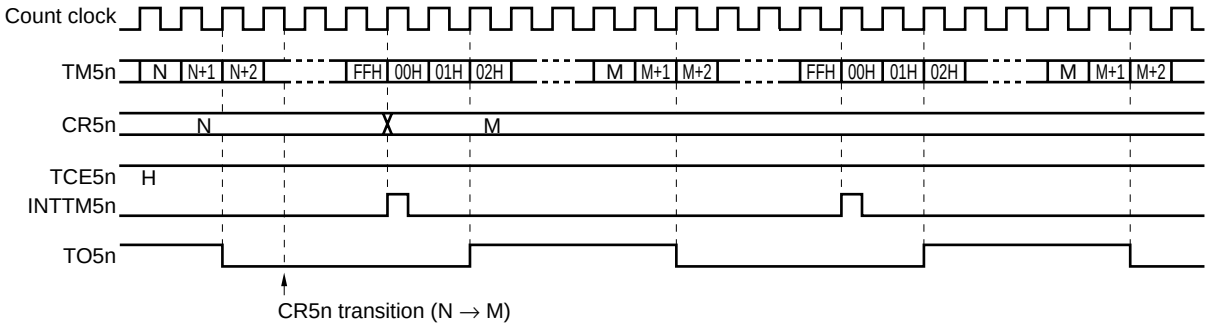


n = 0, 1

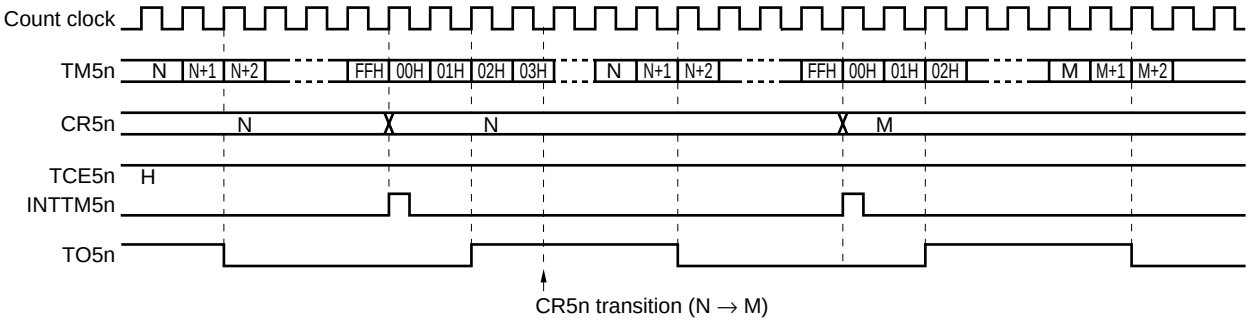
(2) Operated by CR5n transition

Figure 7-11. Timing of Operation by Change of CR5n

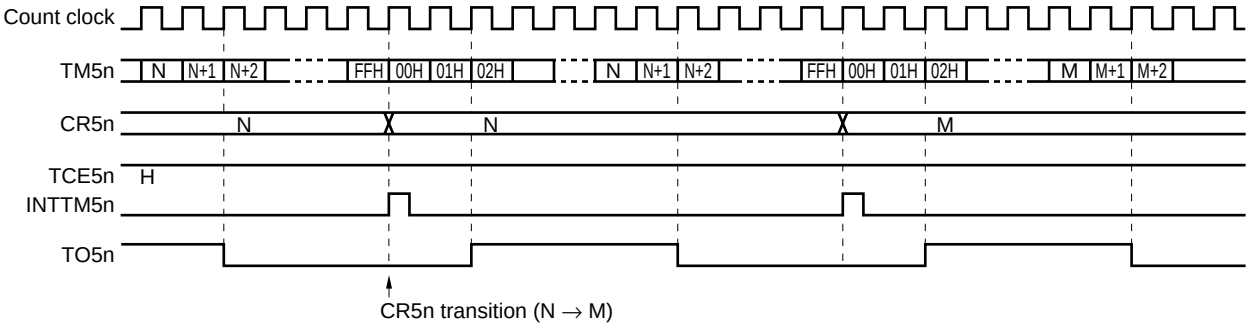
(a) CR5n value transits from N to M before overflow of TM5n



(b) CR5n value transits from N to M after overflow of TM5n



(c) CR5n value transits from N to M between two clocks (00H and 01H) after overflow of TM5n



n = 0, 1

7.4.5 Interval timer (16-bit) operations

When bit 4 (TMC514) of 8-bit timer mode control register 51 (TM51) is set to 1, the 16-bit resolution timer/counter mode is entered.

The 8-bit timer/event counter operates as an interval timer which generates interrupt requests repeatedly at intervals of the count value preset to the 8-bit compare registers (CR50, CR51).

[Setting]

<1> Set each register

TCL50: Select count clock in TM50.
Cascade-connected TM51 need not be selected.

CR50, CR51: Compared value (each value can be set at 00H through FFH)

TMC50, TMC51: Select the clear & start mode by match of TM50 and CR50 (TM51 and CR51).

TM50 → TMC50 = 0000×××0B ×: don't care
TM51 → TMC51 = 0001×××0B ×: don't care

<2> When TMC51 is set to TCE51 = 1 and then, TCE50 is set to TCE50 = 1, count operation starts.

<3> When the values of TM50 and CR50 of cascade-connected timer match, INTTM50 of TM50 is generated. (TM50 and TM51 are cleared to 00H)

<4> INTTM50 generates repeatedly at the same interval.

Cautions 1. Stop timer operation without fail before setting compare register (CR50, CR51).

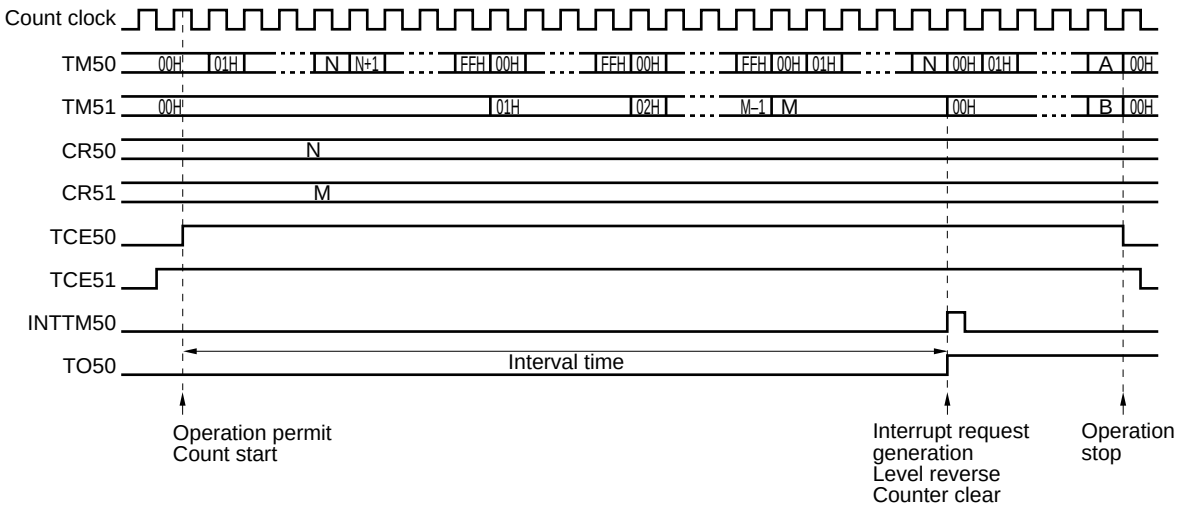
2. INTTM51 of TM51 is generated when TM51 count value matches CR51, even if cascade connection is used. Ensure to mask TM51 to prohibit interrupt.

3. Set TCE50 and TCE51 in a sequential order of TM51 and TM50.

4. Count restart/stop can only be controlled by setting TCE50 of TM50 to 1/0.

Figure 7-12 shows an example of 16-bit resolution cascade connection mode timing.

Figure 7-12. 16-Bit Resolution Cascade Connection Mode

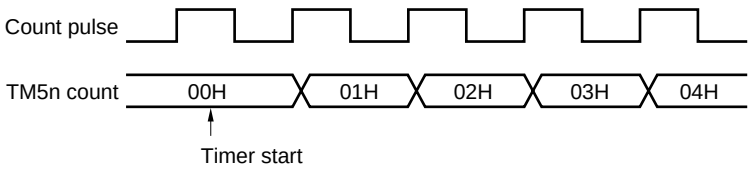


7.5 8-Bit Timer/Event Counter Cautions

(1) Timer start errors

An error with the maximum of one clock may occur concerning the time required for a match signal to be generated after timer start. This is because 8-bit counter 5n (TM5n) is started asynchronously with the count pulse.

Figure 7-13. 8-Bit Counter Start Timing

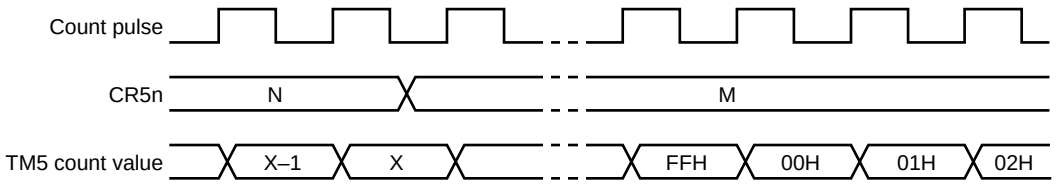


n = 0, 1

(2) Operation after compare register transition during timer count operation

If the value after 8-bit compare register 5n (CR5n) is transmitted is smaller than the value of 8-bit counter 5n (TM5n), TM5n continues counting, overflows and then restarts counting from 0. Thus, if the value (M) after CR5n is smaller than value (N) before transition, it is necessary to restart the timer after transiting CR5n.

Figure 7-14. Timing after Compare Register Transition during Timer Count Operation



Caution Except when the TI5n input is selected, always set TCE5n = 0 before setting the stop state.

Remarks 1. N > X > M

2. n = 0, 1

(3) TM5n (n = 0, 1) reading during timer operation

When reading TM5n during operation, select count clock having high/low level wave form longer than two cycles of CPU clock because count clock stops temporary. For example, in the case where CPU clock (f_{CPU}) is fx, when the selected count clock is fx/4 or below, it can be read.

Remark n = 0, 1

[MEMO]

CHAPTER 8 WATCH TIMER

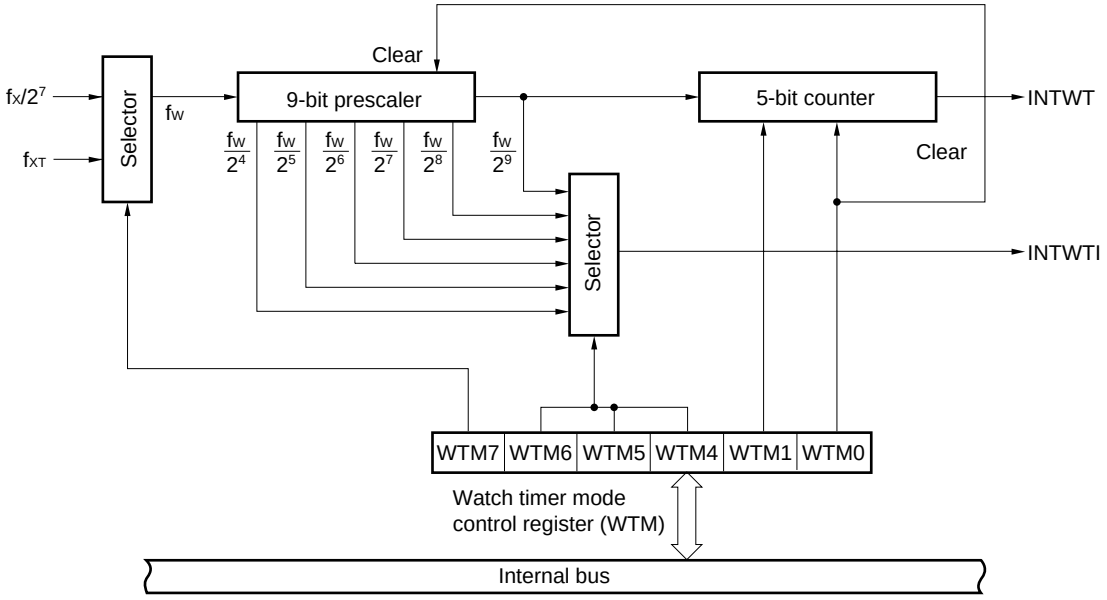
8.1 Watch Timer Functions

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch timer and the interval timer can be used simultaneously.
Figure 8-1 shows the watch timer block diagram.

Figure 8-1. Watch Timer Block Diagram



(1) Watch timer

When the main system clock or subsystem clock is used, interrupt requests (INTWT) are generated at 0.5 second or 0.25 second intervals.

(2) Interval timer

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

Table 8-1. Interval Timer Interval Time

Interval Time		When Operated at f_X = 8.38 MHz	When Operated at f_X = 4.19 MHz	When Operated at f_{XT} = 32.768 kHz
$2^{11} \times 1/f_X$	$2^4 \times 1/f_{XT}$	244 μs	489 μs	488 μs
$2^{12} \times 1/f_X$	$2^5 \times 1/f_{XT}$	489 μs	978 μs	977 μs
$2^{13} \times 1/f_X$	$2^6 \times 1/f_{XT}$	978 μs	1.96 ms	1.95 ms
$2^{14} \times 1/f_X$	$2^7 \times 1/f_{XT}$	1.96 ms	3.91 ms	3.91 ms
$2^{15} \times 1/f_X$	$2^8 \times 1/f_{XT}$	3.91 ms	7.82 ms	7.81 ms
$2^{16} \times 1/f_X$	$2^9 \times 1/f_{XT}$	7.82 ms	15.6 ms	15.6 ms

Remark 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 × 1
Prescaler	9 bits × 1
Control register	Watch timer mode control register (WTM)

8.3 Register to Control Watch Timer

Watch timer mode control register (WTM) is a register to control watch timer.

- **Watch timer mode control register (WTM)**
  This register sets the watch timer count clock, enables/disables operation, prescaler interval time, and 5-bit counter operation control. WTM is set with an 8-bit memory manipulation instruction.
  $\overline{\text{RESET}}$ input sets WTM to 00H.

Figure 8-2. Format of Watch Timer Mode Control Register (WTM)

Address: FF41H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WTM	WTM7	WTM6	WTM5	WTM4	0	0	WTM1	WTM0

WTM7	Watch timer count clock selection
0	$f_X/2^7$ (65.4 kHz)
1	f_{XT} (32.768 kHz)

WTM6	WTM5	WTM4	Prescaler interval time selection
0	0	0	$2^4/f_W$
0	0	1	$2^5/f_W$
0	1	0	$2^6/f_W$
0	1	1	$2^7/f_W$
1	0	0	$2^8/f_W$
1	0	1	$2^9/f_W$
Other than above			Setting prohibited

WTM1	5-bit counter operation control
0	Clear after operation stop
1	Start

WTM0	Watch timer enables operation
0	Operation stop (clear both prescaler and timer)
1	Operation enable

Remarks 1. f_W: Watch timer clock frequency ($f_X/2^7$ or f_{XT})

2. f_X: Main system clock oscillation frequency

3. f_{XT}: Subsystem clock oscillation frequency

4. Figures in parentheses apply to operation with f_X = 8.38 MHz, f_{XT} = 32.768 kHz.

8.4 Watch Timer Operations

8.4.1 Watch timer operation

When the 32.768-kHz subsystem clock or 8.38-MHz main system clock is used, the timer operates as a watch timer with a 0.5-second or 0.25-second interval.

The watch timer generates an interrupt request (INTWT) at the constant time interval.

When bit 0 (WTM0) and bit 1 (WTM1) of the watch timer mode control register (WTM) is set to 1, the 5-bit counter is cleared and the count operation stops.

While the interval timer is simultaneously operating, zero-second start only for the watch timer can be achieved by setting WTM1 to 0. In that case, however, 9-bit prescaler is not cleared, so that an error of $2^9 \times 1/f_W$ at maximum may occur in the first overflow (INTWT) after the zero-second start of watch timer.

8.4.2 Interval timer operation

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

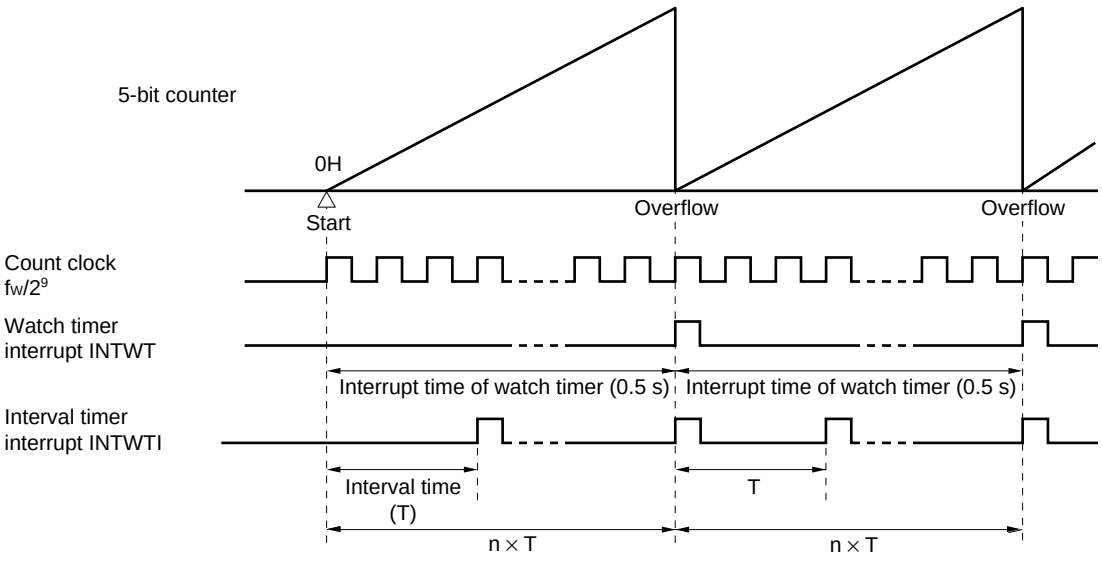
The interval time can be selected with bits 4 to 6 (WTM4 to WTM6) of the watch timer mode control register (WTM).

Table 8-3. Interval Timer Interval Time

WTM6	WTM5	WTM4	Interval Time	When Operated at f_X = 8.38 MHz	When Operated at f_X = 4.19 MHz	When Operated at f_{XT} = 32.768 kHz
0	0	0	$2^4 \times 1/f_W$	244 μs	489 μs	488 μs
0	0	1	$2^5 \times 1/f_W$	489 μs	978 μs	977 μs
0	1	0	$2^6 \times 1/f_W$	978 μs	1.96 ms	1.95 ms
0	1	1	$2^7 \times 1/f_W$	1.96 ms	3.91 ms	3.91 ms
1	0	0	$2^8 \times 1/f_W$	3.91 ms	7.82 ms	7.81 ms
1	0	1	$2^9 \times 1/f_W$	7.82 ms	15.6 ms	15.6 ms
Other than above			Setting prohibited			

Remark f_X: Main system clock oscillation frequency
f_{XT}: Subsystem clock oscillation frequency
f_W: Watch timer clock frequency

Figure 8-3. Operation Timing of Watch Timer/Interval Timer



Remark f_W: Watch timer clock frequency

n: The number of times of interval timer operations

Figures in parentheses are for operation with f_W = 32.768 kHz

[MEMO]

CHAPTER 9 WATCHDOG TIMER

9.1 Watchdog Timer Functions

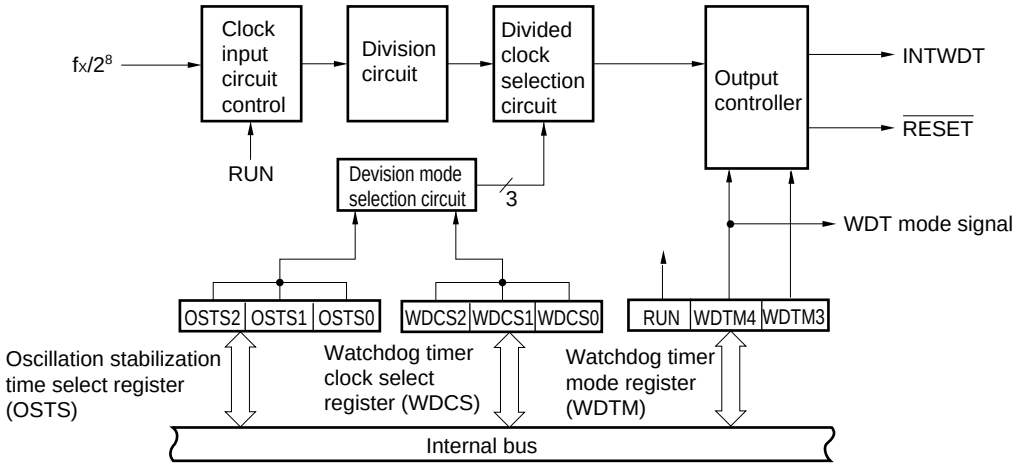
The watchdog timer has the following functions.

- Watchdog timer
- Interval timer
- Oscillation stabilization time selection

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

Figure 9-1 shows a block diagram of the watchdog timer.

Figure 9-1. Watchdog Timer Block Diagram



(1) Watchdog timer mode

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

Table 9-1. Watchdog Timer Runaway Detection Times

Runaway Detection Times
$2^{12} \times 1/f_X$ (489 μs)
$2^{13} \times 1/f_X$ (978 μs)
$2^{14} \times 1/f_X$ (1.96 ms)
$2^{15} \times 1/f_X$ (3.91 ms)
$2^{16} \times 1/f_X$ (7.82 ms)
$2^{17} \times 1/f_X$ (15.6 ms)
$2^{18} \times 1/f_X$ (31.3 ms)
$2^{20} \times 1/f_X$ (125 ms)

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz

(2) Interval timer mode

Interrupt requests are generated at the preset time intervals.

Table 9-2. Interval Times

Interval Time
$2^{12} \times 1/f_X$ (489 μs)
$2^{13} \times 1/f_X$ (978 μs)
$2^{14} \times 1/f_X$ (1.96 ms)
$2^{15} \times 1/f_X$ (3.91 ms)
$2^{16} \times 1/f_X$ (7.82 ms)
$2^{17} \times 1/f_X$ (15.6 ms)
$2^{18} \times 1/f_X$ (31.3 ms)
$2^{20} \times 1/f_X$ (125 ms)

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz

9.2 Watchdog Timer Configuration

The watchdog timer consists of the following hardware.

Table 9-3. Watchdog Timer Configuration

Item	Configuration
Control register	Watchdog timer clock select register (WDCS) Watchdog timer mode register (WDTM) Oscillation stabilization time select register (OSTS)

9.3 Registers to Control the Watchdog Timer

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

- Watchdog timer clock select register (WDCS)
- Watchdog timer mode register (WDTM)
- Oscillation stabilization time select register (OSTS)

(1) Watchdog timer clock select register (WDCS)

This register sets overflow time of the watchdog timer and the interval timer.

WDCS is set by an 8-bit memory manipulation instruction.

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

Figure 9-2. Format of Watchdog Timer Clock Select Register (WDCS)

Address: FF42H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WDCS	0	0	0	0	0	WDCS2	WDCS1	WDCS0

WDCS2	WDCS1	WDCS0	Overflow time of watchdog timer/interval timer
0	0	0	$2^{12}/f_X$ (489 μs)
0	0	1	$2^{13}/f_X$ (978 μs)
0	1	0	$2^{14}/f_X$ (1.96 ms)
0	1	1	$2^{15}/f_X$ (3.91 ms)
1	0	0	$2^{16}/f_X$ (7.82 ms)
1	0	1	$2^{17}/f_X$ (15.6 ms)
1	1	0	$2^{18}/f_X$ (31.3 ms)
1	1	1	$2^{20}/f_X$ (125 ms)

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz

(2) Watchdog timer mode register (WDTM)

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

WDTM is set by 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 (WDTM)

Address: FFF9H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0

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

WDTM4	WDTM3	Watchdog timer operation mode selection[Note 2]
0	×	Interval timer mode[Note 3] (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)

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

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

3. The watchdog timer starts operations as the interval timer when 1 is set to RUN.

Caution When 1 is set to RUN so that the watchdog timer is cleared, the actual overflow time is up to 0.5% shorter than the time set by Watchdog Timer Clock Select Register (WDCS).

Remark ×: don't care

(3) Oscillation Stabilization Time Select Register (OSTS)

A register to select oscillation stabilization time from reset time or STOP mode released time to the time when oscillation is stabilized.

OSTS is set by an 8-bit memory operation instruction.

By $\overline{\text{RESET}}$ input, it is turned into 04H. Thus, when releasing the STOP mode by $\overline{\text{RESET}}$ input, the time required to release is 2^{17}/fx.

Figure 9-4. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFAH After reset: 04H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time
0	0	0	2^{12}/fx (488 μs)
0	0	1	2^{14}/fx (1.95 ms)
0	1	0	2^{15}/fx (3.91 ms)
0	1	1	2^{16}/fx (7.81 ms)
1	0	0	2^{17}/fx (15.6 ms)
Other than above			Setting prohibited

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses are for operation with fx = 8.38 MHz

9.4 Watchdog Timer Operations

9.4.1 Watchdog timer operation

When bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 1, the watchdog timer is operated to detect any runaway.

The runaway detection time interval is selected with bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock select register (WDCS).

Watchdog timer starts by setting bit 7 (RUN) of WDTM to 1. After the watchdog timer is started, set RUN to 1 within the set runaway time interval. The watchdog timer can be cleared and counting is started by setting RUN to 1. If RUN is not set to 1 and the runaway detection time is exceeded, system reset or a non-maskable interrupt request is generated according to WDTM bit 3 (WDTM3) value.

The watchdog timer continues operating in the HALT mode but it stops in the STOP mode. Thus, set RUN to 1 before the STOP mode is set, clear the watchdog timer and then execute the STOP instruction.

Cautions 1. The actual runaway detection time may be shorter than the set time by a maximum of 0.5%.

2. When the subsystem clock is selected for CPU clock, watchdog timer count operation is stopped.

Table 9-4. Watchdog Timer Runaway Detection Time

Runaway Detection Time
$2^{12} \times 1/f_X$ (489 μs)
$2^{13} \times 1/f_X$ (978 μs)
$2^{14} \times 1/f_X$ (1.96 ms)
$2^{15} \times 1/f_X$ (3.91 ms)
$2^{16} \times 1/f_X$ (7.82 ms)
$2^{17} \times 1/f_X$ (15.6 ms)
$2^{18} \times 1/f_X$ (31.3 ms)
$2^{20} \times 1/f_X$ (125 ms)

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz.

9.4.2 Interval timer operation

The watchdog timer operates as an interval timer which generates interrupt requests repeatedly at an interval of the preset count value when bit 3 (WDTM3) and bit 4 (WDTM4) of the watchdog timer mode register (WDTM) are set to 1 and 0, respectively.

The interval time of interval timer is selected with bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock select register (WDCS). When 1 is set to bit 7 (RUN) of WDTM, the watchdog timer operates as the interval timer.

When the watchdog timer operated as the interval timer, the interrupt mask flag (WDTMK) and priority specify flag (WDTPR) are validated and the maskable interrupt request (INTWDT) can be generated. Among maskable interrupts, INTWDT has the highest priority at default.

The interval timer continues operating in the HALT mode but it stops in 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 (this selects the watchdog timer mode), 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 CPU clock, watchdog timer count operation is stopped.

Table 9-5. Interval Timer Interval Time

Interval Time
$2^{12} \times 1/f_X$ (489 μs)
$2^{13} \times 1/f_X$ (978 μs)
$2^{14} \times 1/f_X$ (1.96 ms)
$2^{15} \times 1/f_X$ (3.91 ms)
$2^{16} \times 1/f_X$ (7.82 ms)
$2^{17} \times 1/f_X$ (15.6 ms)
$2^{18} \times 1/f_X$ (31.3 ms)
$2^{20} \times 1/f_X$ (125 ms)

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz.

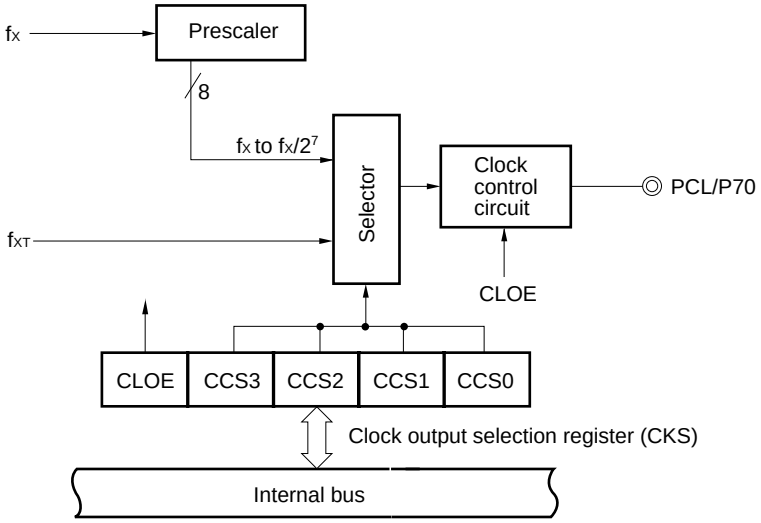
CHAPTER 10 CLOCK OUTPUT CONTROL CIRCUITS

10.1 Clock Output Control Circuit Functions

The clock output control circuit is intended for carrier output during remote controlled transmission and clock output for supply to peripheral LSIs. The clock selected with the clock output selection register (CKS) is output.

Figure 10-1 shows the block diagram of clock output control circuits.

Figure 10-1. Clock Output Control Circuit Block Diagram



10.2 Clock Output Control Circuit Configuration

The clock output control circuits consists of the following hardware.

Table 10-1. Configuration of Clock Output Control Circuits

Item	Configuration
Control register	Clock output select register (CKS) Port mode register (PM7)[Note]

Note See **Block Diagram of Figure 4-8. P70 to P73, P75**.

10.3 Register to Control Clock Output Control Circuit

The following two types of registers are used to control the clock output control circuits.

- Clock output select register (CKS)
- Port mode register (PM7)

(1) Clock output select register (CKS)

This register sets output enable/disable for clock output (PCL) and sets the output clock.

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

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

Figure 10-2. Format of Clock Output Select Register (CKS)

Address: FF40H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKS	0	0	0	CLOE	CCS3	CCS2	CCS1	CCS0

CLOE	PCL output enable/disable setting
0	Stop clock division circuit operation. PCL fixed to low level
1	Enable clock division circuit operation. PCL output enabled.

CCS3	CCS2	CCS1	CCS0	PCL output clock selection
0	0	0	0	f_X (8.38 MHz)
0	0	0	1	$f_X/2$ (4.19 MHz)
0	0	1	0	$f_X/2^2$ (2.09 MHz)
0	0	1	1	$f_X/2^3$ (1.04 MHz)
0	1	0	0	$f_X/2^4$ (524 kHz)
0	1	0	1	$f_X/2^5$ (262 kHz)
0	1	1	0	$f_X/2^6$ (131 kHz)
0	1	1	1	$f_X/2^7$ (65.5 kHz)
1	0	0	0	f_{XT} (32.768 kHz)
Other than above				Setting prohibited

Remarks 1. f_X: main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. Figures in parentheses are for operation with f_X = 8.38 MHz or f_{XT} = 32.768 kHz.

(2) Port mode register (PM7)

This register sets port 7 input/output in 1-bit units.

When using the P70/PCL pin for clock output, set PM70 and the output latch of P70 to 0.

PM7 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM7 to FFH.

Figure 10-3. Format of Port Mode Register 7 (PM7)

Address: FF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	P7n pin input/output mode selection (n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

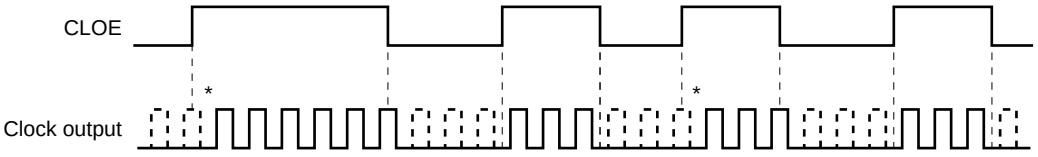
10.4 Clock Output Control Circuit Operations

The clock pulse is output as the following procedure.

<1> Select the clock pulse output frequency with bits 0 to 3 (CCS0 to CCS3) of the clock output selection register (CKS) (clock pulse output in disabled status).
<2> Set bit 4 (CLOE) of CKS to 1, and enable clock output.

Remark The clock output control circuit is designed not to output pulses with a small width during output enable/disable switching of the clock output. As shown in Figure 10-4, be sure to start output from the low period of the clock (marked with ***** in the figure). When stopping output, do so after securing high level of the clock.

Figure 10-4. Remote Control Output Application Example



CHAPTER 11 A/D CONVERTER

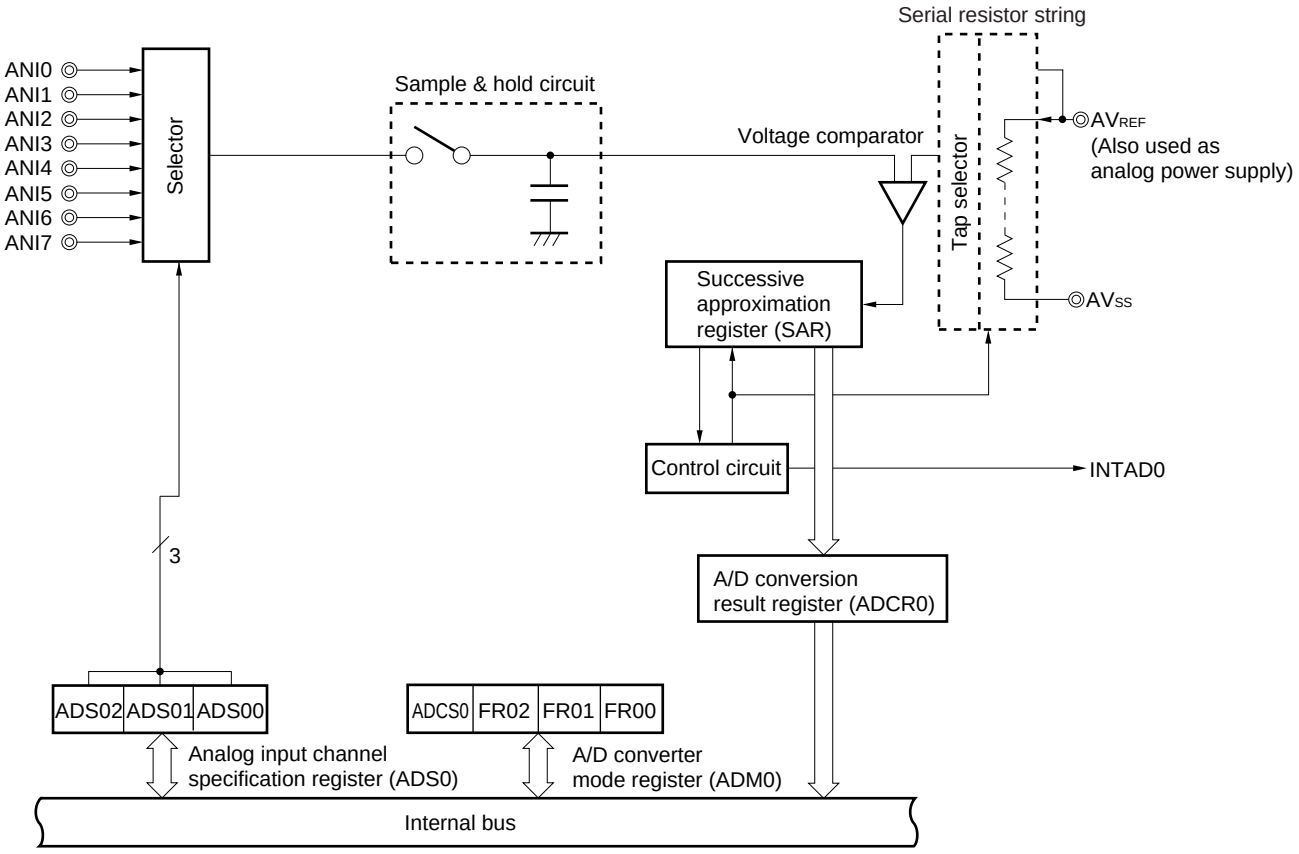
11.1 A/D Converter Functions

A/D converter is an 8-bit resolution converter that converts analog inputs into digital values. It can control up to 8 analog input channels (ANI0 to ANI7).

A/D conversion operation is started by setting the A/D converter mode register (ADM0).

Select one channel for analog input from ANI0 to ANI7 to perform A/D conversion. A/D conversion operation is repeated. Each time an A/D conversion operation ends, an interrupt request (INTAD0) is generated.

Figure 11-1. A/D Converter Block Diagram



11.2 A/D Converter Configuration

The A/D converter consists of the following hardware.

Table 11-1. A/D Converter Configuration

Item	Configuration
Analog input	8 channels (ANI0 to ANI7)
Registers	Successive approximation register (SAR) A/D conversion result register (ADCR0)
Control register	A/D converter mode register (ADM0) Analog input channel specification register (ADS0)

(1) Successive approximation register (SAR)

This register compares the analog input voltage value to the voltage tap (compare value) 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 (ADCR0).

(2) A/D conversion result register (ADCR0)

The ADCR0 is an 8-bit register that stores the A/D conversion result. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register.

ADCR0 is read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADCR0 to 00H.

Caution When writing is performed to the A/D converter mode register (ADM0) and analog input channel specification register (ADS0), the contents of ADCR0 may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS0. Using a timing other than the above may cause an incorrect conversion result to be read.

(3) Sample & hold circuit

The sample & hold circuit samples each analog input signal sequentially applied from the input circuit, and sends it to the voltage comparator. This circuit holds the sampled analog input voltage value during A/D conversion.

(4) Voltage comparator

The voltage comparator compares the analog input to the series resistor string output voltage.

(5) Series resistor string

The series resistor string is connected between AV_{REF} and AV_{SS}, and generates a voltage to be compared to the analog input.

(6) ANI0 to ANI7 pins

These are eight analog input pins to input analog signals to undergo A/D conversion to the A/D converter. ANI0 to ANI7 are alternate-function pins that can also be used for digital input.

Cautions 1. Use ANI0 to ANI7 input voltages within the specification range. If a voltage higher than AV_{REF} or lower than AV_{SS} is applied (even if within the absolute maximum rating range), the conversion value of that channel will be undefined and the conversion values of other channels may also be affected.

2. When a digital pulse is applied to a pin adjacent to the pin in the process of A/D conversion, A/D conversion values may not be obtained as expected due to coupling noise. Thus, do not apply any pulse to a pin adjacent to the pin in the process of A/D conversion.

(7) AV_{REF} pin

This pin inputs the A/D converter reference voltage. It can also be used as analog power supply.

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 approx. 10 kΩ is connected between the AV_{REF} pin and AV_{SS} pin. Therefore, when the output impedance of the reference voltage is too high, it seems as if the AV_{REF} pin and the series resistor string are connected in parallel. This may cause a greater reference voltage error.

(8) AV_{SS} pin

This is the GND potential pin of the A/D converter. Always keep it at the same potential as the V_{SS0} pin even when not using the A/D converter.

11.3 Registers to Control A/D Converter

The following 2 types of registers are used to control the A/D converter.

- A/D converter mode register (ADM0)
- Analog input channel specification register (ADS0)

(1) A/D converter mode register (ADM0)

This register sets the conversion time for analog input to be A/D converted and conversion start/stop.

ADM0 is set by an 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADM0 to 00H.

Figure 11-2. Format of A/D Converter Mode Register (ADM0)

Address: FF80H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADM0	ADCS0	0	FR02	FR01	FR00	0	0	0

ADCS0	A/D conversion operation control
0	Stop conversion operation
1	Enable conversion operation

FR02	FR01	FR00	Conversion time selection[Note 1]
0	0	0	144/f_X (17.1 μs)
0	0	1	120/f_X (14.3 μs)
0	1	0	96/f_X (Setting prohibited[Note 2])
1	0	0	72/f_X (Setting prohibited[Note 2])
1	0	1	60/f_X (Setting prohibited[Note 2])
1	1	0	48/f_X (Setting prohibited[Note 2])
Other than above			Setting prohibited

Notes 1. Set so that the A/D conversion time is 14 μs or more.

2. Setting prohibited because A/D conversion time is less than 14 μs.

Caution When rewriting FR00 to FR02 to other than the same data, stop A/D conversion operations once prior to performing rewrite.

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz.

(2) Analog input channel specification register (ADS0)

This register specifies the analog voltage input port for A/D conversion.

ADS0 is set by an 8-bit memory manipulation.

$\overline{\text{RESET}}$ input sets ADS0 to 00H.

Figure 11-3. Format of Analog Input Channel Specification Register (ADS0)

Address: FF81H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS0	0	0	0	0	0	ADS02	ADS01	ADS00

ADS02	ADS01	ADS00	Analog input channel specification
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

11.4 A/D Converter Operations

11.4.1 Basic operations of A/D converter

<1> Select one channel for A/D conversion with the analog input channel specification register (ADS0).

<2> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<3> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the input analog voltage is held until the A/D conversion operation is ended.

<4> Bit 7 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF} by the tap selector.

<5> The voltage difference between the series resistor string voltage tap and analog input is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{REF}, the MSB of SAR remains set. If the analog input is smaller than (1/2) AV_{REF}, the MSB is reset.

<6> Next, bit 6 of SAR is automatically set, and the operation proceeds to the next comparison. 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) AF_{REF}

The voltage tap and analog input voltage are compared and bit 6 of SAR is manipulated as follows.

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

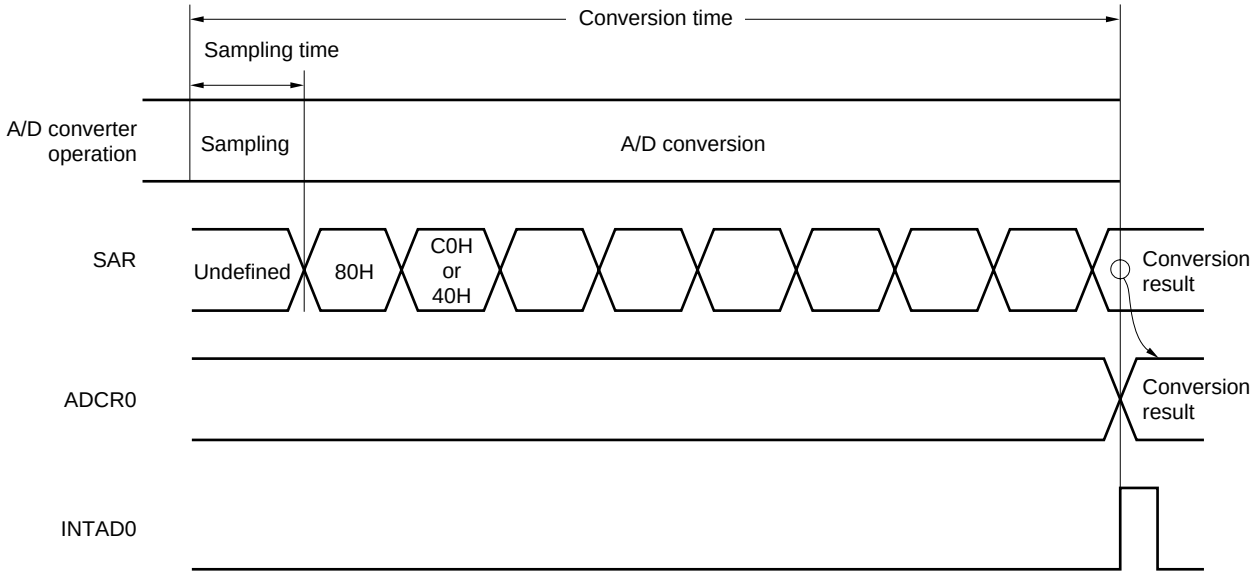
<7> Comparison is continued in this way up to bit 0 of SAR.

<8> Upon completion of the comparison of 8 bits, an effective digital result value remains in SAR, and the result value is transferred to and latched in the A/D conversion result register (ADCR0).

At the same time, the A/D conversion end interrupt request (INTAD0) can also be generated.

Caution The first A/D conversion value just after A/D conversion operations start may not fall within the rating.

Figure 11-4. Basic Operation of A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS0) of the A/D converter mode register (ADM0) is reset (0) by software.

If a write operation is performed to the ADM0 or the analog input channel specification register (ADS0) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS0 bit is set (1), conversion starts again from the beginning.

$\overline{\text{RESET}}$ input sets the A/D conversion result register (ADCR0) to 00H.

11.4.2 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 (stored in the A/D conversion result register (ADCR0)) is shown by the following expression.

$$ADCR0 = INT\left(\frac{V_{IN}}{AV_{REF}} \times 256 + 0.5\right)$$

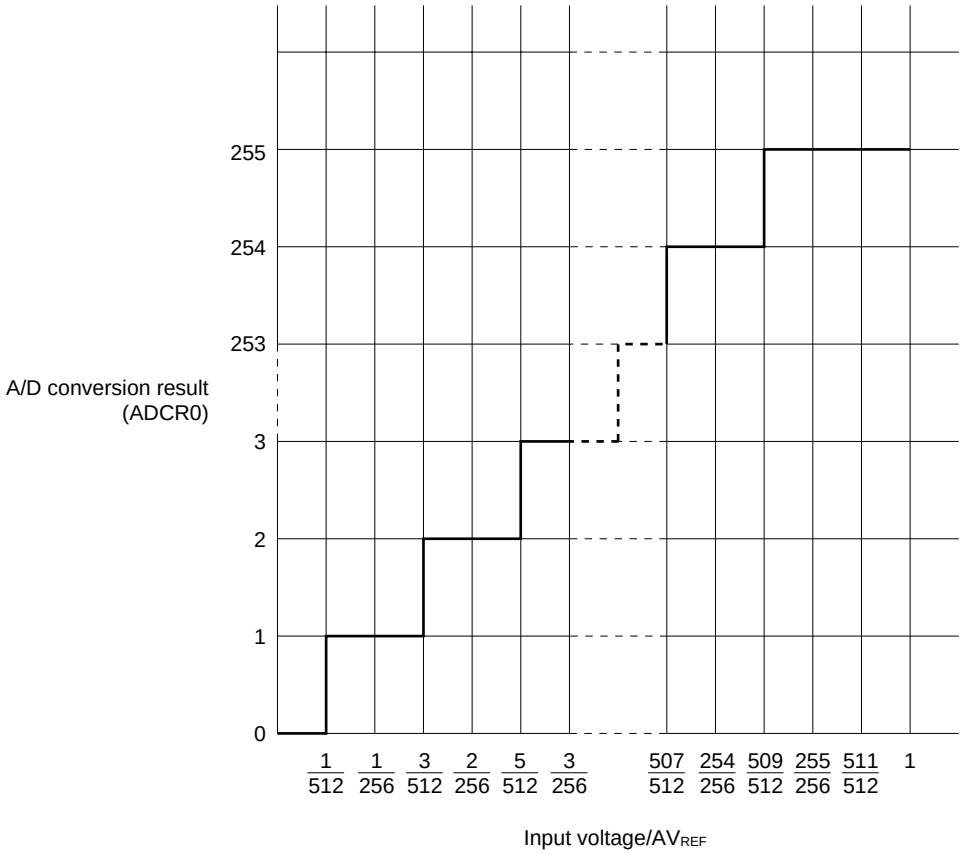
or

$$(ADCR0 - 0.5) \times \frac{AV_{REF}}{256} \leq V_{IN} < (ADCR0 + 0.5) \times \frac{AV_{REF}}{256}$$

where, INT( ): Function which returns integer part of value in parentheses
V_{IN}: Analog input voltage
AV_{REF}: AV_{REF} pin voltage
ADCR0: A/D conversion result register (ADCR0) value

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

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



11.4.3 A/D converter operation mode

One analog input channel is selected from among ANI0 to ANI7 by the analog input channel specification register (ADS0) and start A/D conversion.

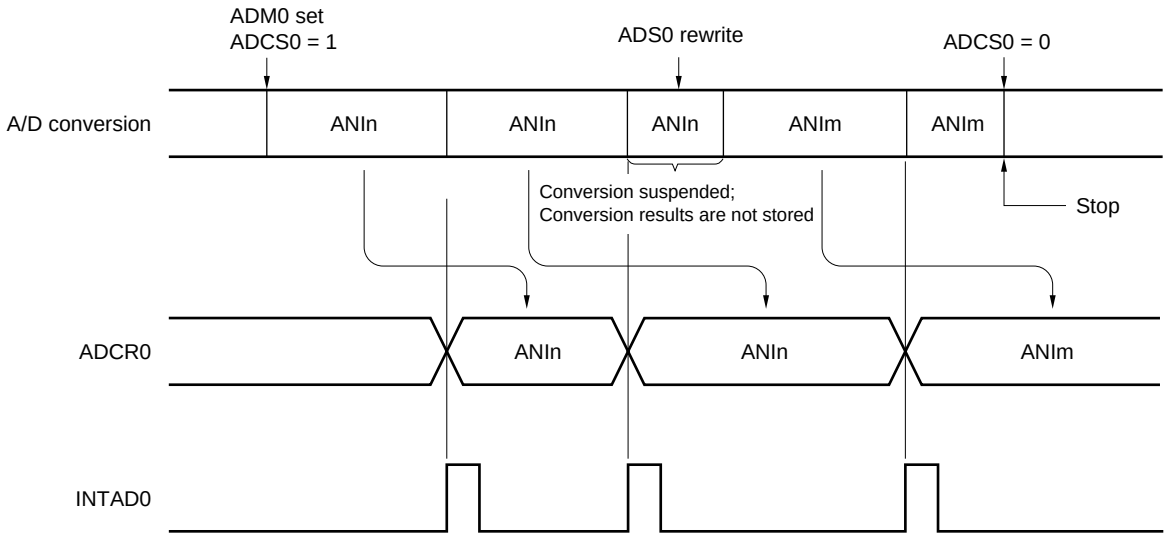
When bit 7 (ADCS0) of the A/D converter mode register (ADM0) is set to 1, A/D conversion of the voltage applied to the analog input pin specified by the analog input channel specification register (ADS0) starts.

Upon the end of the A/D conversion, the conversion result is stored in the A/D conversion result register (ADCR0), and the interrupt request signal (INTAD0) is generated. After one A/D conversion operation is started and ended, the next conversion operation is immediately started. A/D conversion operations are repeated until new data is written to ADS0.

If ADS0 is rewritten during A/D conversion, the converter suspends A/D conversion and A/D conversion of the newly selected analog input channel is started.

If data with ADCS0 set to 0 is written to ADM0 during A/D conversion operation, the A/D conversion operation stops immediately.

Figure 11-6. A/D Conversion



Remarks 1. n = 0, 1, ..., 7

2. m = 0, 1, ..., 7

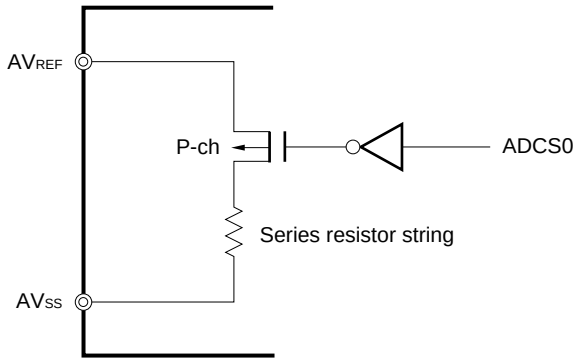
11.5 A/D Converter Cautions

(1) Current consumption in standby mode

A/D converter stops operating in the standby mode. At this time, current consumption can be reduced by stopping the conversion operation (by setting bit 7 (ADCS0) of the A/D converter mode register (ADM0) to 0).

Figure 11-7 shows how to reduce the current consumption in the standby mode.

Figure 11-7. Example of Method of Reducing Current Consumption in Standby Mode



(2) Input range of ANI0 to ANI7

The input voltages of ANI0 to ANI7 should be within the specification range. In particular, if a voltage higher than AV_{REF} or lower than AV_{SS} is input (even if within the absolute maximum rating range), the conversion value of that channel will be undefined and the conversion values of other channels may also be affected.

(3) Contending operations

<1> Contention between A/D conversion result register (ADCR0) write and ADCR0 read by instruction upon the end of conversion

ADCR0 read is given priority. After the read operation, the new conversion result is written to ADCR0.

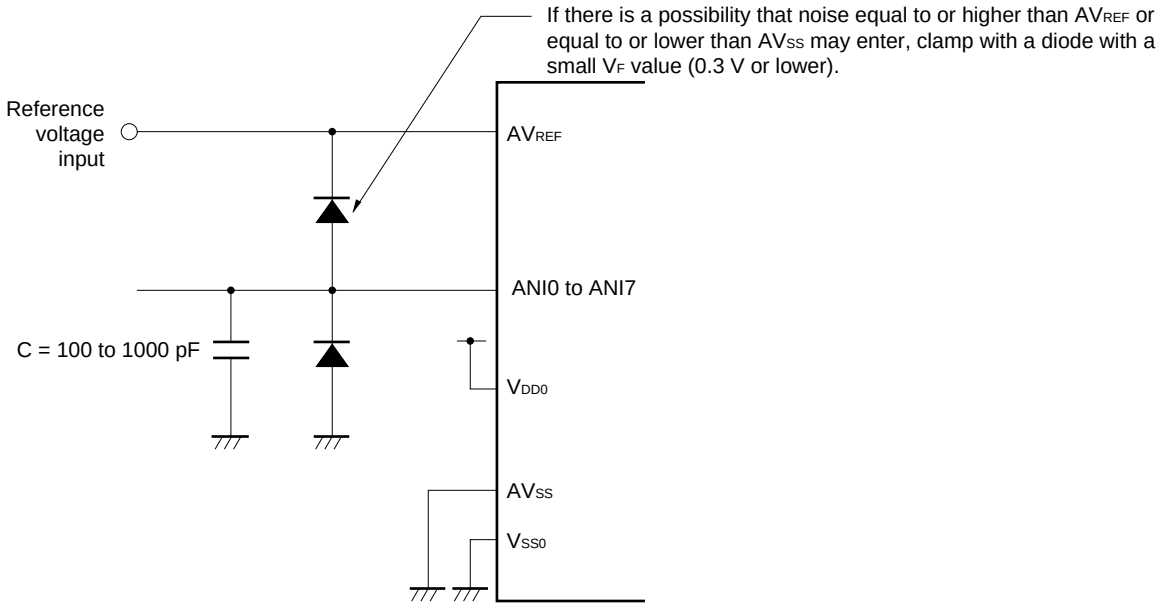
<2> Contention between ADCR0 write and A/D converter mode register (ADM0) write or analog input channel specification register (ADS0) write

ADM0 or ADS0 write is given priority. ADCR0 write is not performed, nor is the conversion end interrupt request signal (INTAD0) generated.

(4) Noise countermeasures

To maintain the 8-bit resolution, attention must be paid to noise input to pin AV_{REF} and pins ANI0 to ANI7. Because 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 11-8 to reduce noise.

Figure 11-8. Analog Input Pin Connection



(5) ANI0 to ANI7

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.

(6) AV_{REF} pin input impedance

A series resistor string of approx. 10 kΩ is connected between the AV_{REF} pin and the AV_{SS} pin.

Therefore, when the output impedance of the reference voltage is too high, it seems as if the AV_{REF} pin and the series resistor string are connected in parallel. This may cause a greater reference voltage error.

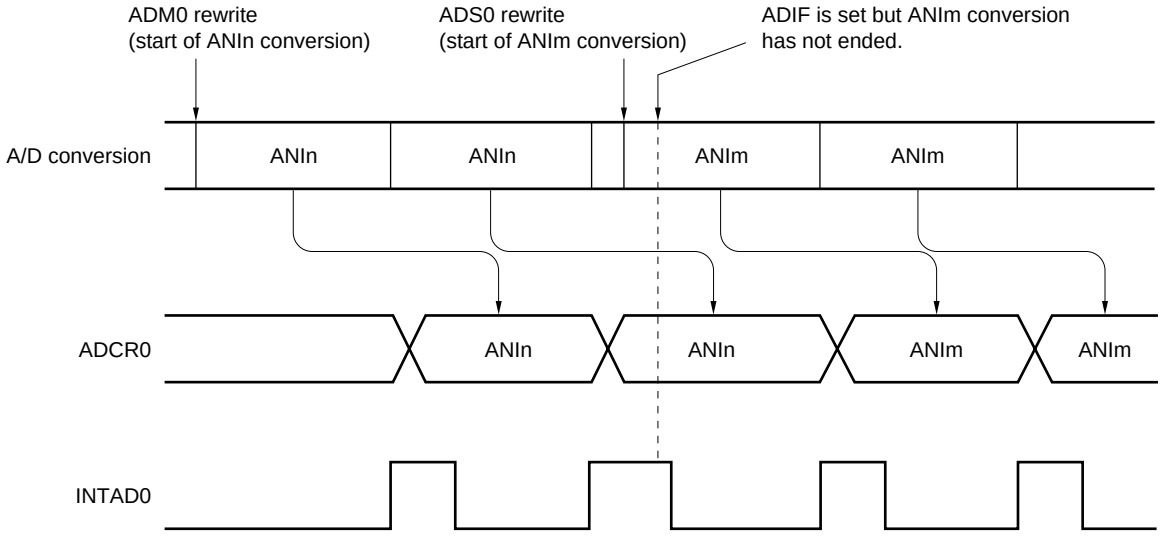
(7) Interrupt request flag (ADIF0)

The interrupt request flag (ADIF0) is not cleared even if the analog input channel specification register (ADS0) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and conversion end interrupt request flag for the pre-change analog input may be set just before the ADS0 rewrite. Caution is therefore required since, at this time, when ADIF0 is read immediately just after the ADS0 rewrite, ADIF0 is set despite the fact that the A/D conversion for the post-change analog input has not ended.

When the A/D conversion is stopped and then resumed, clear ADIF0 before the A/D conversion operation is resumed.

Figure 11-9. A/D Conversion End Interrupt Request Generation Timing



Remarks 1. n = 0, 1, ..., 7

2. m = 0, 1, ..., 7

(8) Conversion results just after A/D conversion start

The first A/D conversion value just after A/D conversion operations start may not fall within the rating. Polling A/D conversion end interrupt request (INTAD0) and take measures such as removing the first conversion results.

(9) A/D conversion result register (ADCR0) read operation

When writing is performed to the A/D converter mode register (ADM0) and analog input channel specification register (ADS0), the contents of ADCR0 may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS0. Using a timing other than the above may cause an incorrect conversion result to be read.

CHAPTER 12 SERIAL INTERFACE OUTLINE

The μPD780065 Subseries is equipped with four channels of on-chip serial interfaces. The outline of the serial interfaces are listed in Table 12-1. For details, refer to the respective chapter.

Table 12-1. Outline of On-Chip Serial Interface of the μPD780065 Subseries

Channel	Serial Transfer Mode
UART0	UART (Asynchronous serial interface)
SIO1	3-wire serial I/O (Automatic transmission/reception function) MSB/LSB first switchable
SIO30	2-wire serial I/O Fixed to MSB first
SIO31	3-wire serial I/O Fixed to MSB first

[MEMO]

CHAPTER 13 SERIAL INTERFACE (UART0)

13.1 Serial Interface (UART0) Functions

The serial interface (UART0) has the following three modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed to reduce power consumption.
For details, see **13.4.1 Operation stop mode**.

(2) Asynchronous serial interface (UART) mode

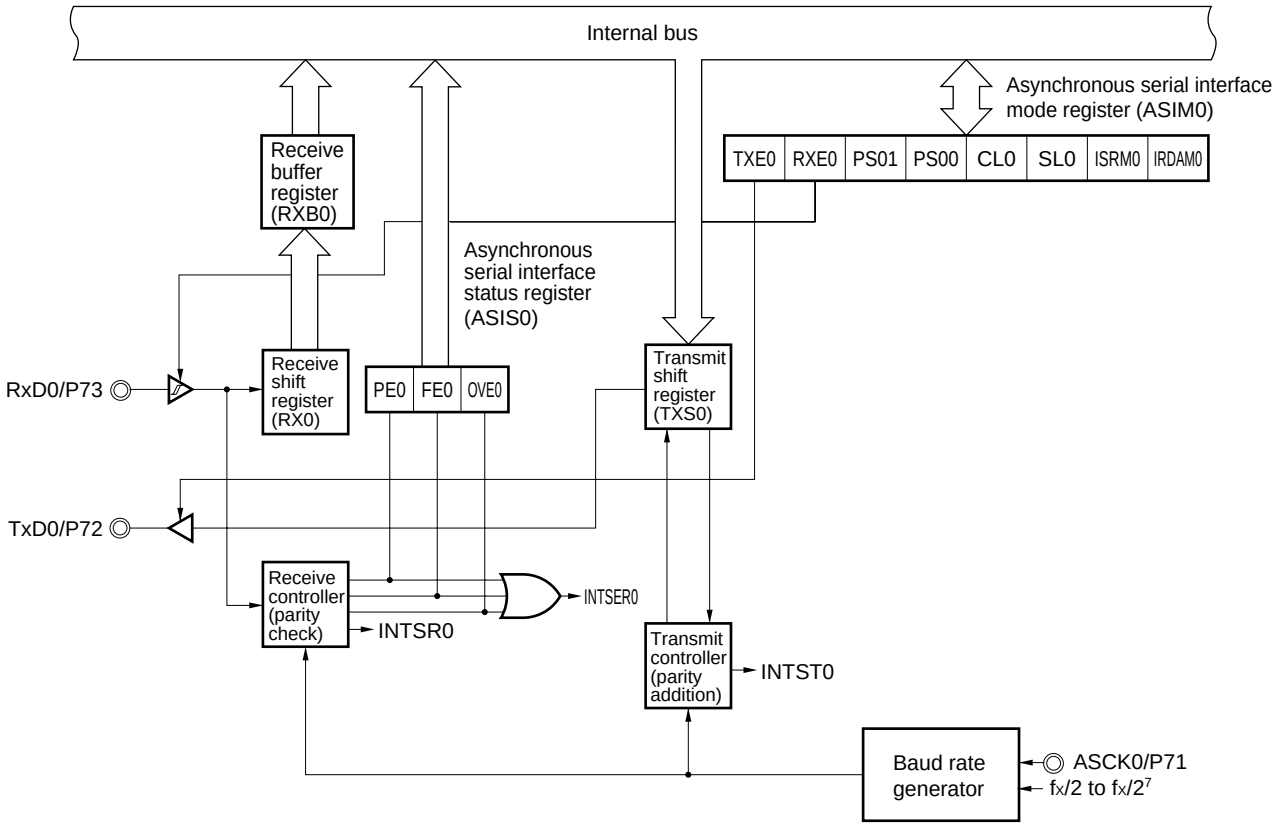
This mode enables full-duplex operation wherein one byte of data after the start bit is transmitted and received.
The on-chip baud rate generator dedicated to UART enables communications using a wide range of selectable baud rates. In addition, a baud rate can also be defined by dividing clocks input to the ASCK0 pin.
The UART baud rate generator can also be used to generate a MIDI-standard baud rate (31.25 kbps).
For details, see **13.4.2 Asynchronous serial interface (UART) mode**.

(3) Infrared data transfer mode

For details, see **13.4.3 Infrared data transfer mode**.

Figure 13-1 shows a block diagram of the serial interface (UART0) macro.

★ **Figure 13-1. Serial Interface (UART0) Block Diagram**



13.2 Serial Interface (UART0) Configuration

The serial interface (UART0) includes the following hardware.

Table 13-1. Serial Interface (UART0) Configuration

Item	Configuration
Registers	Transmit shift register (TXS0) Receive shift register (RX0) Receive buffer register (RXB0)
Control registers	Asynchronous serial interface mode register (ASIM0) Asynchronous serial interface status register (ASIS0) Baud rate generator control register (BRGC0)

(1) Transmit shift register (TXS0)

This is the register for setting transmit data. Data written to TXS0 is transmitted as serial data.

When the data length is set as 7 bits, bits 0 to 6 of the data written to TXS0 are transferred as transmit data.

Writing data to TXS0 starts the transmit operation.

TXS0 can be written by an 8-bit memory manipulation instruction. It cannot be read.

$\overline{\text{RESET}}$ input sets TXS0 to FFH.

Caution Do not write to TXS0 during a transmit operation.
The same address is assigned to TXS0 and the receive buffer register (RXB0). A read operation reads values from RXB0.

(2) Receive shift register (RX0)

This register converts serial data input via the RxD0 pin to parallel data. When one byte of data is received at this register, the receive data is transferred to the receive buffer register (RXB0).

RX0 cannot be manipulated directly by a program.

(3) Receive buffer register (RXB0)

This register is used to hold receive data. When one byte of data is received, one byte of new receive data is transferred from the receive shift register (RX0).

When the data length is set as 7 bits, receive data is sent to bits 0 to 6 of RXB0. In this case, the MSB of RXB0 always becomes 0.

RXB0 can be read by an 8-bit memory manipulation instruction. It cannot be written to.

$\overline{\text{RESET}}$ input sets RXB0 to FFH.

Caution The same address is assigned to RXB0 and the transmit shift register (TXS0). During a write operation, values are written to TXS0.

(4) Transmission control circuit

The transmission control circuit controls transmit operations, such as adding a start bit, parity bit, and stop bit to data that is written to the transmit shift register (TXS0), based on the values set to the asynchronous serial interface mode register (ASIM0).

(5) Reception control circuit

The reception control circuit controls receive operations based on the values set to the asynchronous serial interface mode register (ASIM0). During a receive operation, it performs error checking, such as for parity errors, and sets various values to the asynchronous serial interface status register (ASIS0) according to the type of error that is detected.

13.3 Registers to Control Serial Interface (UART0)

The serial interface (UART0) uses the following three types of registers for control functions.

- Asynchronous serial interface mode register (ASIM0)
- Asynchronous serial interface status register (ASIS0)
- Baud rate generator control register (BRGC0)

(1) Asynchronous serial interface mode register (ASIM0)

This is an 8-bit register that controls serial interface (UART0)'s serial transfer operations.

ASIM0 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIM0 to 00H.

Figure 13-2 shows the format of ASIM0.

Caution In UART mode, set the port mode register (PMXX) as follows. In each case, set the output latch to 0.

- **During receive operation**
  Set P73 (RxD0) to input mode (PM73 = 1)
- **During transmit operation**
  Set P72 (TxD0) to output mode (PM72 = 0)
- **During transmit/receive operation**
  Set P73 (RxD0) to input mode, and P72 to output mode

Figure 13-2. Format of Asynchronous Serial Interface Mode Register (ASIM0)

Address: FFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	IRDAM0

TXE0	RXE0	Operation mode	RxD0/P73 pin function	TxD0/P72 pin function
0	0	Operation stop	Port function (P73)	Port function (P72)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P73)	Serial function (TxD0)
1	1	UART mode (transmit and receive)	Serial function (RxD0)	

PS01	PS00	Parity bit specification
0	0	No parity
0	1	Zero parity always added during transmission No parity detection during reception (parity errors do not occur)
1	0	Odd parity
1	1	Even parity

CL0	Character length specification
0	7 bits
1	8 bits

SL0	Stop bit length specification for transmit data
0	1 bit
1	2 bits

ISRM0	Receive completion interrupt control when error occurs
0	Receive completion interrupt request is issued when an error occurs
1	Receive completion interrupt request is not issued when an error occurs

IRDAM0	Operation specified for infrared data transfer mode**Note 1**
0	UART (transmit/receive) mode
1	Infrared data transfer (transmit/receive) mode**Note 2**

Notes 1. The UART/infrared data transfer mode specification is controlled by TXE0 and RXE0.

2. When using infrared data transfer mode, be sure to set “10H” to the baud rate generator control register (BRGC0).

Caution Do not switch the operation mode until the current serial transmit/receive operation has stopped.

(2) Asynchronous serial interface status register (ASIS0)

When a receive error occurs during UART mode, this register indicates the type of error.

ASIS0 can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIS0 to 00H.

Figure 13-3. Format of Asynchronous Serial Interface Status Register (ASIS0)

Address: FFA1H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	Parity error flag
0	No parity error
0	Parity error (Incorrect parity bit detected)

FE0	Framing error flag
0	No framing error
1	Framing error[Note 1] (Stop bit not detected)

OVE0	Overrun error flag
0	No overrun error
1	Overrun error[Note 2] (Next receive operation was completed before data was read from receive buffer register)

Notes 1. Even if a stop bit length is set to two bits by setting bit 2 (SL0) in the asynchronous serial interface mode register (ASIM0), stop bit detection during a receive operation only applies to a stop bit length of 1 bit.

2. Be sure to read the contents of the receive buffer register (RXB0) when an overrun error has occurred. Until the contents of RXB0 are read, further overrun errors will occur when receiving data.

(3) Baud rate generator control register (BRGC0)

This register sets the serial clock for serial interface.

BRGC0 can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets BRGC0 to 00H.

Figure 13-4 shows the format of BRGC0.

Figure 13-4. Format of Baud Rate Generator Control Register (BRGC0)

Address: FFA2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	0	TPS02	TPS01	TPS00	MDL03	MDL02	MDL01	MDL00

(f_X = 8.38 MHz)

TPS02	TPS01	TPS00	Source clock selection for 5-bit counter	n
0	0	0	P71/ASCK0	0
0	0	1	$f_X/2$	1
0	1	0	$f_X/2^2$	2
0	1	1	$f_X/2^3$	3
1	0	0	$f_X/2^4$	4
1	0	1	$f_X/2^5$	5
1	1	0	$f_X/2^6$	6
1	1	1	$f_X/2^7$	7

MDL03	MDL02	MDL01	MDL00	Input clock selection for baud rate generator	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	Setting prohibit	—

Cautions 1. Writing to BRGC0 during a communication operation may cause abnormal output from the baud rate generator and disable further communication operations. Therefore, do not write to BRGC0 during a communication operation.

2. Set 10H to BRGC0 when using in infrared data transfer mode.

Remarks 1. f_{SCK}: Source clock for 5-bit counter

2. n: Value set via TPS00 to TPS02 ($0 \leq n \leq 7$)

3. k: Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

13.4 Serial Interface (UART0) Operations

This section explains the three modes of the serial interface (UART0).

13.4.1 Operation stop mode

Because serial transfer is not performed during this mode, the power consumption can be reduced. In addition, pins can be used as ordinary ports.

(1) Register settings

Operation stop mode are set by the asynchronous serial interface mode register (ASIM0).

ASIM0 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIM0 to 00H.

Address: FFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	IRDAM0

TXE0	RXE0	Operation mode	RxD0/P73 pin function	TxD0/P72 pin function
0	0	Operation stop	Port function (P73)	Port function (P72)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P73)	Serial function (TxD0)
1	1	UART mode (transmit and receive)	Serial function (RxD0)	

Caution Do not switch the operation mode until the current serial transmit/receive operation has stopped.

13.4.2 Asynchronous serial interface (UART) mode

This mode enables full-duplex operation wherein one byte of data after the start bit is transmitted or received.

The on-chip baud rate generator dedicated to UART enables communications using a wide range of selectable baud rates.

The UART baud rate generator can also be used to generate a MIDI-standard baud rate (31.25 kbps).

(1) Register settings

UART mode is set by the asynchronous serial interface mode register (ASIM0), asynchronous serial interface status register (ASIS0), and the baud rate generator control register (BRGC0).

(a) Asynchronous serial interface mode register (ASIM0)

ASIM0 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIM0 to 00H.

Caution In UART mode, set the port mode register (PMXX) as follows. In each case, the output latch to 0.

- **During receive operation**
  Set P73 (RxD0) to input mode (PM73 = 1)
- **During transmit operation**
  Set P72 (TxD0) to output mode (PM72 = 0)
- **During transmit/receive operation**
  Set P73 (RxD0) to input mode, and P72 to output mode

Address: FFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ASIM0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	IRDAM0

TXE0	RXE0	Operation mode	RxD0/P73 pin function	TxD0/P72 pin function
0	0	Operation stop	Port function (P73)	Port function (P72)
0	1	UART mode (receive only)	Serial function (RxD0)	
1	0	UART mode (transmit only)	Port function (P73)	Serial function (TxD0)
1	1	UART mode (transmit and receive)	Serial function (RxD0)	

PS01	PS00	Parity bit specification
0	0	No parity
0	1	Zero parity always added during transmission No parity detection during reception (parity errors do not occur)
1	0	Odd parity
1	1	Even parity

CL0	Character length specification
0	7 bits
0	8 bits

SL0	Stop bit length specification for transmit data
0	1 bit
1	2 bits

ISRM0	Receive completion interrupt control when error occurs
0	Receive completion interrupt request is issued when an error occurs
1	Receive completion interrupt request is not issued when an error occurs

IRDAM0	Operation specified for infrared data transfer mode[Note 1]
0	UART (transmit/receive) mode
1	Infrared data transfer (transmit/receive) mode[Note 2]

Notes 1. The UART/infrared data transfer mode specification is controlled by TXE0 and RXE0.

2. When using infrared data transfer mode, be sure to set the baud rate generator control register (BRGC0) to 10H.

Caution Do not switch the operation mode until the current serial transmit/receive operation has stopped.

(b) Asynchronous serial interface status register (ASIS0)

ASIS0 can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIS0 to 00H.

Address: FFA1H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	Parity error flag
0	No parity error
1	Parity error (Incorrect parity bit detected)

FE0	Framing error flag
0	No framing error
1	Framing error**Note 1** (Stop bit not detected)

OVE0	Overrun error flag
0	No overrun error
1	Overrun error**Note 2** (Next receive operation was completed before data was read from receive buffer register)

Notes 1. Even if a stop bit length is set to two bits by setting bit 2 (SL0) in the asynchronous serial interface mode register (ASIM0), stop bit detection during a receive operation only applies to a stop bit length of 1 bit.

2. Be sure to read the contents of the receive buffer register (RXB0) when an overrun error has occurred.

Until the contents of RXB0 are read, further overrun errors will occur when receiving data.

(c) Baud rate generator control register (BRGC0)

BRGC0 can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets BRGC0 to 00H.

Address: FFA2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	0	TPS02	TPS01	TPS00	MDL03	MDL02	MDL01	MDL00

(f_X = 8.38 MHz)

TPS02	TPS01	TPS00	Source clock selection for 5-bit counter	n
0	0	0	P71/ASCK0	0
0	0	1	$f_X/2$	1
0	1	0	$f_X/2^2$	2
0	1	1	$f_X/2^3$	3
1	0	0	$f_X/2^4$	4
1	0	1	$f_X/2^5$	5
1	1	0	$f_X/2^6$	6
1	1	1	$f_X/2^7$	7

MDL03	MDL02	MDL01	MDL00	Input clock selection for baud rate generator	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	Setting prohibit	—

Cautions 1. Writing to BRGC0 during a communication operation may cause abnormal output from the baud rate generator and disable further communication operations. Therefore, do not write to BRGC0 during a communication operation.

2. Set 10H to BRGC0 when using infrared data transfer mode.

Remarks 1. f_{SCK}: Source clock for 5-bit counter

2. n: Value set via TPS00 to TPS02 ($0 \leq n \leq 7$)

3. k: Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

The transmit/receive clock that is used to generate the baud rate is obtained by dividing the main system clock.

- Transmit/receive clock generation for baud rate by using main system clock
  The main system clock is divided to generate the transmit/receive clock. The baud rate generated from the main system clock is determined according to the following formula.

$$[\text{Baud rate}] = \frac{f_X}{2^{n+1}(k + 16)} \quad [\text{Hz}]$$

  f_X: Main system clock oscillation frequency
  n: Value set via TPS00 to TPS02 ($0 \leq n \leq 7$)
  For details, see Table 13-2.
  k: Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

Table 13-2 shows the relationship between the 5-bit counter's source clock assigned to bits 4 to 6 (TPS00 to TPS02) of BRGC0 and the "n" value in the above formula.

Table 13-2. Relationship between 5-Bit Counter's Source Clock and "n" Value

TPS02	TPS01	TPS00	5-Bit Counter's Source Clock Selected	n
0	0	0	P71/ASCK0	0
0	0	1	$f_X/2$	1
0	1	0	$f_X/2^2$	2
0	1	1	$f_X/2^3$	3
1	0	0	$f_X/2^4$	4
1	0	1	$f_X/2^5$	5
1	1	0	$f_X/2^6$	6
1	1	1	$f_X/2^7$	7

Remark f_X: Main system clock oscillation frequency

- **Error tolerance range for baud rates**

  The tolerance range for baud rates depends on the number of bits per frame and the counter's division rate [1/(16 + k)].

  Table 13-3 describes the relationship between the main system clock and the baud rate and Figure 13-5 shows an example of a baud rate error tolerance range.

Table 13-3. Relationship between Main System Clock and Baud Rate (1/2)

Baud Rate (bps)	f_X = 8.386 MHz		f_X = 8.000 MHz		f_X = 7.3728 MHz		f_X = 5.000 MHz		f_X = 4.1943 MHz	
	BRGC0	ERR (%)	BRGC0	ERR (%)	BRGC0	ERR (%)	BRGC0	ERR (%)	BRGC0	ERR (%)
600	–	–	–	–	–	–	–	–	7BH	1.14
1200	7BH	1.10	7AH	0.16	78H	0	70H	1.73	6BH	1.14
2400	6BH	1.10	6AH	0.16	68H	0	60H	1.73	5BH	1.14
4800	5BH	1.10	5AH	0.16	58H	0	50H	1.73	4BH	1.14
9600	4BH	1.10	4AH	0.16	48H	0	40H	1.73	3BH	1.14
19200	3BH	1.10	3AH	0.16	38H	0	30H	1.73	2BH	1.14
31250	31H	–1.3	30H	0	2DH	1.70	24H	0	21H	–1.3
38400	2BH	1.10	2AH	0.16	28H	0	20H	1.73	1BH	1.14
76800	1BH	1.10	1AH	0.16	18H	0	10H	1.73	–	–
115200	12H	1.10	11H	2.12	10H	0	–	–	–	–
Infrared data transfer mode[Note]	131031 bps		125000 bps		115200 bps		78125 bps		65536 bps	

Note The UART/infrared data transfer mode specification is controlled by TXE0 and RXE0.

When using the infrared data transfer mode, be sure to set the baud rate generator control register (BRGC0) as follows.

- k = 0 (MDL0 to MDL3 = 0000)
- n = 1 (TPS00 to TPS02 = 001)

Remark f_X: Main system clock oscillation frequency

n: Value set via TPS00 to TPS02 ($0 \leq n \leq 7$)

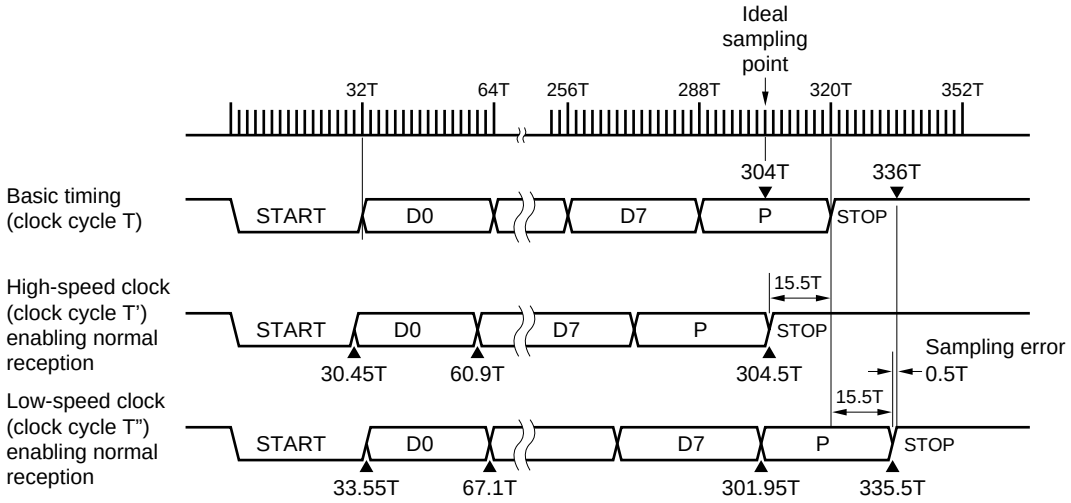
k: Value set via MDL00 to MDL03 ($0 \leq k \leq 14$)

Table 13-3. Relationship between Main System Clock and Baud Rate (2/2)

Baud Rate (bps)	f_X = 2.000 MHz		f_X = 1.000 MHz	
	BRGC0	ERR (%)	BRGC0	ERR (%)
75	9AH	0.16	8AH	0.16
100	92H	1.36	82H	1.36
150	8AH	0.16	7AH	0.16
300	7AH	0.16	6AH	0.16
600	6AH	0.16	5AH	0.16
1200	5AH	0.16	4AH	0.16
2400	4AH	0.16	3AH	0.16
4800	3AH	0.16	2AH	0.16
9600	2AH	0.16	1AH	0.16
19200	1AH	0.16	–	–
312500	10H	0	–	–

Remark f_X: Main system clock oscillation frequency

Figure 13-5. Error Tolerance (when k = 0), Including Sampling Errors



Remark T: 5-bit counter's source clock cycle

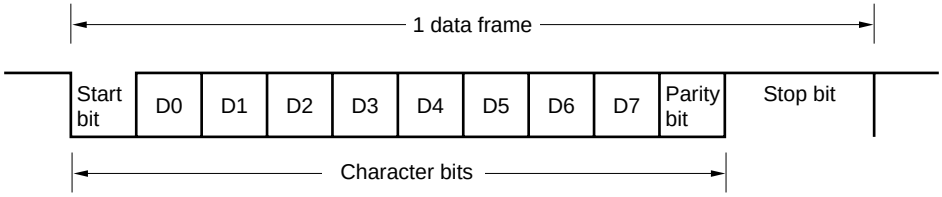
Baud rate error tolerance (when k = 0) = $\frac{\pm 15.5}{320} \times 100 = 4.8438$ (%)

(2) Communication operations

(a) Data format

Figure 13-6 shows the format of the transmit/receive data.

Figure 13-6. Format of Transmit/Receive Data in Asynchronous Serial Interface



1 data frame consists of the following bits.

- Start bit ..... 1 bit
- Character bits ... 7 bits or 8 bits
- Parity bit ..... Even parity, odd parity, zero parity, or no parity
- Stop bit(s) ..... 1 bit or 2 bits

The asynchronous serial interface mode register (ASIM0) is used to set the character bit length, parity selection, and stop bit length within each data frame.

When “7 bits” is selected as the number of character bits, only the low-order 7 bits (bits 0 to 6) are valid, so that during a transmission the highest bit (bit 7) is ignored and during reception the highest bit (bit 7) must be set to “0”.

The ASIM0 and the baud rate generator control register (BRGC0) are used to set the serial transfer rate.

If a receive error occurs, information about the receive error can be recognized by reading the asynchronous serial interface status register (ASIS0).

(b) Parity types and operations

The parity bit is used to detect bit errors in transfer data. Usually, the same type of parity bit is used by the transmitting and receiving sides. When odd parity or even parity is set, errors in the parity bit (the odd-number bit) can be detected. When zero parity or no parity is set, errors are not detected.

(i) Even parity

- During transmission
  The number of bits in transmit data that includes a parity bit is controlled so that there are an even number of bits whose value is 1. The value of the parity bit is as follows.

  If the transmit data contains an odd number of bits whose value is 1: the parity bit is “1”
  If the transmit data contains an even number of bits whose value is 1: the parity bit is “0”

- During reception
  The number of bits whose value is 1 is counted among the transfer data that include a parity bit, and a parity error occurs when the counted result is an odd number.

(ii) Odd parity

- During transmission
  The number of bits in transmit data that includes a parity bit is controlled so that there is an odd number of bits whose value is 1. The value of the parity bit is as follows.

  If the transmit data contains an odd number of bits whose value is 1: the parity bit is “0”
  If the transmit data contains an even number of bits whose value is 1: the parity bit is “1”

- During reception
  The number of bits whose value is 1 is counted among the transfer data that include a parity bit, and a parity error occurs when the counted result is an even number.

(iii) Zero parity

During transmission, the parity bit is set to “0” regardless of the transmit data.
During reception, the parity bit is not checked. Therefore, no parity errors will occur regardless of whether the parity bit is a “0” or a “1”.

(iv) No parity

No parity bit is added to the transmit data.
During reception, receive data is regarded as having no parity bit. Since there is no parity bit, no parity errors will occur.

(c) Transmission

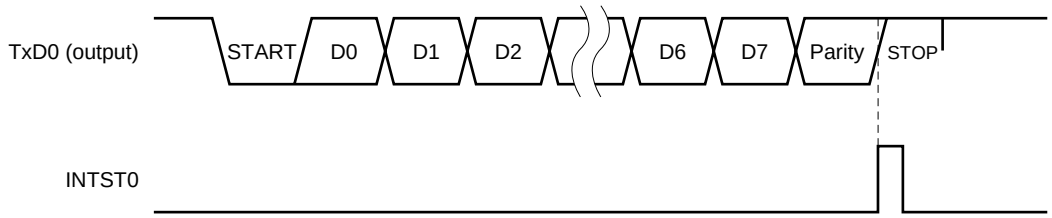
The transmit operation is started when transmit data is written to the transmit shift register (TXS0). A start bit, parity bit, and stop bit(s) are automatically added to the data.

Starting the transmit operation shifts out the data in TXS0, thereby emptying TXS0, after which a transmit completion interrupt request (INTST0) is issued.

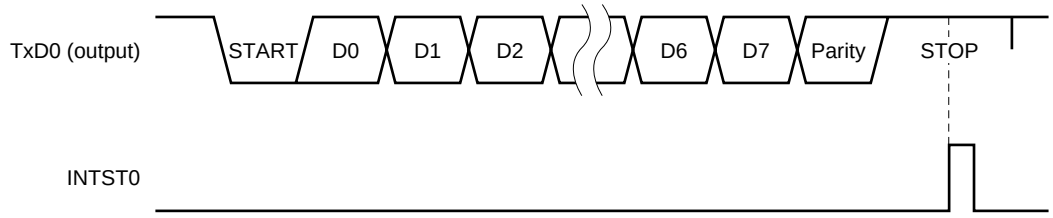
The timing of the transmit completion interrupt request is shown in Figure 13-7.

Figure 13-7. Timing of Asynchronous Serial Interface Transmit Completion Interrupt Request

(i) Stop bit length: 1 bit



(ii) Stop bit length: 2 bits



Caution Do not rewrite to the asynchronous serial interface mode register (ASIM0) during a transmit operation. Rewriting ASIM0 register during a transmit operation may disable further transmit operations (in such cases, enter a $\overline{\text{RESET}}$ to restore normal operation). Whether or not a transmit operation is in progress can be determined via software using the transmit completion interrupt request (INTST0) or the interrupt request flag (STIF0) that is set by INTST0.

(d) Reception

The receive operation is enabled when “1” is set to bit 6 (RXE0) of the asynchronous serial interface mode register (ASIM0), and input via the RxD0 pin is sampled.

The serial clock specified by ASIM0 is used to sample the RxD0 pin.

When the RxD0 pin goes low, the 5-bit counter of the baud rate generator begins counting and the start timing signal for data sampling is output when half of the specified baud rate time has elapsed. If sampling the RxD0 pin input with this start timing signal yields a low-level result, a start bit is recognized, after which the 5-bit counter is initialized and starts counting and data sampling begins. After the start bit is recognized, the character data, parity bit, and one-bit stop bit are detected, at which point reception of one data frame is completed.

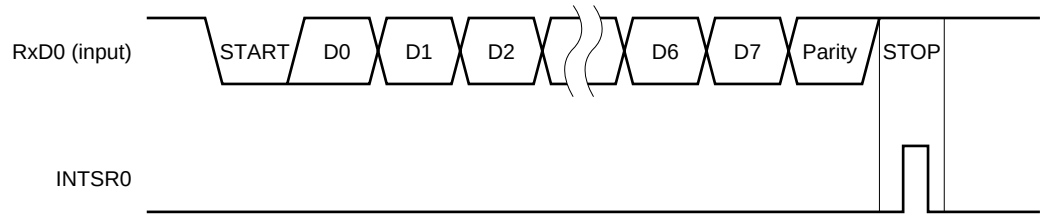
Once reception of one data frame is completed, the receive data in the shift register is transferred to the receive buffer register (RXB0) and a receive completion interrupt request (INTSR0) occurs.

Even if an error has occurred, the receive data in which the error occurred is still transferred to RXB0. When ASIM0 bit 1 (ISRM0) is cleared (0) upon occurrence of an error, INTSR0 occurs (see Figure 13-9). When ISRM0 bit is set (1), INTSR0 does not occur.

If the RXE0 bit is reset (to “0”) during a receive operation, the receive operation is stopped immediately. At this time, the contents of RXB0 and ASIS0 do not change, nor does INTSR0 or INTSER0 occur.

Figure 13-8 shows the timing of the asynchronous serial interface receive completion interrupt request.

Figure 13-8. Timing of Asynchronous Serial Interface Receive Completion Interrupt Request



Caution Be sure to read the contents of the receive buffer register (RXB0) even when a receive error has occurred. Overrun errors will occur during the next data receive operations and the receive error status will remain until the contents of RXB0 are read.

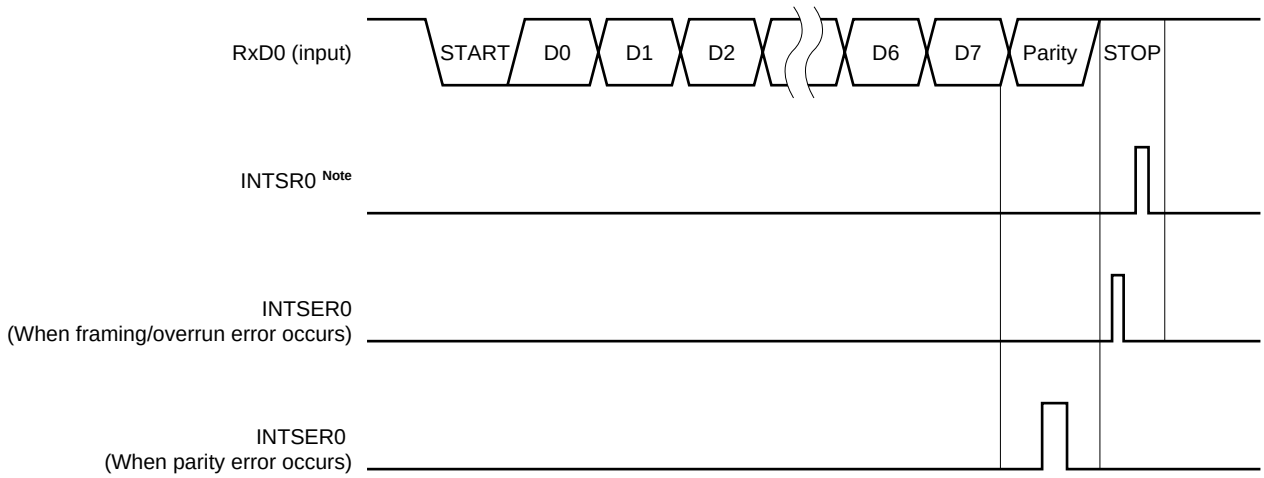
(e) Receive errors

Three types of errors can occur during a receive operation: parity error, framing error, or overrun error. If, as the result of data reception, an error flag is set to the asynchronous serial interface status register (ASIS0), a receive error interrupt request (INTSER0) will occur. Receive error interrupts requests are generated before receiving completion interrupts request (INTSR0). Table 13-4 lists the causes behind receive errors.
As part of receive error interrupt request (INTSER0) servicing, the contents of ASIS0 can be read to determine which type of error occurred during the receive operation (see Table 13-4 and Figure 13-9).
The contents of ASIS0 are reset (to "0") when the receive buffer register (RXB0) is read or when the next data is received (if the next data contains an error, its error flag will be set).

Table 13-4. Causes of Receive Errors

Receive Error	Cause	ASIS0 Value
Parity error	Parity specified during transmission does not match parity of receive data	04H
Framing error	Stop bit was not detected	02H
Overrun error	Reception of the next data was completed before data was read from the receive buffer register	01H

Figure 13-9. Receive Error Timing



Note If a receive error occurs when ISRM0 bit has been set (1), INTSR0 does not occur.

Cautions 1. The contents of ASIS0 are reset (to "0") when the receive buffer register (RXB0) is read or when the next data is received. To obtain information about the error, be sure to read the contents of ASIS0 before reading RXB0.

2. Be sure to read the contents of the receive buffer register (RXB0) even when a receive error has occurred. Overrun errors will occur during the next data receive operations and the receive error status will remain until the contents of RXB0 are read.

13.4.3 Infrared data transfer mode

In infrared data transfer mode, the following data format pulse output and pulse receiving are enabled. The relationship between the main system clock and baud rate is shown in Table 13-3.

(1) Data format

Figure 13-10 compares the data format used in UART mode with that used in infrared data transfer mode.

The IR (infrared) frame corresponds to the bit string of the UART frame, which consists of pulses – a start bit, eight data bits, and a stop bit.

The length of the electrical pulses that are used to transmit and receive in an IR frame is 3/16 the length of the cycle time for one bit (i.e., the “bit time”). This pulse (whose width is 3/16 the length of one bit time) rises from the middle of the bit time (see the figure below).

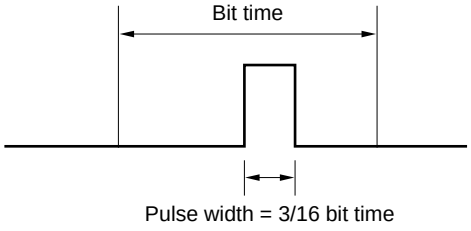
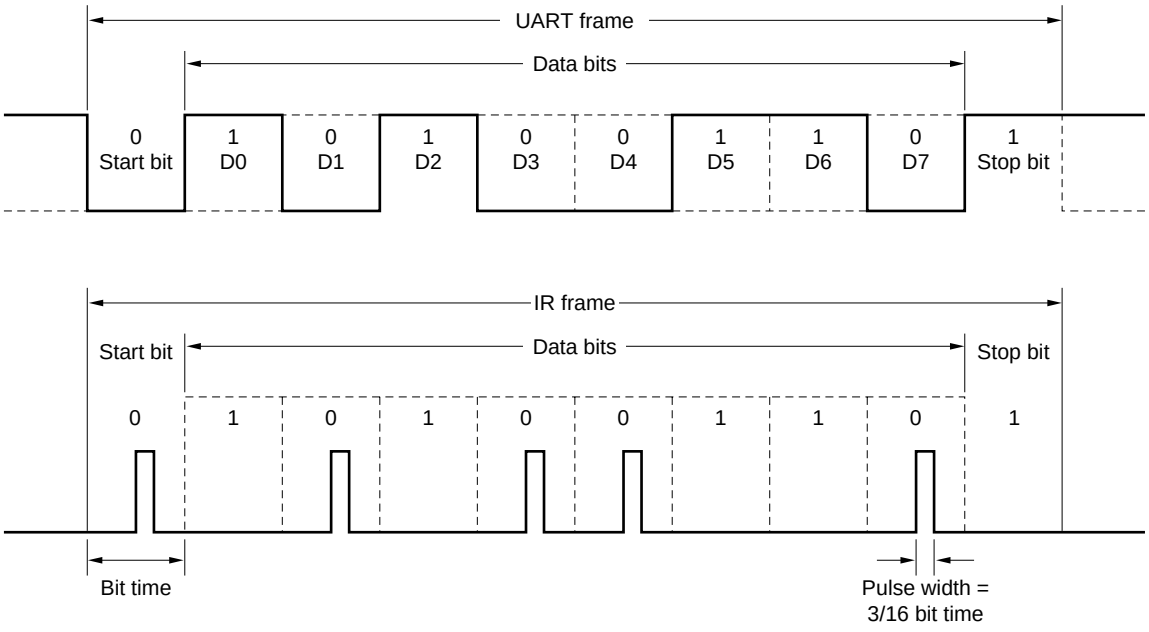


Figure 13-10. Data Format Comparison between Infrared Data Transfer Mode and UART Mode



(2) Bit rate and pulse width

Table 13-5 lists bit rates, bit rate error tolerances, and pulse width values.

Table 13-5. Bit Rate and Pulse Width Values

Bit Rate (kbits/s)	Bit Rate Error Tolerance (% of bit rate)	Pulse Width Minimum Value (μs) Note 2	3/16 Pulse Width <Nominal Value> (μs)	Maximum Pulse Width (μs)
115.2 Note 1	+/– 0.87	1.41	1.63	2.71

Notes 1. At the operation time with f_X = 7.3728 MHz

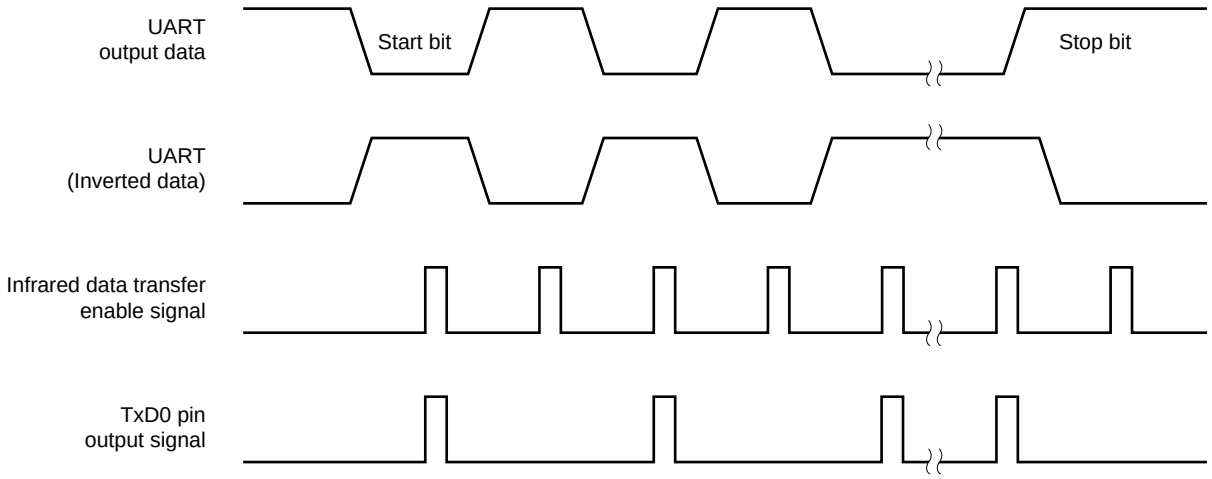
2. When a digital noise elimination circuit is used in a microcontroller operating at 1.41 MHz or above.

Caution When using the baud rate generator control register (BRGC0) in infrared data transfer mode, set 10H to it.

Remark f_X: main system clock oscillation frequency

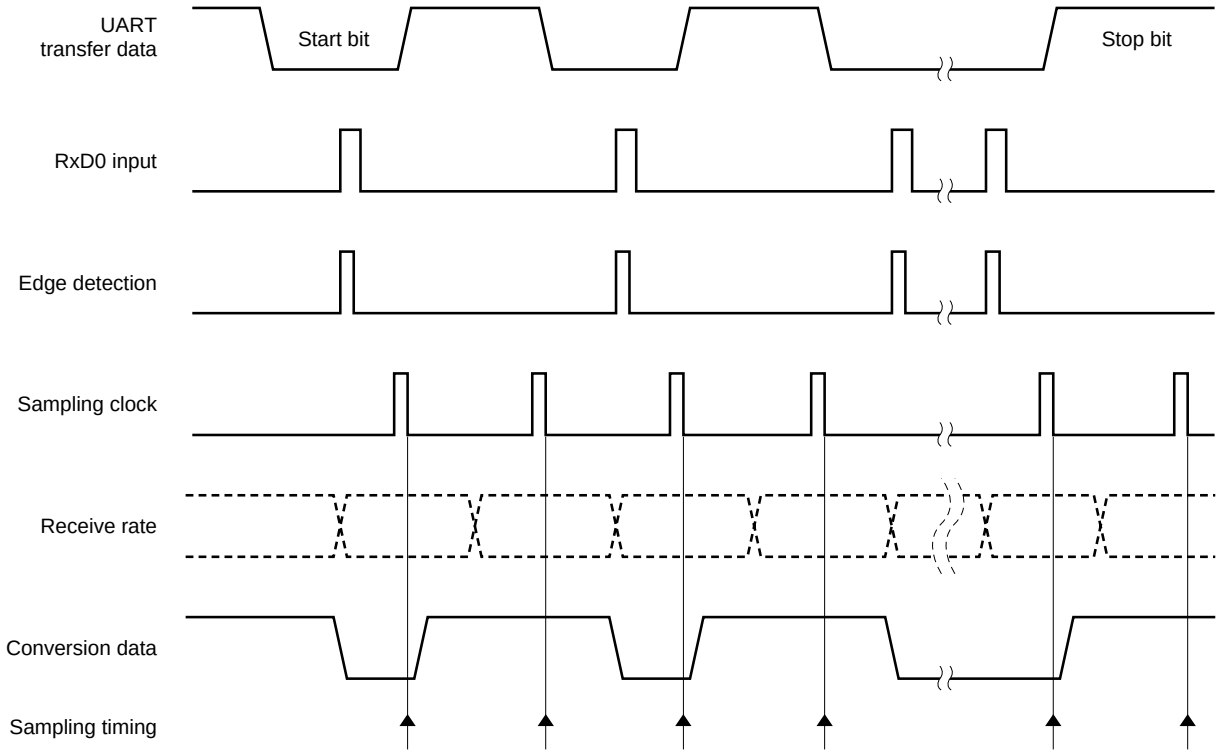
(3) Input data and internal signals

- **Transmit operation timing**



- **Receive operation timing**

  Data reception is delayed for one-half of the specified baud rate.



★

CHAPTER 14 SERIAL INTERFACE (SIO1)

14.1 Serial Interface (SIO1) Functions

The serial interface (SIO1) employs the following three modes.

- Operation stop mode
- 3-wire serial I/O mode
- 3-wire serial I/O mode with automatic transmit/receive function

(1) Operation stop mode

This mode is used when serial transfer is not carried out.

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

This mode is used for 8-bit data transfer using three lines, each for serial clock ($\overline{\text{SCK1}}$), serial output (SO1) and serial input (SI1).

The 3-wire serial I/O mode enables simultaneous transmission/reception and so decreases the data transfer processing time.

Since the start bit of the 8-bit data that undergoes serial transfer is switchable between MSB and LSB, connection is enabled with either start bit device.

The 3-wire serial I/O mode is valid for connection of peripheral I/O units and display controllers which incorporate a conventional synchronous serial interface.

(3) 3-wire serial I/O mode with automatic transmit/receive function

This mode has an automatic transmit/receive function in addition to the functions in **(2)** above.

The automatic transmit/receive function is used to transmit/receive data with a maximum of 32 bytes. This function enables the hardware to transmit/receive data independently of the CPU, to/from the OSD (On Screen Display) device and a device with a built-in display controller/driver, thus lightning the software load can be alleviated.

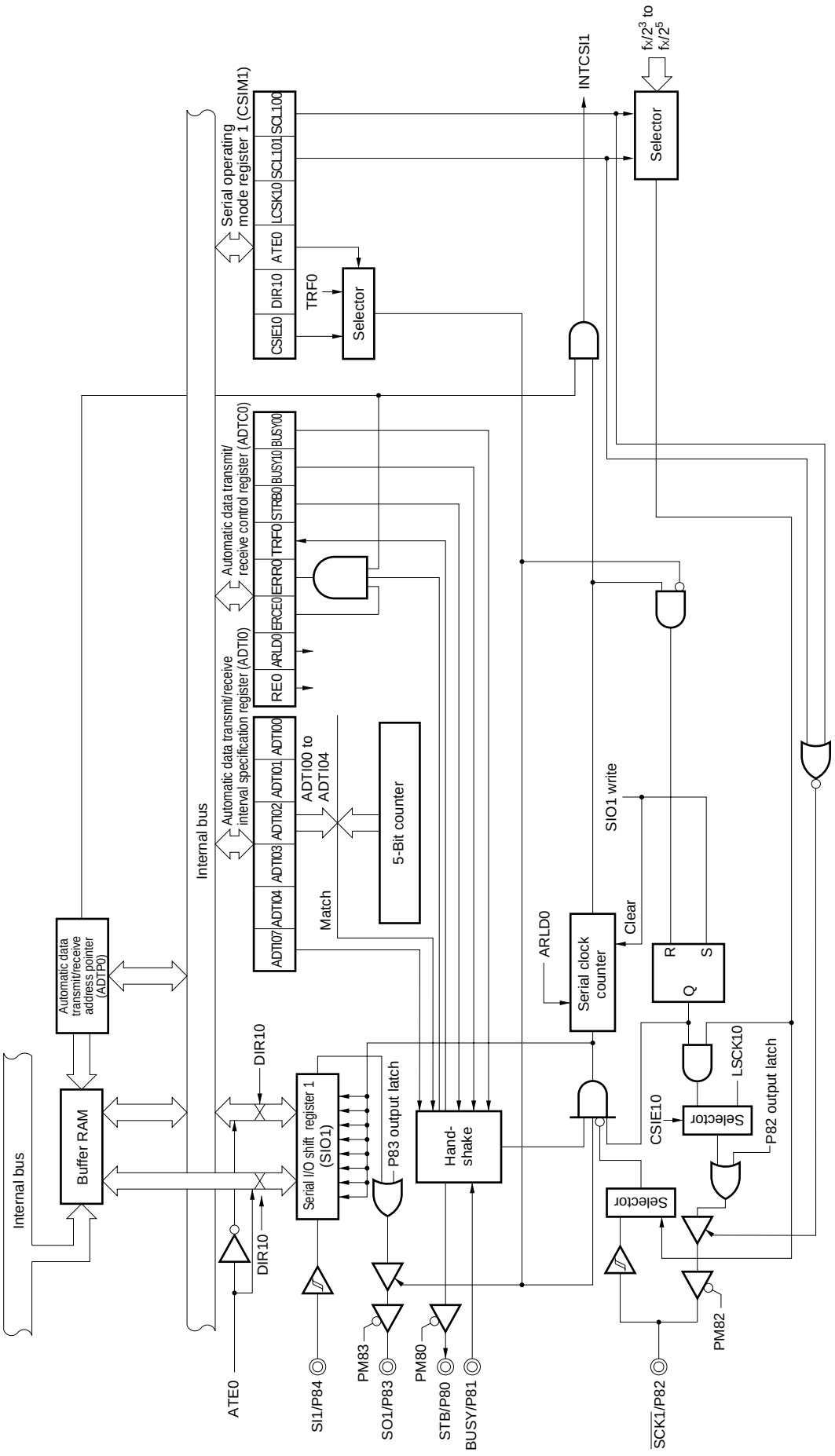
14.2 Serial Interface (SIO1) Configuration

Serial interface (SIO1) consists of the following hardware.

Table 14-1. Configuration of Serial Interface (SIO1)

Item	Configuration
Register	Serial I/O shift register 1 (SIO1) Automatic data transmit/receive address pointer (ADTP0)
Control register	Serial operating mode register 1 (CSIM1) Automatic data transmit/receive control register (ADTC0) Automatic data transmit/receive interval specification register (ADTI0)

Figure 14-1. Block Diagram of Serial Interface (SIO1)



(1) Serial I/O shift register 1 (SIO1)

This is an 8-bit register used to carry out parallel/serial conversion and to carry out serial transmission/reception (shift operation) in synchronization with the serial clock.

SIO1 is set with an 8-bit memory manipulation instruction.

When the value in bit 7 (CSIE10) of serial operating mode register 1 (CSIM1) is 1, writing data to SIO1 starts the 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 (ADTP0)

This register stores the value of (transmit data byte –1) while the automatic transmit/receive function is activated. As data is transferred/received, it is automatically decremented.

ADTP0 is set with an 8-bit memory manipulation instruction. The high-order 3 bits must be set to 0.

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

Caution Do not write data to ADTP0 while the automatic transmit/receive function is activated.

(3) Serial clock counter

This counter counts the serial clocks to be output and input during transmission/reception to check whether 8-bit data has been transmitted/received.

14.3 Serial Interface (SIO1) Control Registers

The following three types of registers are used to control serial interface (SIO1).

- Serial operation mode register 1 (CSIM1)
- Automatic data transmit/receive control register (ADTC0)
- Automatic data transmit/receive interval specification register (ADTI0)

(1) Serial operation mode register 1 (CSIM1)

This register sets the serial interface (SIO1) serial clock, operation 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.

Caution In serial I/O mode, set the port mode register (PMXX) as follows. In each case, set the output latch to 0 respectively.

- **During serial clock output (master transmission or master reception)**
  Set P82 (SCK1) to output mode (PM82 = 0).
- **During serial clock input (slave transmission or slave reception)**
  Set P82 to input mode (PM82 = 1).
- **During transmission/transmission and reception mode**
  Set P83 (SO1) to output mode (PM83 = 0).
- **During reception mode**
  Set P84 (SI1) to input mode (PM84 = 1).

Figure 14-2. Format of Serial Operation Mode Register 1 (CSIM1)

Address: FF68H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM1	CSIE10	DIR10	ATE0	LCSK10	0	0	SCL101	SCL100

CSIE10	Enables/disables operation of serial interface (SIO1)		
	Shift register operation	Serial counter	Port[Note 1]
0	Stops operation	Cleared	Port function
1	Enables operation	Enables count operation	Serial function + port function

DIR10	Specifies first bit of serial transfer data
0	MSB
1	LSB

ATE0	Selects operating mode of serial interface (SIO1)
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

LCSK10	Chip enable control of $\overline{SCK1}$ pin
0	$\overline{SCK1}$ is used as a port (P82) when CSIE10 = 0. $\overline{SCK1}$ is used for clock output when CSIE10 = 1.
1	$\overline{SCK1}$ is fixed to high level output when CSIE10 = 0. $\overline{SCK1}$ is used for clock output when CSIE10 = 1.

SCL101	SCL100	Selects serial clock of serial interface (SIO1)
0	0	External clock input to $\overline{SCK1}$ pin[Note 2]
0	1	$f_X/2^3$ (1.05 MHz)
1	0	$f_X/2^4$ (524 kHz)
1	1	$f_X/2^5$ (262 kHz)

Notes 1. When CSIE10 = 0 (SIO1 operation stop status), P84/SI1, P83/SO1, P82/$\overline{SCK1}$, P81/BUSY, and P80/STB pins can be used as port pins.

2. When external clock input is selected by clearing SCL101 and SCL100 to 0, 0, clear bits 2 (STRB0) and 1 (BUSY10) of the automatic data transmit/receive control register (ADTC0) to 0, 0.

★ **Caution Be sure to set bits 2 and 3 to 0.**

Remarks 1. f_X: Main system clock oscillation frequency

2. ( ): f_X = 8.38 MHz

(2) Automatic data transmit/receive control register (ADTC0)

This register sets the automatic receive enable/disable, operating mode, strobe output enable/disable, busy input enable/disable and displays the automatic transmit/receive execution.

ADTC0 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 14-3. Format of Automatic Data Transmit/Receive Control Register (ADTC0)

Address: FF69H After reset: 00H R/W[Note 1]

Symbol	7	6	5	4	3	2	1	0
ADTC0	RE0	ARLD0	ERCE0	ERR0	TRF0	STRB0	BUSY10	BUSY00

R/W

BUSY10	BUSY00	Busy input control
0	×	Not using busy input[Note 2]
1	0	Busy input enable (active high)
1	1	Busy input enable (active low)

R/W

STRB0	Strobe output control
0	Strobe output disable[Note 3]
1	Strobe output enable

R

TRF0	Status of automatic transmit/receive function[Note 4]
0	Detection of termination of automatic transmission/ reception (This bit is set to 0 upon suspension of automatic transmission/reception or when ARLD0 = 0.)
1	During automatic transmission/reception (This bit is set to 1 when data is written to SIO1.)

R

ERR0	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

ERCE0	Error check control of automatic transmit/ receive function
0	Error check disable
1	Error check enable (only when BUSY10 = 1)

R/W

ARLD0	Operating mode selection of automatic transmit/ receive function
0	Single operating mode
1	Repetitive operating mode

R/W

RE0	Receive control of automatic transmit/receive function
0	Receive disable[Note 5]
1	Receive enable

Notes 1. Bits 3 and 4 (TRF0 and ERR0) are read-only bits.

2. When BUSY10 is reset to 0, P81 (CMOS I/O) is used even when bit 7 (CSIE10) of the serial operating mode register 1 (CSIM1) is set to 1.

3. When STRB0 is reset to 0, P80 (CMOS I/O) is used even when bit 7 (CSIE10) of CSIM1 is set to 1.

4. When an interrupt is acknowledged, interrupt request flag CSIIF1 is cleared. Therefore, use TRF0, instead of CSIIF1 to identify the completion of automatic transmission/reception.

5. When RE0 is reset to 0, P84 (CMOS I/O) is used even when bit 7 (CSIE10) of CSIM1 is set to 1.

Caution When an external clock input is selected with bits 0 and 1 (SCL101 and SCL100) of CSIM1 set to 0, set bits 2 and 1 (STRB0 and BUSY10) of ADTC0 to 0, 0.

Remark ×: don't care

(3) Automatic data transmit/receive interval specification register (ADTI0)

This register sets the automatic data transmit/receive function data transfer interval.

ADTI0 is set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 14-4. Format of Automatic Data Transmit/Receive Interval Specification Register (ADTI0) (1/2)

Address: FF6BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTI0	ADTI07	0	0	ADTI04	ADTI03	ADTI02	ADTI01	ADTI00

ADTI07	Data transfer interval control
0	No control of interval by ADTI0 Note 1
1	Control of interval by ADTI0 (ADTI00 to ADTI04)

ADTI04	ADTI03	ADTI02	ADTI01	ADTI00	Data transfer interval specification (f_X = 8.38 MHz, f_{SCK} = 1.05 MHz) Note 2
0	0	0	0	0	1.90 μs + 0.5/f_{SCK}
0	0	0	0	1	
0	0	0	1	0	2.86 μs + 0.5/f_{SCK}
0	0	0	1	1	3.81 μs + 0.5/f_{SCK}
0	0	1	0	0	4.76 μs + 0.5/f_{SCK}
0	0	1	0	1	5.71 μs + 0.5/f_{SCK}
0	0	1	1	0	6.67 μs + 0.5/f_{SCK}
0	0	1	1	1	7.62 μs + 0.5/f_{SCK}
0	1	0	0	0	8.57 μs + 0.5/f_{SCK}
0	1	0	0	1	9.52 μs + 0.5/f_{SCK}
0	1	0	1	0	10.5 μs + 0.5/f_{SCK}
0	1	0	1	1	11.4 μs + 0.5/f_{SCK}
0	1	1	0	0	12.4 μs + 0.5/f_{SCK}
0	1	1	0	1	13.3 μs + 0.5/f_{SCK}
0	1	1	1	0	14.3 μs + 0.5/f_{SCK}
0	1	1	1	1	15.2 μs + 0.5/f_{SCK}

Notes 1. The interval time is $2/f_{SCK}$.

2. The data transfer interval time is calculated from the following expressions (n: Value set to ADTI00 to ADTI04).

<1> n = 0

$$\text{Interval time} = \frac{2}{f_{SCK}} + \frac{0.5}{f_{SCK}}$$

<2> n = 1 to 31

$$\text{Interval time} = \frac{n+1}{f_{SCK}} + \frac{0.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI0 during operation of automatic data transmit/receive function.

2. Be sure to set bits 5 and 6 to 0.

3. When controlling the interval time of automatic transmit/receive data transfer by using ADTI0, busy control is invalid. (Refer to 14.4.3 (4) (a) Busy control option.)

Remark f_X: Main system clock oscillation frequency

f_{SCK}: Serial clock frequency

Figure 14-4. Format of Automatic Data Transmit/Receive Interval Specification Register (ADTI0) (2/2)

Address: FF6BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTI0	ADTI07	0	0	ADTI04	ADTI03	ADTI02	ADTI01	ADTI00

ADTI04	ADT0I3	ADTI02	ADTI01	ADTI00	Data transfer interval specification (f_X = 8.38 MHz, f_{SCK} = 1.05 MHz)Note
1	0	0	0	0	16.2 μs + 0.5/f_{SCK}
1	0	0	0	1	17.1 μs + 0.5/f_{SCK}
1	0	0	1	0	18.1 μs + 0.5/f_{SCK}
1	0	0	1	1	19.0 μs + 0.5/f_{SCK}
1	0	1	0	0	20.0 μs + 0.5/f_{SCK}
1	0	1	0	1	21.0 μs + 0.5/f_{SCK}
1	0	1	1	0	21.9 μs + 0.5/f_{SCK}
1	0	1	1	1	22.9 μs + 0.5/f_{SCK}
1	1	0	0	0	23.8 μs + 0.5/f_{SCK}
1	1	0	0	1	24.8 μs + 0.5/f_{SCK}
1	1	0	1	0	25.7 μs + 0.5/f_{SCK}
1	1	0	1	1	26.7 μs + 0.5/f_{SCK}
1	1	1	0	0	27.6 μs + 0.5/f_{SCK}
1	1	1	0	1	28.6 μs + 0.5/f_{SCK}
1	1	1	1	0	29.5 μs + 0.5/f_{SCK}
1	1	1	1	1	30.5 μs + 0.5/f_{SCK}

Note The data transfer interval time is calculated from the following expressions (n: Value set to ADTI00 to ADTI04).

<1> n = 0

$$\text{Interval time} = \frac{2}{f_{SCK}} + \frac{0.5}{f_{SCK}}$$

<2> n = 1 to 31

$$\text{Interval time} = \frac{n+1}{f_{SCK}} + \frac{0.5}{f_{SCK}}$$

Cautions
1. **Do not write ADTI0 during operation of automatic data transmit/receive function.**
2. **Be sure to set bits 5 and 6 to 0.**
3. **When controlling the interval time of automatic transmit/receive data transfer by using ADTI0, busy control is invalid. (Refer to 14.4.3 (4) (a) Busy control option.)**

Remark f_X: Main system clock oscillation frequency
f_{SCK}: Serial clock frequency

14.4 Serial Interface (SIO1) Operations

The following three operating modes are available to the serial interface (SIO1).

- 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. The serial I/O shift register 1 (SIO1) does not carry out shift operation either, and thus it can be used as an ordinary 8-bit register.

In the operation stop mode, the P84/SI1, P83/SO1, P82/$\overline{\text{SCK1}}$, P81/BUSY and P80/STB pins can be used as ordinary input/output ports.

(1) Register setting

The operation stop mode is set with the serial operating mode register 1 (CSIM1).

CSIM1 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM1 to 00H.

Address: FF68H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM1	CSIE10	DIR10	ATE0	LCSK10	0	0	SCL101	SCL100

CSIE10	Enables/disables operation of serial interface (SIO1)		
	Shift register operation	Serial counter	Port[Note 1]
0	Stops operation	Cleared	Port function
1	Enables operation	Enables count operation	Serial function + port function

Note When CSIE10 = 0 (SIO1 operation stop status), P84/SI1, P83/SO1, P82/$\overline{\text{SCK1}}$, P81/BUSY, and P80/STB pins can be used as port pins.

★ **Caution Be sure to set bits 2 and 3 to 0.**

14.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is valid for connection of peripheral I/O units and display controllers which incorporate a conventional synchronous serial interface.

Communication is carried out with three lines of serial clock ($\overline{SCK1}$), serial output (SO1) and serial input (SI1).

(1) Register setting

The 3-wire serial I/O mode is set with 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.

Caution In serial I/O mode, set the port mode register (PMXX) as follows. In each case, set the output latch to 0.

- **During serial clock output (master transmission or master reception)**
  Set P82 ($\overline{SCK1}$) to output mode (PM82 = 0).
- **During serial clock input (slave transmission or slave reception)**
  Set P82 to input mode (PM82 = 1).
- **During transmission/transmission and reception mode**
  Set P83 (SO1) to output mode (PM83 = 0).
- **During reception mode**
  Set P84 (SI1) to input mode (PM84 = 1).

Address: FF68H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM1	CSIE10	DIR10	ATE0	LCSK10	0	0	SCL101	SCL100

CSIE10	Enables/disables operation of serial interface (SIO1)		
	Shift register operation	Serial counter	Port[Note 1]
0	Stops operation	Cleared	Port function
1	Enables operation	Enables count operation	Serial function + port function

DIR10	Start bit
0	MSB
1	LSB

ATE0	Selects operating mode of serial interface (SIO1)
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

LCSK10	Chip enable control of $\overline{SCK1}$ pin
0	$\overline{SCK1}$ is used as port (P82) when CSIE10 = 0. $\overline{SCK1}$ is used for clock output when CSIE10 = 1.
1	$\overline{SCK1}$ is fixed to high level when CSIE10 = 0. $\overline{SCK1}$ is used for clock output when CSIE10 = 1.

SCL101	SCL100	Selects serial clock of serial interface (SIO1)
0	0	External clock input to $\overline{SCK1}$ pin**Note 2**
0	1	$f_X/2^3$ (1.05 MHz)
1	0	$f_X/2^4$ (524 kHz)
1	1	$f_X/2^5$ (262 kHz)

Notes 1. When CSIE10 = 0 (SIO1 operation stop status), P84/SI1, P83/SO1, P82/$\overline{SCK1}$, P81/BUSY, and P80/STB pins can be used as port pins.

2. When external clock input is selected by clearing SCL101 and SCL100 to 0, 0, clear bits 2 (STRB0) and 1 (BUSY10) of the automatic data transmit/receive control register (ADTC0) to 0,0.

★ **Caution Be sure to set bits 2 and 3 to 0.**

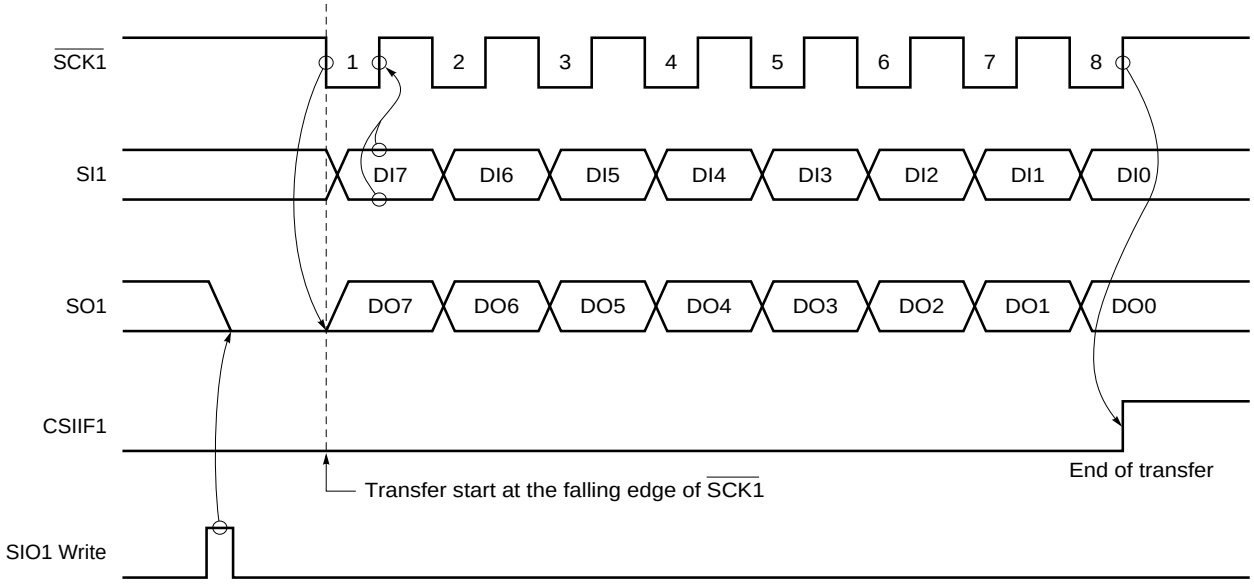
(2) Communication operation

The 3-wire serial I/O mode is used for data transmission/reception in 8-bit units. 1-bit unit data transmission/reception is carried out in synchronization with the serial clock.

Shift operation of the serial I/O shift register 1 (SIO1) is carried out at the falling edge of the serial clock ($\overline{SCK1}$). The transit 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 14-5. 3-Wire Serial I/O Mode Timings



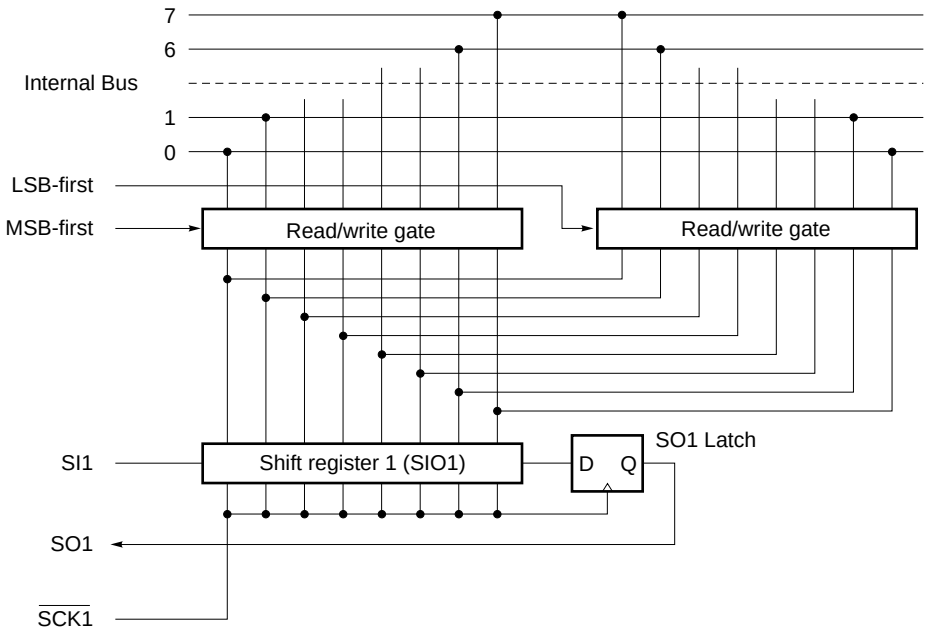
Caution SIO1 pin becomes low level by SIO1 write.

(3) MSB/LSB switching as the start bit

The 3-wire serial I/O mode has a function that enables it to select the transfer to start from either MSB or LSB. Figure 14-6 shows the configuration of the serial I/O shift register 1 (SIO1) and internal bus. As shown in the figure, MSB/LSB can be read/written in reverse form.

MSB/LSB switching as the start bit can be specified with bit 6 (DIR10) of the serial operating mode register 1 (CSIM1).

Figure 14-6. Switching Circuit in Transfer Bit Order



Start bit switching is realized by switching the bit order for data write to SIO1. The SIO1 shift order remains unchanged.

Thus, MSB/LSB start bit must be switched before writing data to the shift register.

(4) Transfer start

Serial transfer is started by setting transfer data to the serial I/O shift register 1 (SIO1) when the following two conditions are satisfied.

- Serial interface (SIO1) operation control bit (bit 7 (CSIE10) of the serial operating mode register 1 (CSIM1)) = 1
- Internal serial clock is stopped or SCK1 is at high level after 8-bit serial transfer.

Caution If CSIE10 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 with automatic transmit/receive function

This 3-wire serial I/O mode is used for transmission/reception of a maximum 32-byte data without the use of software. Once transfer is started, the data prestored in the RAM can be transmitted by the set number of bytes, and data can be received and stored in the RAM by the set number of bytes.

Handshake signals (STB and BUSY) are supported by hardware to transmit/receive data continuously. OSD (On Screen Display) LSI and peripheral LSIs 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 with the serial operation mode register 1 (CSIM1), the automatic data transmit/receive control register (ADTC0) and the automatic data transmit/receive interval specification register (ADTI0).

(a) Serial operation 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.

Caution In serial I/O mode, set the port mode register (PMXX) as follows. In each case, set the output latch to 0.

- **During serial clock output (master transmission or master reception)**
  Set P82 (SCK1) to output mode (PM82 = 0).
- **During serial clock input (slave transmission or slave reception)**
  Set P82 to input mode (PM82 = 1).
- **During transmission/transmission and reception mode**
  Set P83 (SO01) to output mode (PM83 = 0).
- **During reception mode**
  Set P84 (SI1) to input mode (PM84 = 1).

Address: FF68H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM1	CSIE10	DIR10	ATE0	LCSK10	0	0	SCL101	SCL100

CSIE10	Enables/disables operation of serial interface (SIO1)		
	Shift register operation	Serial counter	Port[Note 1]
0	Stops operation	Cleared	Port function
1	Enables operation	Enables count operation	Serial function + port function

DIR10	Start bit
0	MSB
1	LSB

ATE0	Selects operation mode of serial interface (SIO1)
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

LCSK10	Chip enable control of $\overline{SCK1}$ pin
0	$\overline{SCK1}$ is used as port (P82) when CSIE10 = 0. $\overline{SCK1}$ is used for clock output when CSIE10 = 1.
1	$\overline{SCK1}$ is fixed to high level output when CSIE10 = 0. $\overline{SCK1}$ is used for clock output when CSIE10 = 1.

SCL101	SCL100	Selects serial clock of serial interface (SIO1)
0	0	External clock input to $\overline{SCK1}$ pin[Note 2]
0	1	$f_X/2^3$ (1.05 MHz)
1	0	$f_X/2^4$ (524 kHz)
1	1	$f_X/2^5$ (262 kHz)

Notes 1. When CSIE10 = 0 (SIO1 operation stop status), P84/SI1, P83/SO1, $\overline{P82/SCK1}$, P81/BUSY, and P80/STB pins can be used as port pins.

2. When external clock input is selected by clearing SCL101 and SCL100 to 0, 0, clear bits 2 (STRB0) and 1 (BUSY10) of the automatic data transmit/receive control register (ADTC0) to 0, 0.

★ **Caution Be sure to set bits 2 and 3 to 0.**

(b) Automatic data transmit/receive control register (ADTC0)

ADTC0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADTC0 to 00H.

Address: FF69H After reset: 00H R/W[Note 1]

Symbol	7	6	5	4	3	2	1	0
ADTC0	RE0	ARLD0	ERCE0	ERR0	TRF0	STRB0	BUSY10	BUSY00

R/W

BUSY10	BUSY00	Busy input control
0	×	Not using busy input[Note 2]
1	0	Busy input enable (active high)
1	1	Busy input enable (active low)

R/W

STRB0	Strobe output control
0	Strobe output disable[Note 3]
1	Strobe output enable

R

TRF0	Status of automatic transmit/receive function[Note 4]
0	Detection of termination of automatic transmission/ reception (This bit is set to 0 upon suspension of automatic transmission/reception or when ARLD0 = 0.)
1	During automatic transmission/reception (This bit is set to 1 when data is written to SIO1.)

R

ERR0	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

ERCE0	Error check control of automatic transmit/receive function
0	Error check disable
1	Error check enable (only when BUSY10 = 1)

R/W

ARLD0	Operation mode selection of automatic transmit/ receive function
0	Single operating mode
1	Repetitive operating mode

R/W

RE0	Receive control of automatic transmit/receive function
0	Receive disable[Note 5]
1	Receive enable

Notes 1. Bits 3 and 4 (TRF0 and ERR0) are read-only bits.

2. When BUSY10 is reset to 0, P81 (CMOS I/O) is used even when bit 7 (CSIE10) of the serial operating mode register 1 (CSIM1) is set to 1.

3. When STRB0 is reset to 0, P80 (CMOS I/O) is used even when bit 7 (CSIE10) of CSIM1 is set to 1.

4. When an interrupt is acknowledged, interrupt request flag CSIIF1 is cleared. Therefore, use TRF0, instead of CSIIF1, to identify the completion of automatic transmission/reception.

5. When RE0 is reset to 0, P84 (CMOS I/O) is used even when bit 7 (CSIE10) of CSIM1 is set to 1.

Caution When an external clock input is selected with CSIM1 bits 1, 0 (SCL101, SCL100) are set to 0, set ADTC0 bits 2, 1 (STRB0, BUSY10) to 0, 0. (when an external clock is input, the hand-shake control cannot be performed).

Remark ×: don't care

(c) Automatic data transmit/receive interval specification register (ADTI0)

This register sets the automatic data transmit/receive function data transfer interval.

ADTI0 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Address: FF6BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTI0	ADTI07	0	0	ADTI04	ADTI03	ADTI02	ADTI01	ADTI00

ADTI07	Data transfer interval control
0	No control of interval by ADTI0 Note 1
1	Control of interval by ADTI0 (ADTI00 to ADTI04)

ADTI04	ADTI03	ADTI02	ADTI01	ADTI00	Data transfer interval specification (f_X = 8.38 MHz, f_{SCK} = 1.05 MHz) Note 2
0	0	0	0	0	1.90 μs + 0.5/f_{SCK}
0	0	0	0	1	
0	0	0	1	0	2.86 μs + 0.5/f_{SCK}
0	0	0	1	1	3.81 μs + 0.5/f_{SCK}
0	0	1	0	0	4.76 μs + 0.5/f_{SCK}
0	0	1	0	1	5.71 μs + 0.5/f_{SCK}
0	0	1	1	0	6.67 μs + 0.5/f_{SCK}
0	0	1	1	1	7.62 μs + 0.5/f_{SCK}
0	1	0	0	0	8.57 μs + 0.5/f_{SCK}
0	1	0	0	1	9.52 μs + 0.5/f_{SCK}
0	1	0	1	0	10.5 μs + 0.5/f_{SCK}
0	1	0	1	1	11.4 μs + 0.5/f_{SCK}
0	1	1	0	0	12.4 μs + 0.5/f_{SCK}
0	1	1	0	1	13.3 μs + 0.5/f_{SCK}
0	1	1	1	0	14.3 μs + 0.5/f_{SCK}
0	1	1	1	1	15.2 μs + 0.5/f_{SCK}

Notes 1. The interval time is $2/f_{SCK}$.

2. The data transfer interval time is calculated from the following expressions (n: Value set to ADTI00 to ADTI04).

<1> n = 0

$$\text{Interval time} = \frac{2}{f_{SCK}} + \frac{0.5}{f_{SCK}}$$

<2> n = 1 to 31

$$\text{Interval time} = \frac{n+1}{f_{SCK}} + \frac{0.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI0 during operation of automatic data transmit/receive function.

2. Be sure to set bits 5 and 6 to 0.

3. When controlling the interval time of automatic transmit/receive data transfer by using ADTI0, busy control is invalid. (Refer to 14.4.3 (4) (a) Busy control option.)

Remark f_X: Main system clock oscillation frequency

f_{SCK}: Serial clock frequency

Address: FF6BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTI0	ADTI07	0	0	ADTI04	ADTI03	ADTI02	ADTI01	ADTI00

ADTI04	ADTI03	ADTI02	ADTI01	ADTI00	Data transfer interval specification (f_X = 8.38 MHz, f_{SCK} = 1.05 MHz)[Note]
1	0	0	0	0	16.2 μs + 0.5/f_{SCK}
1	0	0	0	1	17.1 μs + 0.5/f_{SCK}
1	0	0	1	0	18.1 μs + 0.5/f_{SCK}
1	0	0	1	1	19.0 μs + 0.5/f_{SCK}
1	0	1	0	0	20.0 μs + 0.5/f_{SCK}
1	0	1	0	1	21.0 μs + 0.5/f_{SCK}
1	0	1	1	0	21.9 μs + 0.5/f_{SCK}
1	0	1	1	1	22.9 μs + 0.5/f_{SCK}
1	1	0	0	0	23.8 μs + 0.5/f_{SCK}
1	1	0	0	1	24.8 μs + 0.5/f_{SCK}
1	1	0	1	0	25.7 μs + 0.5/f_{SCK}
1	1	0	1	1	26.7 μs + 0.5/f_{SCK}
1	1	1	0	0	27.6 μs + 0.5/f_{SCK}
1	1	1	0	1	28.6 μs + 0.5/f_{SCK}
1	1	1	1	0	29.5 μs + 0.5/f_{SCK}
1	1	1	1	1	30.5 μs + 0.5/f_{SCK}

Note The data transfer interval time is calculated from the following expressions (n: Value set to ADTI00 to ADTI04).

<1> n = 0

$$\text{Interval time} = \frac{2}{f_{SCK}} + \frac{0.5}{f_{SCK}}$$

<2> n = 1 to 31

$$\text{Interval time} = \frac{n+1}{f_{SCK}} + \frac{0.5}{f_{SCK}}$$

Cautions
1. **Do not write ADTI0 during operation of automatic data transmit/receive function.**
2. **Be sure to set bits 5 and 6 to 0.**
3. **When controlling the interval time of automatic transmit/receive data transfer by using ADTI0, busy control is invalid. (Refer to 14.4.3 (4) (a) Busy control option.)**

Remark f_X: Main system clock oscillation frequency
f_{SCK}: Serial clock frequency

(2) Automatic transmit/receive data setting

(a) Transmit data setting

<1> Write transmit data from the least significant address FAC0H of buffer RAM (up to FADFH at maximum). The transmit data should be in the order from high-order address to low-order address.
<2> Set to the automatic data transmit/receive address pointer (ADTP0) the value obtained by subtracting 1 from the number of transmit data bytes.

(b) Automatic transmit/receive mode setting

<1> Set bit 7 (CSIE10) and bit 5 (ATE0) of the serial operating mode register 1 (CSIM1) to 1.
<2> Set bit 7 (RE0) of the automatic data transmit/receive control register (ADTC0) to 1.
<3> Set a data transmit/receive interval in the automatic data transmit/receive interval specification register (ADTI0).
<4> Write any value to the serial I/O shift register 1 (SIO1) (transfer start trigger).

Caution Writing any value to SIO1 orders the start of automatic transmit/receive operation and the written value has no meaning.

The following operations are automatically carried out when (a) and (b) are carried out.

- After the buffer RAM data specified with ADTP0 is transferred to SIO1, transmission is carried out (start of automatic transmission/reception).
- The received data is written to the buffer RAM address specified with ADTP0.
- ADTP0 is decremented and the next data transmission/reception is carried out. Data transmission/reception continues until the ADTP0 decremental output becomes 00H and address FAC0H data is output (end of automatic transmission/reception).
- When automatic transmission/reception is terminated, bit 3 (TRF0) of ADTC0 is cleared to 0.

(3) Communication operation

(a) Basic transmit/receive mode

This transmit/receive mode is the same as the 3-wire serial I/O mode in which a specified number of data are transmitted/received in 8-bit units.

Serial transfer is started when any data is written to the serial I/O shift register 1 (SIO1) while bit 7 (CSIE10) of the serial operating mode register 1 (CSIM1) is set to 1.

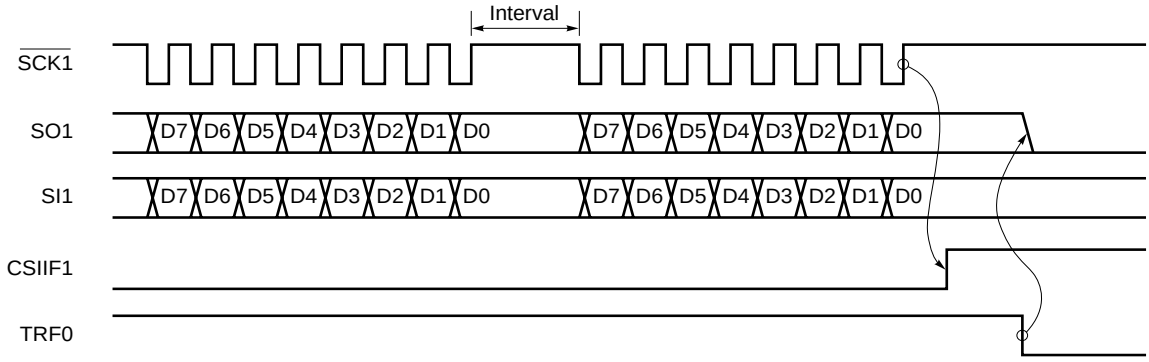
Upon completion of transmission of the last byte, the interrupt request flag (CSIIF1) is set. The termination of automatic transmission/reception should be checked by using bit 3 (TRF0) of the automatic data transmit/receive control register (ADTC0), not by CSIIF1 because CSIIF1 of the interrupt request flag is cleared if an interrupt is accepted.

If busy control and strobe control are not executed, the P80/STB and P81/BUSY pins can be used as normal input/output ports.

Figure 14-7 shows the basic transmit/receive mode operation timings, and Figure 14-8 shows the operation flowchart.

Figure 14-9 shows buffer RAM operation at 6-byte transmission.

Figure 14-7. Basic Transmit/Receive Mode Operation Timings

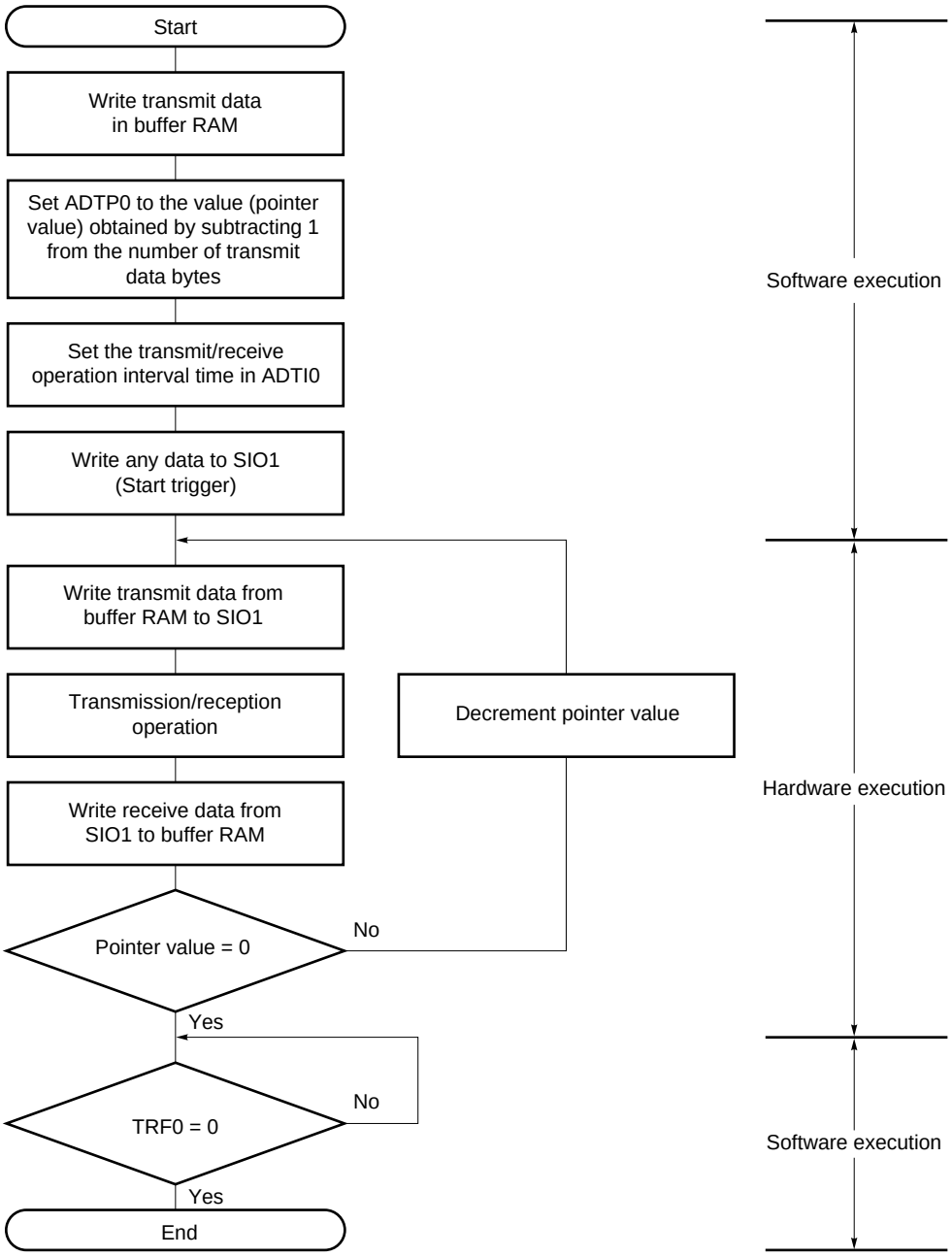


Cautions

1. **Because, in the basic transmit/receive mode, the automatic transmit/receive function writes/reads data to/from the buffer RAM after 1-byte transmission/reception, an interval is inserted till the next transmission/reception. As the buffer RAM write/read is performed at the same time as CPU processing, the maximum interval is dependent upon CPU processing and the value of the automatic data transmit/receive interval specification register (ADTI0) (refer to 14.4.3 (6) Automatic data transmit/receive interval).**
2. **When TRF0 is cleared, the SO1 pin becomes low level.**

Remark CSIIF1: Interrupt request flag
TRF0: Bit 3 of automatic data transmit/receive control register (ADTC0)

Figure 14-8. Basic Transmit/Receive Mode Flowchart



ADTP0: Automatic data transmit/receive address pointer
ADTI0: Automatic data transmit/receive interval specification register
SIO1: Serial I/O shift register 1
TRF0: Bit 3 of automatic data transmit/receive control register (ADTC0)

In 6-byte transmission/reception (bit 6 (ARLD0) and bit 7 (RE0) of the automatic data transmit/receive control register (ADTC0) = 0, and 1, respectively) in basic transmit/receive mode, buffer RAM operates as follows.

(i) Before transmission/reception (refer to Figure 14-9 (a))

After any data has been written to SIO1 (start trigger: this data is not transferred), transmit data 1 (T1) is transferred from the buffer RAM to SIO1. When transmission of the first byte is completed, the receive data 1 (R1) is transferred from SIO1 to the buffer RAM, and automatic data transmit/receive address pointer (ADTP0) is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1.

(ii) 4th byte transmit/receive point (refer to Figure 14-9 (b))

Transmission/reception of the third byte is completed, and transmit data 4 (T4) is transferred from the buffer RAM to SIO1. When transmission of the fourth byte is completed, the receive data 4 (R4) is transferred from SIO1 to the buffer RAM, and ADTP0 is decremented.

(iii) Completion of transmission/reception (refer to Figure 14-9 (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 14-9. Buffer RAM Operation in 6-Byte Transmission/Reception (in Basic Transmit/Receive Mode) (1/2)

(a) Before transmission/reception

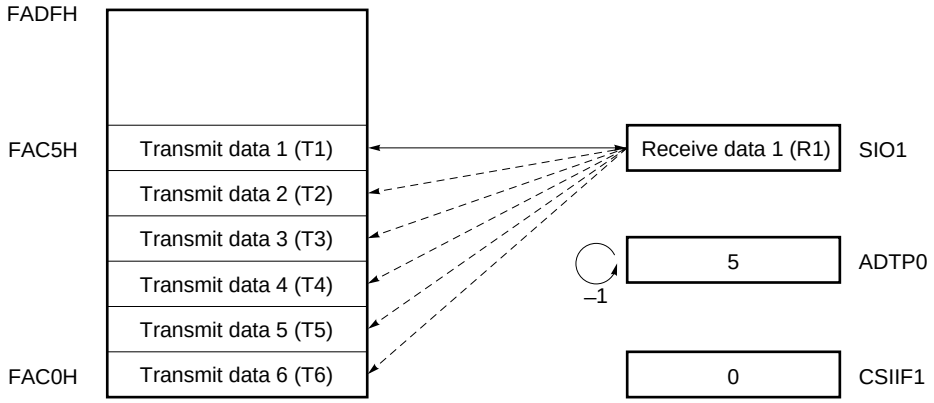
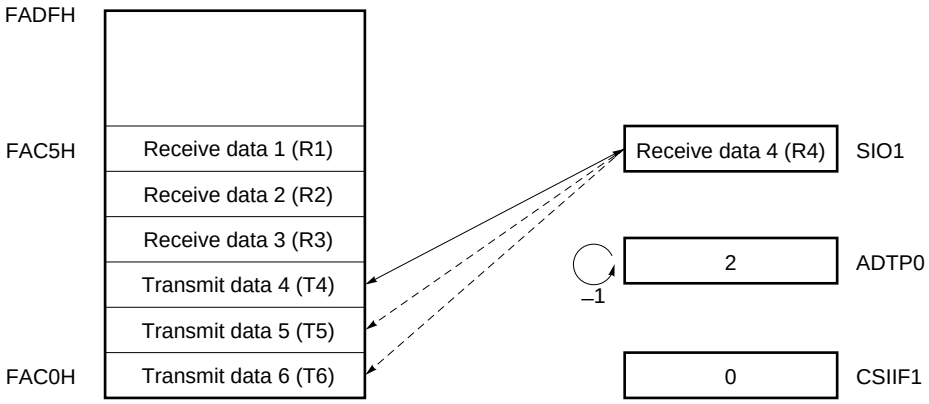
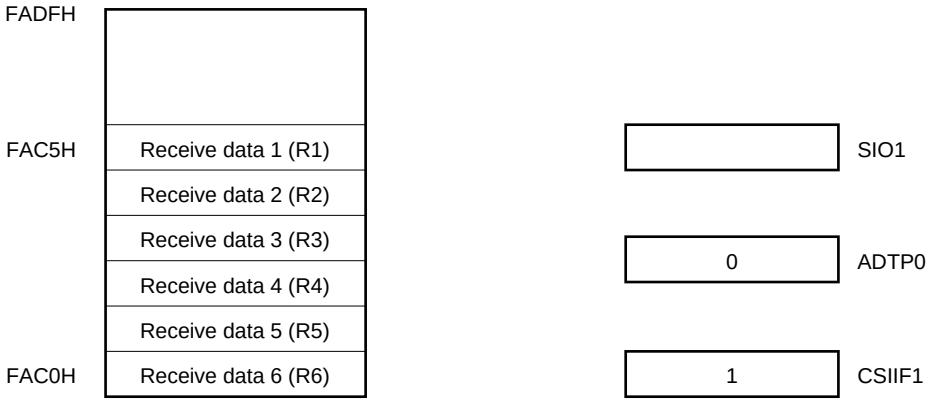


Figure 14-9. Buffer RAM Operation in 6-Byte Transmission/Reception (in Basic Transmit/Receive Mode) (2/2)

(b) 4th byte transmission/reception



(c) Completion of transmission/reception



(b) Basic transmit mode

In this mode, the specified number of 8-bit unit data are transmitted.

Serial transfer is started when any data is written to the serial I/O shift register 1 (SIO1) while bit 7 (CSIE10) of the serial operating mode register 1 (CSIM1) is set to 1, and bit 7 (RE0) of the automatic data transmit/receive control register (ADTC0) is set to 0.

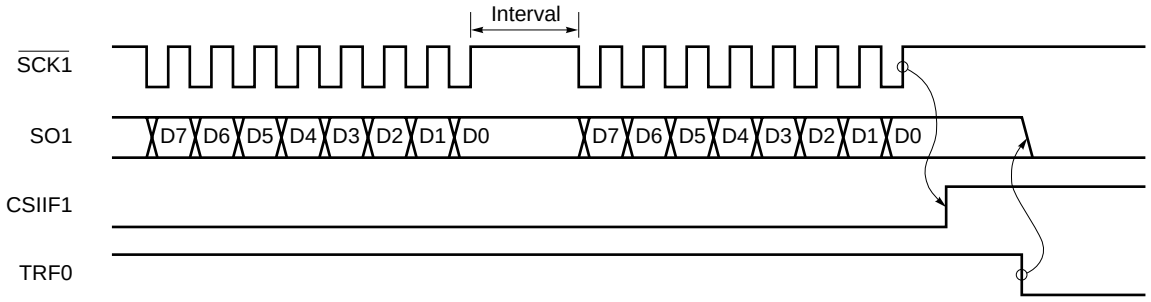
Upon completion of transmission of the last byte, the interrupt request flag (CSIIF1) is set. The termination of automatic transmission/reception should be checked by using bit 3 (TRF0) of the automatic data transmit/receive control register (ADTC0), not by CSIIF1.

If receive operation, busy control and strobe control are not executed, the P84/SI1, P81/BUSY, and P80/STB pins can be used as normal input ports.

Figure 14-10 shows the basic transmit mode operation timings, and Figure 14-11 shows the operation flowchart.

Figure 14-12 shows buffer RAM operation in 6-byte repeat transmission.

Figure 14-10. Basic Transmit Mode Operation Timings



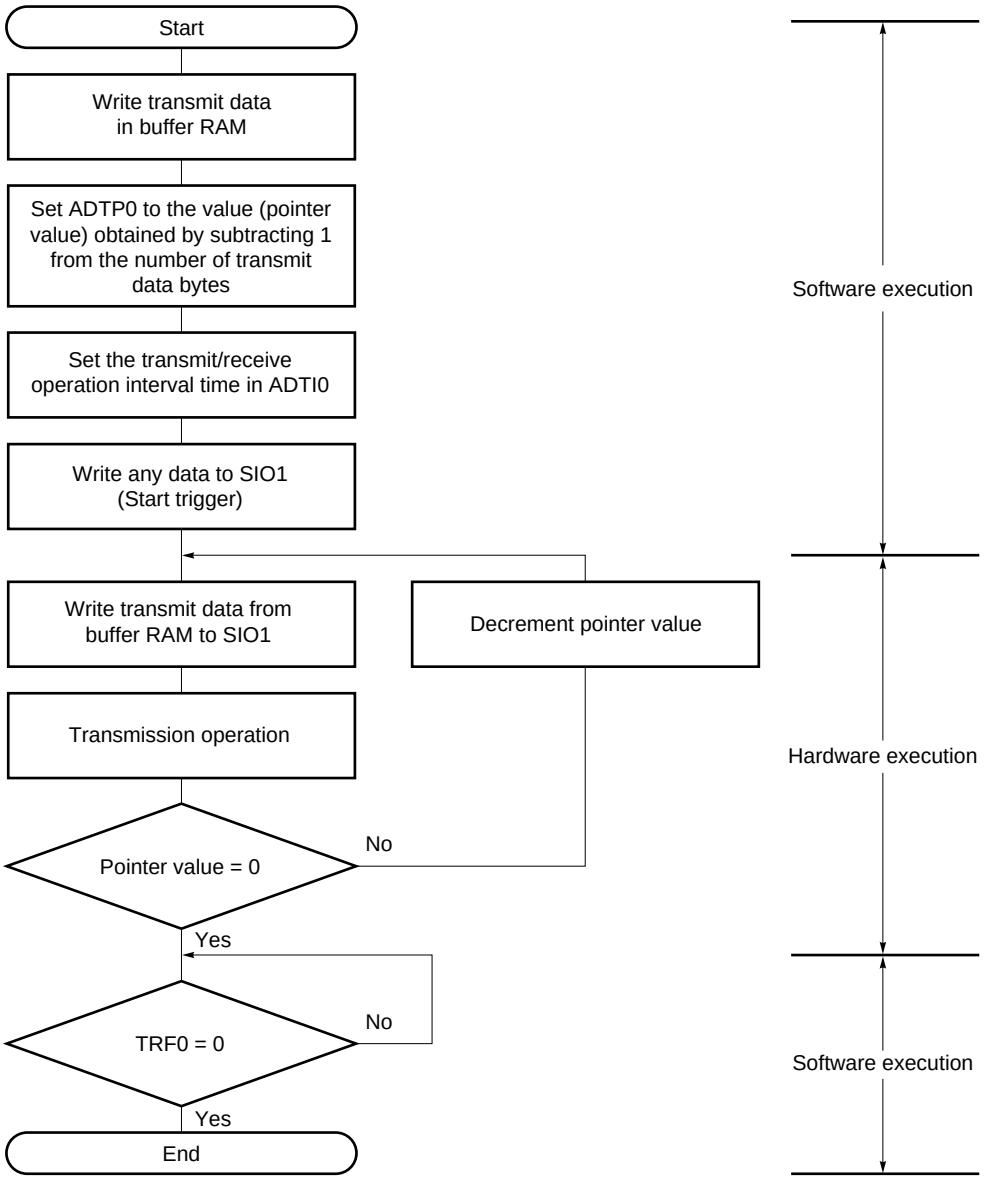
Cautions 1. Because, in the basic transmit mode, the automatic transmit/receive function reads data from the buffer RAM after 1-byte transmission, an interval is inserted till the next transmission. As the buffer RAM read is performed at the same time as CPU processing, the maximum interval is dependent upon CPU processing and the value of the automatic data transmit/receive interval specification register (ADTI0) (refer to 14.4.3 (6) Automatic data transmit/receive interval).

2. When TRF0 is cleared, the SO1 pin becomes low level.

Remark CSIIF1: Interrupt request flag

TRF0: Bit 3 of automatic data transmit/receive control register (ADTC0)

Figure 14-11. Basic Transmit Mode Flowchart



ADTP0: Automatic data transmit/receive address pointer
ADTI0: Automatic data transmit/receive interval specification register
SIO1: Serial I/O shift register 1
TRF0: Bit 3 of automatic data transmit/receive control register (ADTC0)

In 6-byte transmission (bit 6 (ARLD0) and bit 7 (RE0) of the automatic data transmit/receive control register (ADTC0) are 0) in basic transmit mode, buffer RAM operates as follows.

(i) Before transmission (refer to Figure 14-12 (a))

After any data has been written to SIO1 (start trigger: this data is not transferred), transmit data 1 (T1) is transferred from the buffer RAM to SIO1. When transmission of the first byte is completed, ADTP0 is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1.

(ii) 4th byte transmission point (refer to Figure 14-12 (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, ADTP0 is decremented.

(iii) Completion of transmission/reception (refer to Figure 14-12 (c))

When transmission of the sixth byte is completed, the interrupt request flag (CSIIF1) is set (INTCSI1 generation).

Figure 14-12. Buffer RAM Operation in 6-Byte Transmission (in Basic Transmit Mode) (1/2)

(a) Before transmission

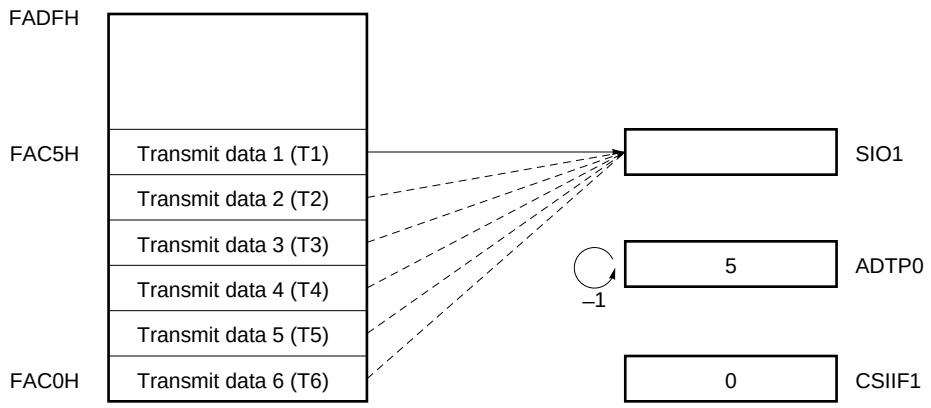
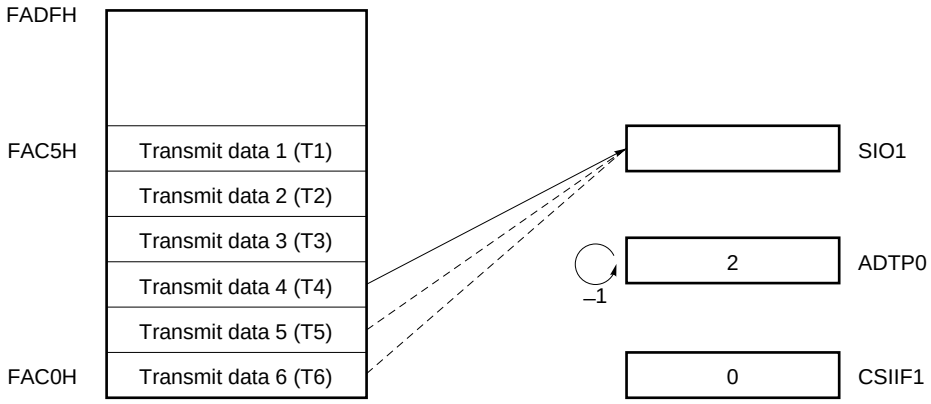
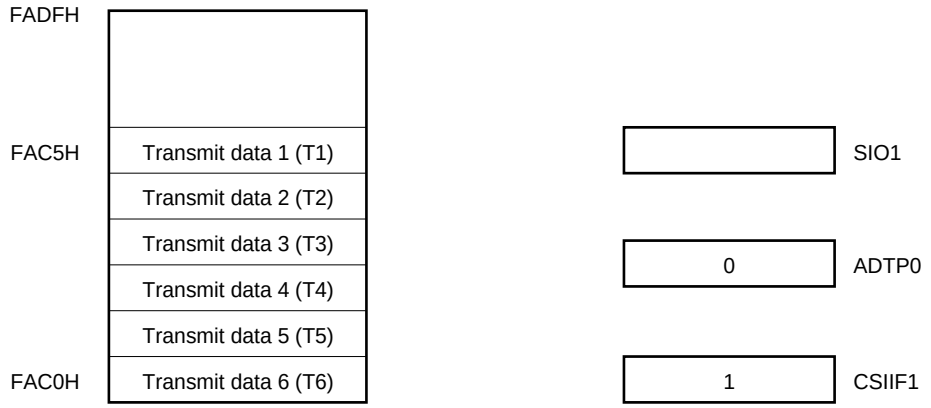


Figure 14-12. Buffer RAM Operation in 6-Byte Transmission (in Basic Transmit Mode) (2/2)

(b) 4th byte transmission point



(c) Completion of transmission/reception



(c) Repeat transmit mode

In this mode, data stored in the buffer RAM is transmitted repeatedly.

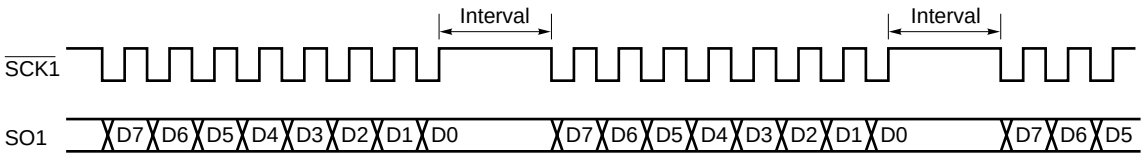
Serial transfer is started by writing any data to serial I/O shift register 1 (SIO1) when bit 7 (CSIE10) of the serial operating mode register 1 (CSIM1) is set to 1, and bit 7 (RE0) of the automatic data transmit/receive control register (ADTC0) is set to 0.

Unlike the basic transmission mode, after the last byte (data in address FAC0H) has been transmitted, the interrupt request flag (CSIIF1) is not set, the value at the time when the transmission was started is set in the automatic data transmit/receive address pointer (ADTP0) again, and the buffer RAM contents are transmitted again.

When a reception operation, busy control and strobe control are not performed, the P84/SI1, P81/BUSY, and P80/STB pins can be used as ordinary input/output ports.

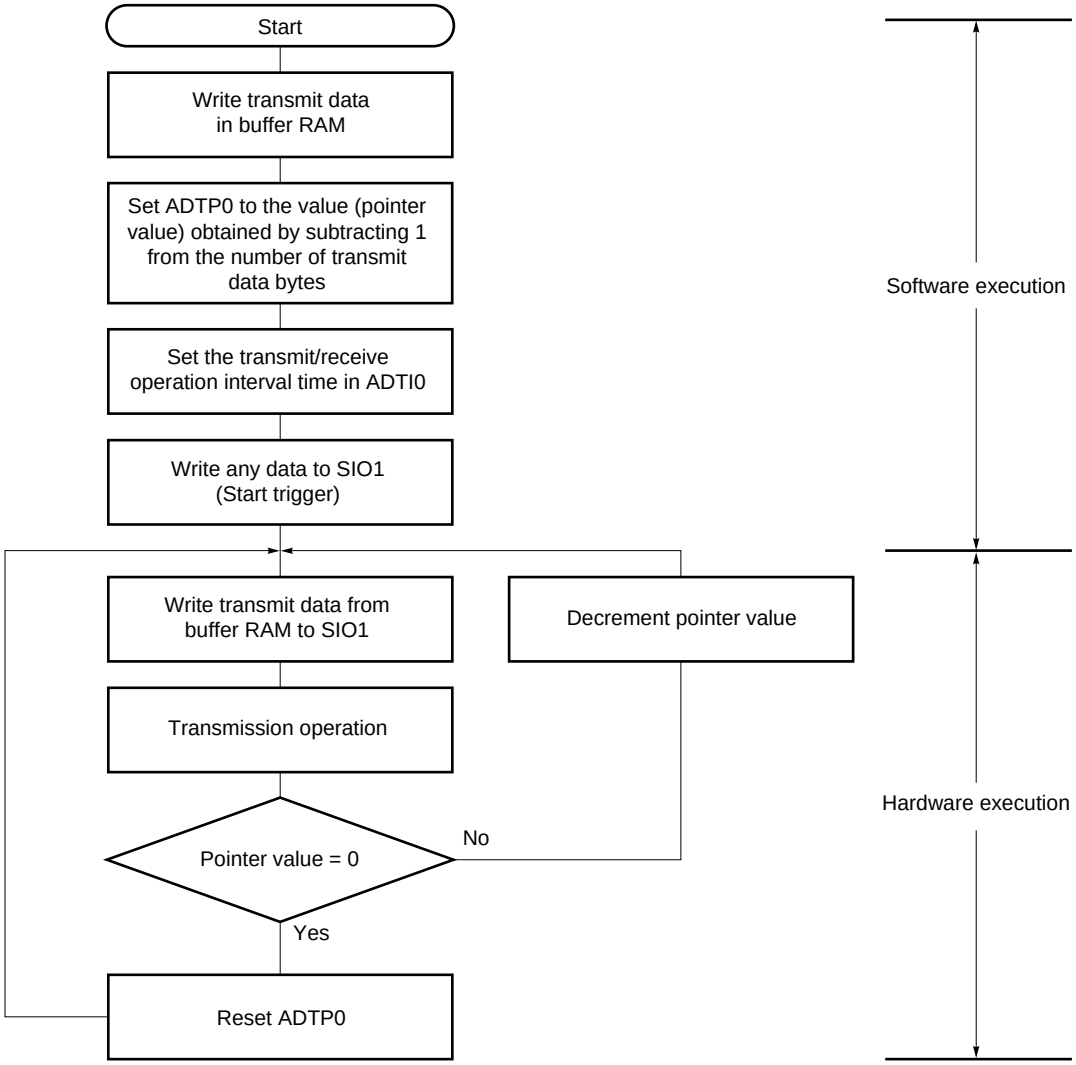
The repeat transmit mode operation timing is shown in Figure 14-13, and the operation flowchart in Figure 14-14.

Figure 14-13. Repeat Transmit Mode Operation Timing



Caution Since, in the repeat transmit 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 (ADTI0) (refer to 14.4.3 (6) Automatic data transmit/receive interval).

Figure 14-14. Repeat Transmit Mode Flowchart



ADTP0: Automatic data transmit/receive address pointer
ADTI0: Automatic data transmit/receive interval specification register
SIO1: Serial I/O shift register 1

In 6-byte transmission (bit 6 (ARLD0) and bit 7 (RE0) of the automatic data transmit/receive control register (ADTC0) are 1 and 0, respectively) in repeat transmit mode, buffer RAM operates as follows.

(i) Before transmission (refer to Figure 14-15 (a))

After any data has been written to SIO1 (start trigger: this data is not transferred), transmit data 1 (T1) is transferred from the buffer RAM to SIO1. When transmission of the first byte is completed, ADTP0 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 14-15 (b))

When transmission of the sixth byte is completed, the interrupt request flag (CSIIF1) is not set. The previous pointer value is assigned to the ADTP0.

(iii) 7th byte transmission point (refer to Figure 14-15 (c))

Transmit data 1 (T1) is transferred from the buffer RAM to SIO1 again. When transmission of the first byte is completed, ADTP0 is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1.

Figure 14-15. Buffer RAM Operation in 6-Byte Transmission (in Repeat Transmit Mode) (1/2)

(a) Before transmission

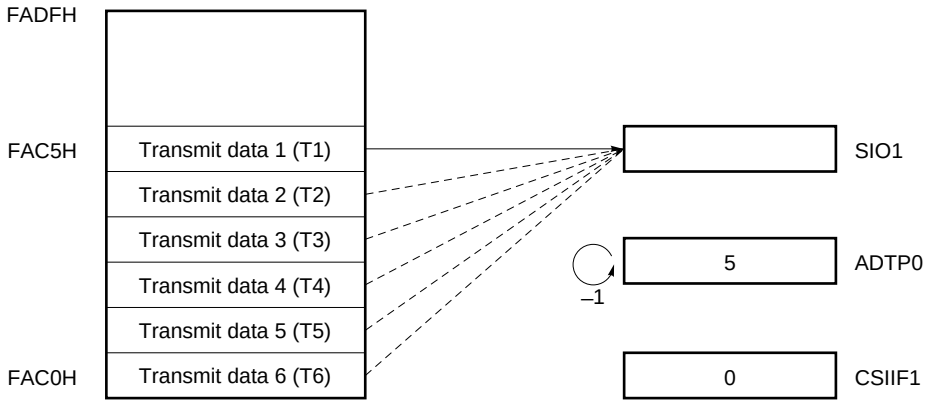
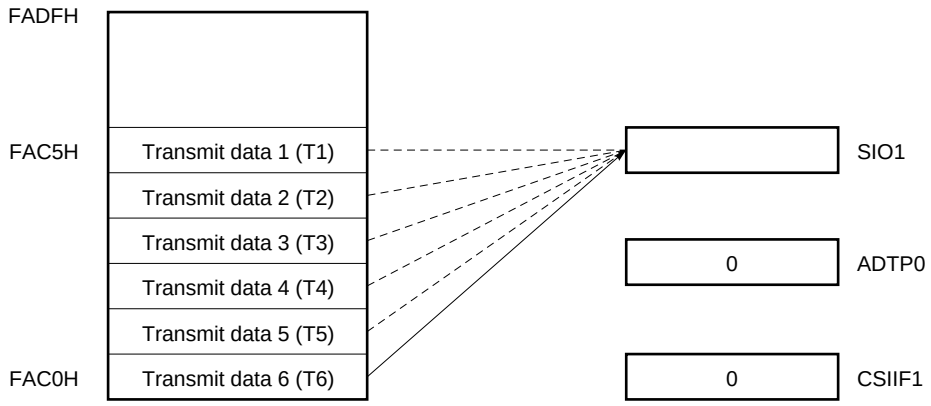
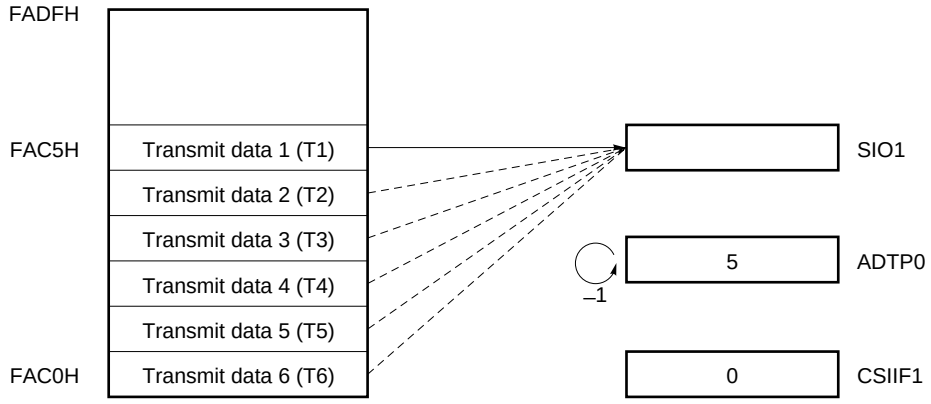


Figure 14-15. Buffer RAM Operation in 6-Byte Transmission (in Repeat Transmit Mode) (2/2)

(b) Upon completion of transmission of 6 bytes



(c) 7th byte transmission point



(d) Automatic transmission/reception suspension and restart

Automatic transmission/reception can be temporarily suspended by setting bit 7 (CSIE10) of the serial operating mode register 1 (CSIM1) to 0.

If, during 8-bit data transfer, the transmission/reception is not suspended, it is suspended upon completion of 8-bit data transfer.

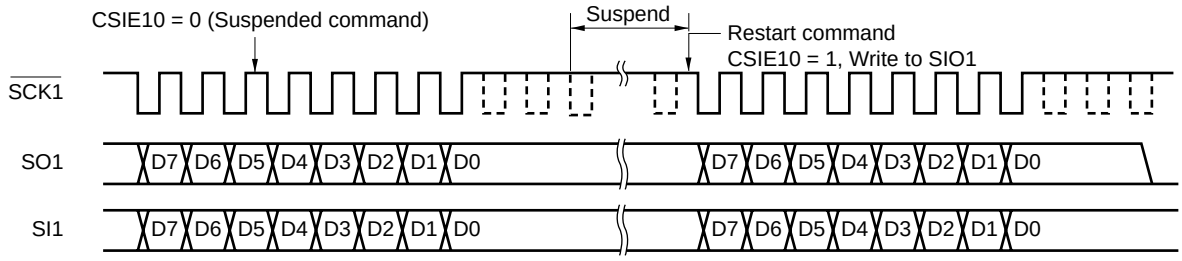
When suspended, bit 3 (TRF0) of the automatic data transmit/receive control register (ADTC0) is set to 0 after transfer of the 8th bit, and all the port pins used with the serial interface pins for dual function (P84/SI1, P83/SO1, P82/$\overline{\text{SCK1}}$, P81/BUSY, and P80/STB) are set to the port mode.

During restart of transmission/reception, remaining data can be transferred by setting CSIE10 to 1 and writing any data to the serial I/O shift register 1 (SIO1).

Cautions 1. If the HALT instruction is executed during automatic transmission/reception, transfer is suspended and the HALT mode is set even if 8-bit data transfer is in progress.

2. When suspending automatic transmission/reception, do not change the operating mode to 3-wire serial I/O mode while TRF0 = 1.

Figure 14-16. Automatic Transmission/Reception Suspension and Restart



CSIE10: Bit 7 of serial operating mode register 1 (CSIM1)

(4) Synchronization control

Busy control and strobe control are functions to synchronize transmission/reception between the master device and a slave device.

By using these functions, a shift in bits being transmitted or received can be detected.

(a) Busy control option

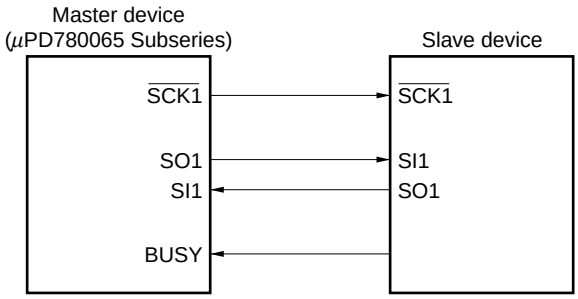
Busy control is a function to keep the serial transmission/reception by the master device waiting while the busy signal output by a slave device to the master is active.

When using this busy control option, the following conditions must be satisfied.

- Bit 5 (ATE0) of the serial operating mode register 1 (CSIM1) is set to 1.
- Bit 1 (BUSY10) of the automatic data transmit/receive control register (ADTC0) is set to 1.

Figure 14-17 shows the system configuration of the master device and a slave device when the busy control option is used.

Figure 14-17. System Configuration When Busy Control Option Is Used



The master device inputs the busy signal output by the slave device to the BUSY/P81 pin. The master device samples the input busy signal in synchronization with the falling of the serial clock. Even if the busy signal becomes active while 8-bit data is being transmitted or received, transmission/reception by the master is not kept waiting. If 2 clocks after completion of transmission/reception of the 8-bit data, the busy signal is active at the rising edge of the serial clock , the busy input becomes valid. Following this, the master transmission/reception is kept waiting while the busy signal is active.

The active level of the busy signal is set by bit 0 (BUSY00) of ADTC0.

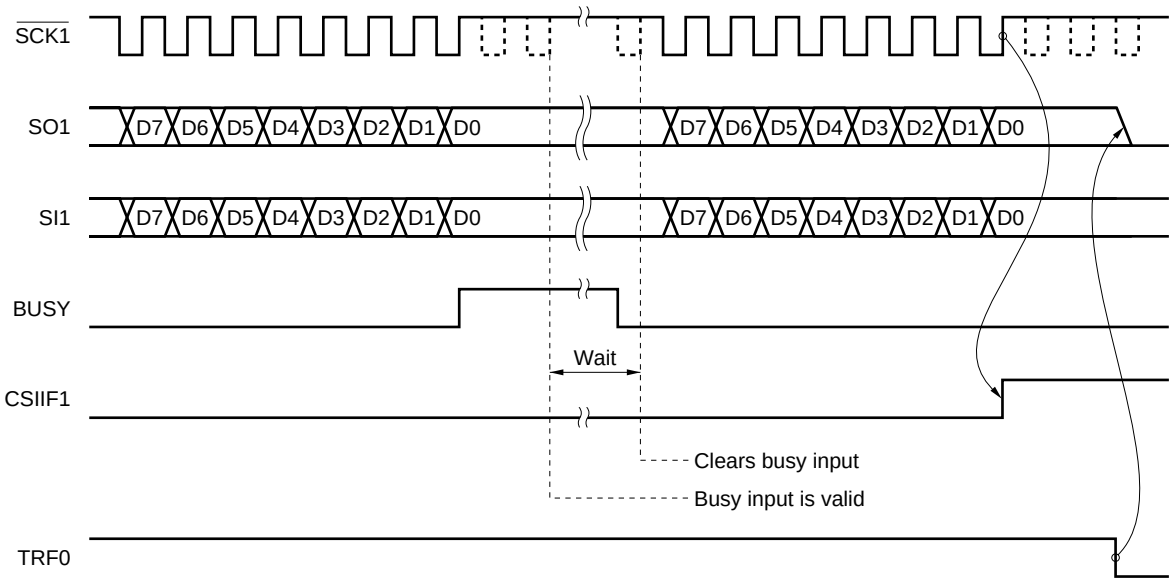
BUSY00 = 0: Active high

BUSY00 = 1: Active low

When using the busy control option, select the internal clock as the serial clock. Control with the busy signal cannot be implemented with the external clock.
Figure 14-18 shows the operation timing when the busy control option is used.

Caution The busy control cannot be used simultaneously with the interval time control function of the automatic data transmit/receive interval specification register (ADTI0). If used, the busy control is invalid.

Figure 14-18. Operation Timing When Busy Control Option Is Used (when BUSY00 = 0)

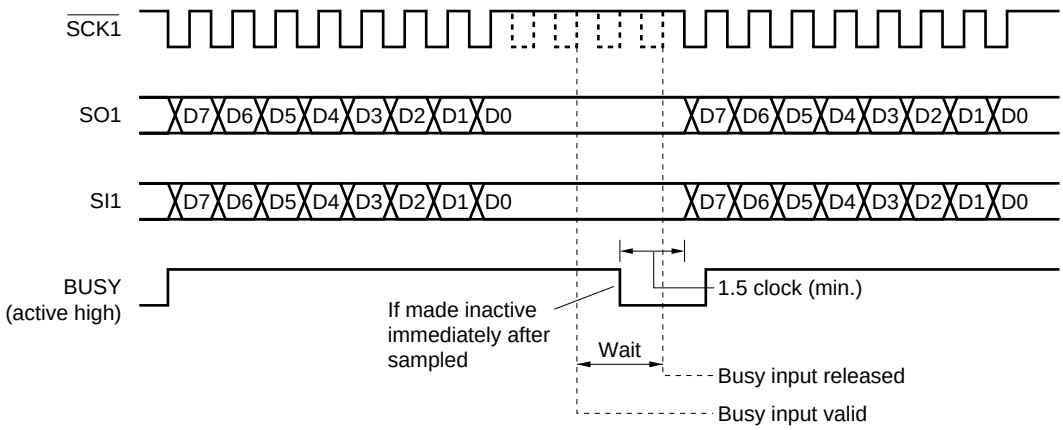


Caution If the TRF0 is cleared, the SO1 pin becomes low level.

Remark CSIIF1: Interrupt request flag
TRF0: Bit 3 of automatic data transmit/receive control register (ADTC0)

When the busy signal becomes inactive, waiting is released. If the sampled busy signal is inactive, transmission/reception of the next 8-bit data is started at the falling edge of the next clock.
Because the busy signal is asynchronous with the serial clock, it takes up to 1 clock until the busy signal is sampled, even if made inactive by the slave. It takes 0.5 clock until data transfer is started after the busy signal is sampled.
To accurately release waiting, the slave must keep the busy signal inactive at least for the duration of 1.5 clocks.
Figure 14-19 shows the timing of the busy signal and wait release. This figure shows an example where the busy signal is active as soon as transmission/reception is started.

Figure 14-19. Busy Signal and Wait Release (when BUSY00 = 0)



(b) Strobe control option

Strobe control is a function for synchronizing data transmission/reception between the master and slave devices. The master device outputs the strobe signal from the STB/P80 pin when 8-bit transmission/reception has been completed. By this signal, the slave device can determine the timing of the end of data transmission. Therefore, synchronization is established even if a bit shift occurs because noise is superimposed on the serial clock, and transmission of the next byte is not affected by the bit shift.

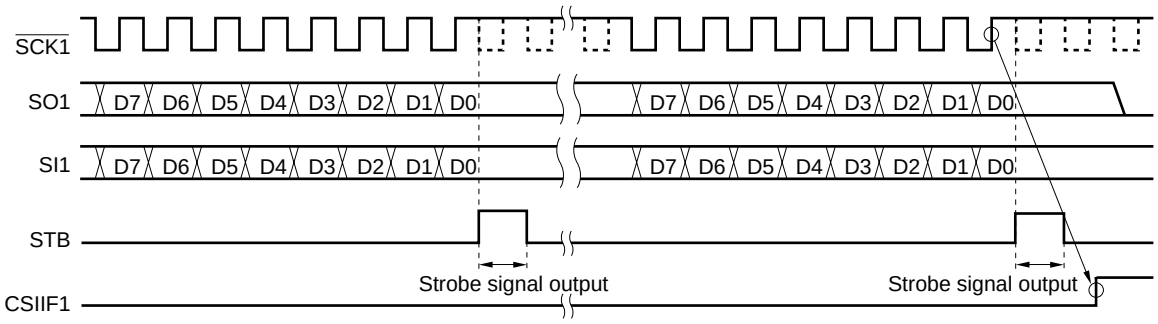
To use the strobe control option, the following conditions must be satisfied:

- Bit 5 (ATE0) of the serial operating mode register 1 (CSIM1) is set to 1.
- Bit 2 (STRB0) of the automatic data transmit/receive control register (ADTC0) is set to 1.

A strobe signal is output from the STB/P80 pin for the duration of 1 clock in synchronization with the falling of the ninth serial clock.

Figure 14-20 shows the operation timing when the strobe control option is used.

Figure 14-20. Operation Timing When Strobe Control Option Is Used



(c) Busy & strobe control option

Usually, the busy control and strobe control options are simultaneously used as handshake signals. In this case, the strobe signal is output from the STB/P80 pin, and the BUSY/P81 pin is sampled, and transmission/reception can be kept waiting while the busy signal is input.

When the strobe control option is not used, the P80/STB pin can be used as a normal I/O port pin.

To use the busy & strobe control option, the following conditions must be satisfied.

- Bit 5 (ATE0) of the serial operating mode register 1 (CSIM1) is set to 1.
- Bit 2 (STRB0) and bit 1 (BUSY10) of the automatic data transmit/receive control register (ADTC0) are set to 1.

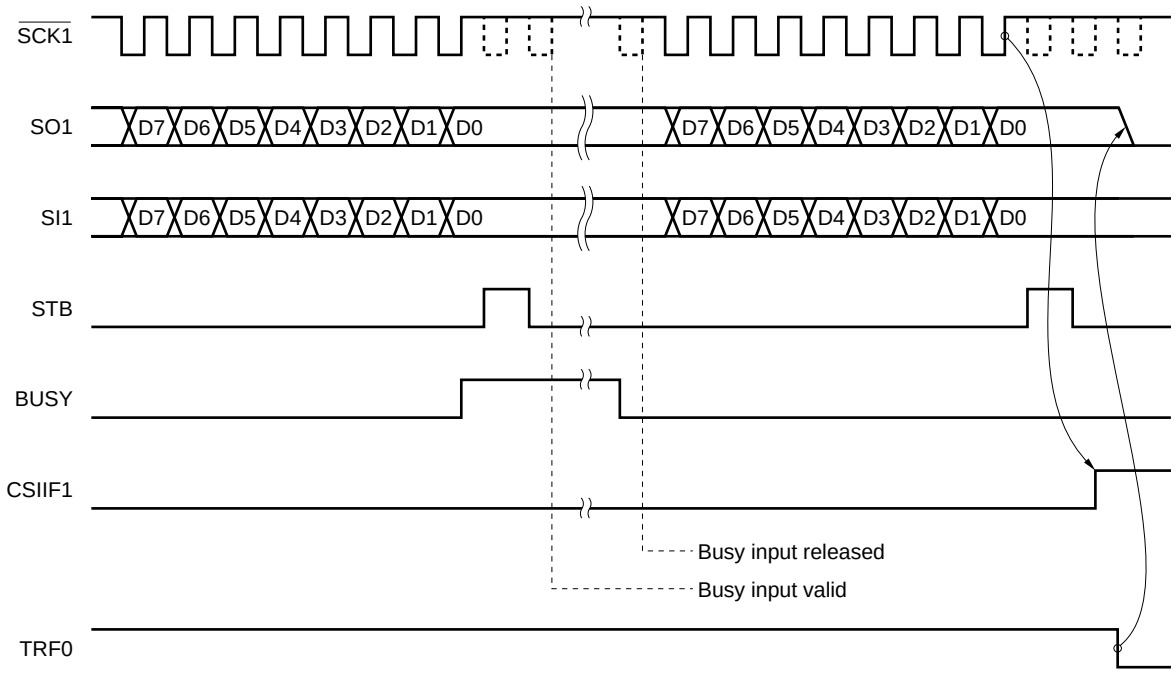
The active level of the busy signal is set by bit 0 (BUSY00) of ADTC0.

BUSY00 = 0: Active high
BUSY00 = 1: Active low

Figure 14-21 shows the operation timing when the busy & strobe control options are used.

When the strobe control option is used, the interrupt request flag (CSIIF1) that is set on completion of transmission/reception is set after the strobe signal is output.

Figure 14-21. Operation Timing When Busy & Strobe Control Options Are Used (when BUSY00 = 0)



Caution When TRF0 is cleared, the SO1 pin becomes low level.

Remark CSIIF1: Interrupt request flag
TRF0: Bit 3 of automatic data transmit/receive control register (ADTC0)

(d) Bit shift detection by busy signal

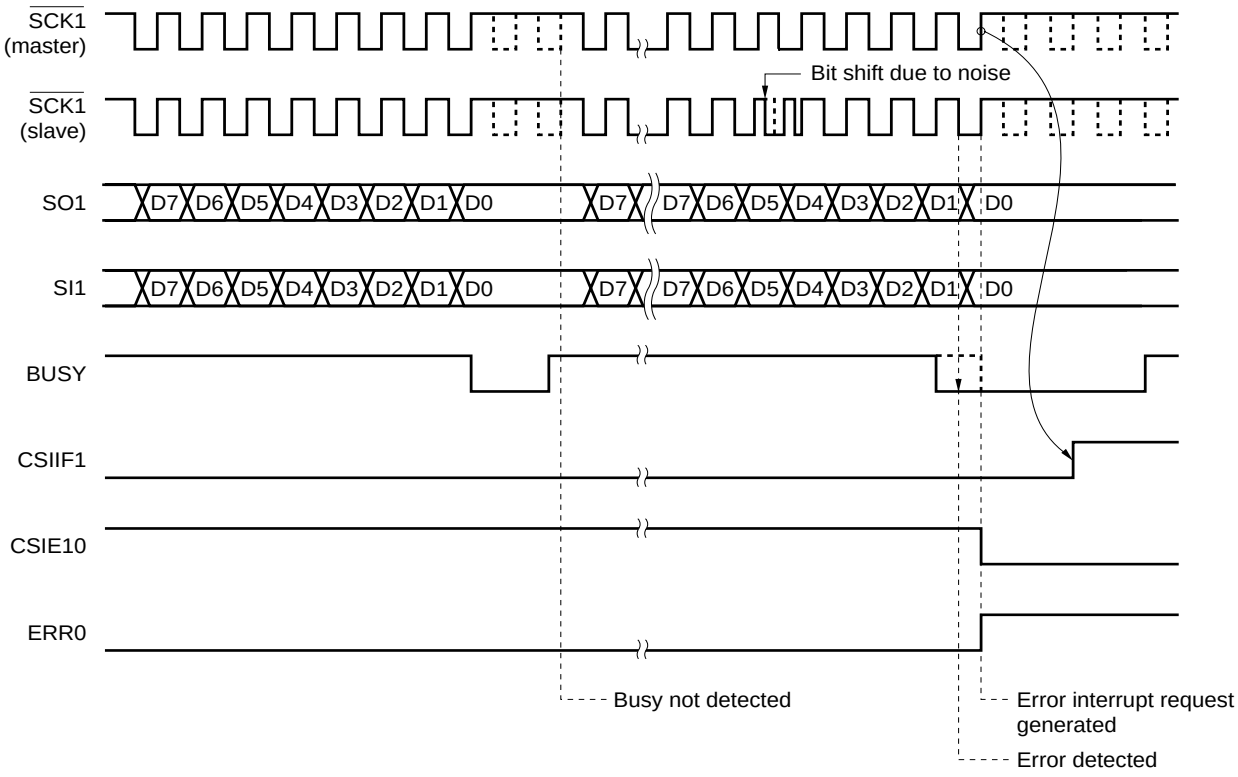
During automatic transmission/reception, a bit shift of the serial clock of the slave device may occur because noise is superimposed on the serial clock signal output by the master device. Unless the strobe control option is used at this time, the bit shift affects transmission of the next byte. In this case, the master can detect the bit shift by checking the busy signal during transmission by using the busy control option. A bit shift is detected by using the busy signal as follows.

The slave outputs the busy signal after the rising edge of the eighth serial clock during data transmission/reception (in order not to keep transmission/reception waiting by the busy signal at this time, make the busy signal inactive within 2 clocks).

The master samples the busy signal in synchronization of the falling of the leading side of the serial clock. If a bit shift does not occur, all the eight serial clocks that have been sampled are inactive. If the sampled serial clocks are active, it is assumed that a bit shift has occurred, and error processing is executed (by setting bit 4 (ERR0) of the automatic transmit/receive control register (ADTC0) to 1).

Figure 14-22 shows the operation timing of the bit shift detection function by the busy signal.

Figure 14-22. Operation Timing of Bit Shift Detection Function by Busy Signal (when BUSY00 = 1)



CSIIF1: Interrupt request flag

CSIE10: Bit 7 of serial operating mode register 1 (CSIM1)

ERR0: Bit 4 of automatic data transmit/receive control register (ADTC0)

(5) Timing of interrupt request signal generation

The interrupt request signal is generated in synchronization with the timing shown in Table 14-2.

Table 14-2. Timing of Interrupt Request Signal Generation

Operating Mode		Timing of Interrupt Request Signal
Single mode	Master mode	10th serial clock at end of transfer
	Slave mode	8th serial clock at end of transfer
Repetitive transmit mode		Not generated
If bit shift occurs during transmission/reception		8th serial clock

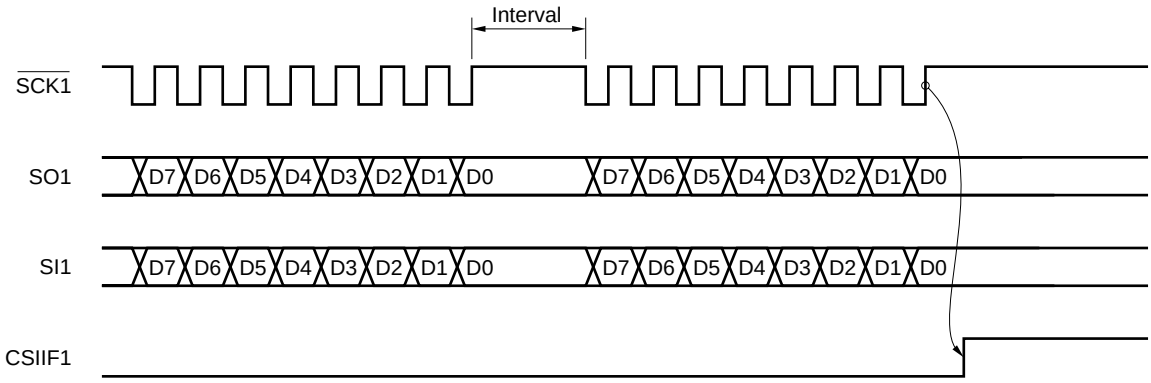
(6) Interval time of automatic transmission/reception

Because read/write to/from the buffer RAM using the automatic transmit/receive function is performed asynchronously with CPU processing, the interval time is dependent on the CPU processing of the timing of the eighth rising of the serial clock and the set value of the automatic data transmit/receive interval specification register (ADTI0). Whether the interval time is dependent on ADTI0 is selected by setting of the bit 7 (ADTI07) of ADTI0. If ADTI07 is reset to 0, the interval time is 2/f_{SCK}. If ADTI07 is set to 1, the interval time determined by the set contents of ADTI0 or interval time (2/f_{SCK}) according to CPU processing is selected, whichever is is greater.

Figure 14-23 shows the interval time of automatic transmission/reception

Remark f_{SCK}: Serial clock frequency

Figure 14-23. Interval Time of Automatic Transmission/Reception



The following expression must be satisfied to access the buffer RAM.

1 transfer cycle + interval time ≥ Read access + Write access + CPU buffer RAM access (time)

In the case of a “high-speed CPU & low-speed SCK**Note**”, the interval time is not necessary. In the case of a “low-speed CPU & high-speed SCK**Note**”, the interval time is necessary.

In this case, make sure that a sufficient interval time elapses by using the automatic data transmit/receive interval specification register (ADTI0), so that the above expression is satisfied.

Note The speeds of the CPU clock and SCK differ depending on the type of CPU core.

CHAPTER 15 SERIAL INTERFACE (SIO30)

15.1 Serial Interface (SIO30) Functions

The serial interface (SIO30) has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. For details, see **15.4.1 Operation stop mode**.

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

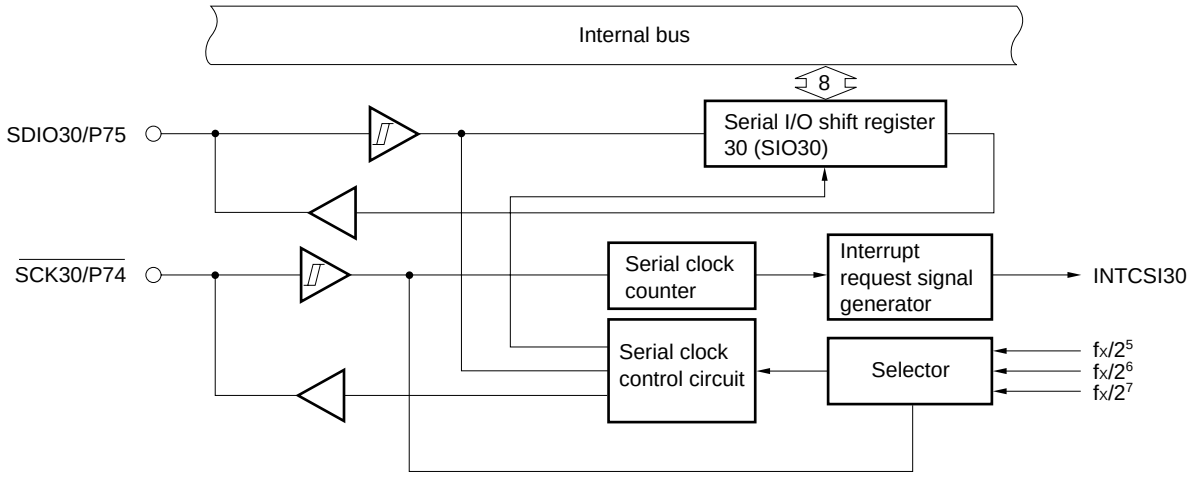
This is an 8-bit data transfer mode using two lines: a serial clock line ($\overline{\text{SCK30}}$) and serial data input/output line (SDIO30).

The first bit of the serial transferred 8-bit data is fixed as the MSB.

2-wire serial I/O mode is useful for connection to a peripheral I/O incorporating a clock-synchronous serial interface, or a display controller, etc. For details see **15.4.2 2-wire serial I/O mode**.

Figure 15-1 shows a block diagram of the serial interface (SIO30).

★ **Figure 15-1. Serial Interface (SIO30) Block Diagram**



15.2 Serial Interface (SIO30) Configuration

The serial interface (SIO30) includes the following hardware.

Table 15-1. Serial Interface (SIO30) Configuration

Item	Configuration
Registers	Serial I/O shift register 30 (SIO30)
Control registers	Serial operation mode register 30 (CSIM30)

(1) Serial I/O shift register 30 (SIO30)

This is an 8-bit register that performs parallel-serial conversion and serial transmit/receive (shift operations) synchronized with the serial clock.

SIO30 is set by an 8-bit memory manipulation instruction.

When “1” is set to bit 7 (CSIE30) of the serial operation mode register 30 (CSIM30), a serial operation can be started by writing data to or reading data from SIO30.

When transmitting, data written to SIO30 is output to the serial output (SO30).

When receiving, data is read from the serial input (SI30) and written to SIO30.

$\overline{\text{RESET}}$ input resets SIO30 to 00H.

Caution Do not access SIO30 during a transmit operation unless the access is triggered by a transfer start. (Read operation is disabled when MODE0 = 0 and write operation is disabled when MODE0 = 1.)

15.3 Register to Control Serial Interface (SIO30)

The serial interface (SIO30) uses the following type of register to control functions.

- Serial operation mode register 30 (CSIM30)

(1) Serial operation mode register 30 (CSIM30)

This register is used to enable or disable SIO30's serial clock, operation modes, and specific operations.
CSIM30 can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets CSIM30 to 00H.

Caution In 2-wire serial I/O mode, set the port mode register (PMXX) as follows. In each case, set the output latch to 0.

- **During serial clock output (master transmission or master reception)**
  Set P74 ($\overline{\text{SCK30}}$) to output mode (PM74 = 0)
- **During serial clock input (slave transmission or slave reception)**
  Set P74 to input mode (PM74 = 1)
- **During transmit mode**
  Set P75 (SDIO30) to output mode (PM75 = 0)
- **During receive mode**
  Set P75 (SDIO30) to input mode (PM75 = 1)

Figure 15-2. Format of Serial Operation Mode Register 30 (CSIM30)

Address: FFB0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM30	CSIE30	0	0	0	0	MODE0	SCL301	SCL300

CSIE30	Enable/disable specification for SIO30		
	Shift register operation	Serial counter	Port
0	Operation stop	Clear	Port function[Note]
1	Operation enable	Count operation enable	Serial function

MODE0	Transfer operation modes and flags	
	Operation mode	Transfer start trigger
0	Transmit mode	Write to SIO30
1	Receive mode	Read from SIO30

SCL301	SCL300	Clock selection
0	0	External clock input to $\overline{\text{SCK30}}$
0	1	$f_X/2^5$ (262 kHz)
1	0	$f_X/2^6$ (131 kHz)
1	1	$f_X/2^7$ (65.5 kHz)

Note When CSIE30 = 0 (SIO30 operation stop status), pins SDIO30 and $\overline{\text{SCK30}}$ can be used as port functions.

Remarks 1. f_X: main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz.

15.4 Serial Interface (SIO30) Operations

This section explains the two modes of serial interface SIO30.

15.4.1 Operation stop mode

Because serial transfer is not performed in this mode, power consumption can be reduced.

In addition, pins can be used as normal I/O ports.

(1) Register settings

Operation stop mode is set by serial operation mode register 30 (CSIM30).

CSIM30 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM0 to 00H.

Address: FFB0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM30	CSIE30	0	0	0	0	MODE0	SCL301	SCL300

CSIE30	SIO30 operation enable/disable specification		
	Shift register operation	Serial counter	Port
0	Operation stop	Clear	Port function[Note]
1	Operation enable	Count operation enable	Serial function

Note When CSIE30 = 0 (SIO30 operation stop status), pins SDIO30 and $\overline{\text{SCK30}}$ can be used as port functions.

15.4.2 2-wire serial I/O mode

The 2-wire serial I/O mode is useful for connection to a peripheral I/O incorporating a clock-synchronous serial interface, a display controller, etc.

This mode executes data transfers via two lines: a serial clock line ($\overline{SCK30}$) and serial data input/output line (SDIO30).

(1) Register settings

2-wire serial I/O mode is set by serial operation mode register 30 (CSIM30).

CSIM30 can be set by an 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input sets CSIM30 to 00H .

Caution In 2-wire serial I/O mode, set the port mode register (PMXX) as follows. In each case, set the output latch to 0.

- **During serial clock output (master transmission or master reception)**
  Set P74 ($\overline{SCK30}$) to output mode (PM74 = 0)
- **During serial clock input (slave transmission or slave reception)**
  Set P74 to input mode (PM74 = 1)
- **During transmit mode**
  Set P75 (SDIO30) to output mode (PM75 = 0)
- **During receive mode**
  Set P75 (SDIO30) to input mode (PM75 = 1)

Address: FFB0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM30	CSIE30	0	0	0	0	MODE0	SCL301	SCL300

CSIE30	Enable/disable specification for SIO30		
	Shift register operation	Serial counter	Port
0	Operation stop	Clear	Port function[Note]
1	Operation enable	Count operation enable	Serial function

MODE0	Transfer operation modes and flags	
	Operation mode	Transfer start trigger
0	Transmit mode	Write to SIO30
1	Receive mode	Read from SIO30

SCL301	SCL300	Clock selection
0	0	External clock input to $\overline{SCK30}$
0	1	$f_X/2^5$ (262 kHz)
1	0	$f_X/2^6$ (131 kHz)
1	1	$f_X/2^7$ (65.5 kHz)

Note When CSIE30 = 0 (SIO30 operation stop status), pins SDIO30 and $\overline{SCK30}$ can be used as port functions.

Remarks 1. f_X: main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz.

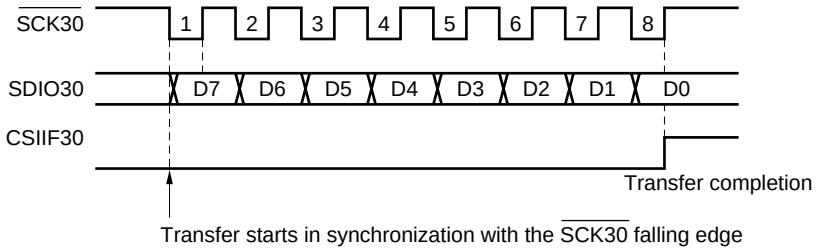
(2) Communication operations

In 2-wire serial I/O mode, data is transmitted and received in 8-bit units. Each bit of data is sent or received in synchronization with the serial clock.

Serial I/O shift register 30 (SIO30) is shifted in synchronization with the falling edge of the serial clock. Transmission data is held in the SDIO30 latch and is output from the SDIO30 pin. Data that is received via the SDIO30 pin in synchronization with the rising edge of the serial clock is latched to SIO30.

Completion of an 8-bit transfer automatically stops operation of SIO30 and sets interrupt request flag (CSIIF30).

Figure 15-3. Timing of 2-Wire Serial I/O Mode



(3) Transfer start

Serial transfer starts when the following two conditions have been satisfied and transfer data has been set (or read) to serial I/O shift register 30 (SIO30).

- SIO30 operation control bit (CSIE30) = 1
- After an 8-bit serial transfer, either the internal serial clock is stopped or $\overline{\text{SCK30}}$ is set to high level.
- Transmit mode
  When CSIE30 = 1 and MODE0 = 0, transfer starts when writing to SIO30.
- Receive mode
  When CSIE30 = 1 and MODE0 = 1, transfer starts when reading from SIO30.

Caution After data has been written to SIO30, transfer will not start even if the CSIE30 bit value is set to "1".

Completion of an 8-bit transfer automatically stops the serial transfer operation and interrupt request flag (CSIIF30) is set.

CHAPTER 16 SERIAL INTERFACE (SIO31)

16.1 Serial Interface (SIO31) Functions

The serial interface (SIO31) has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. For details, see **16.4.1 Operation stop mode**.

(2) 3-wire serial I/O mode (fixed as MSB first)

This is an 8-bit data transfer mode using three lines: a serial clock line ($\overline{\text{SCK31}}$), serial output line (SO31), and serial input line (SI31).

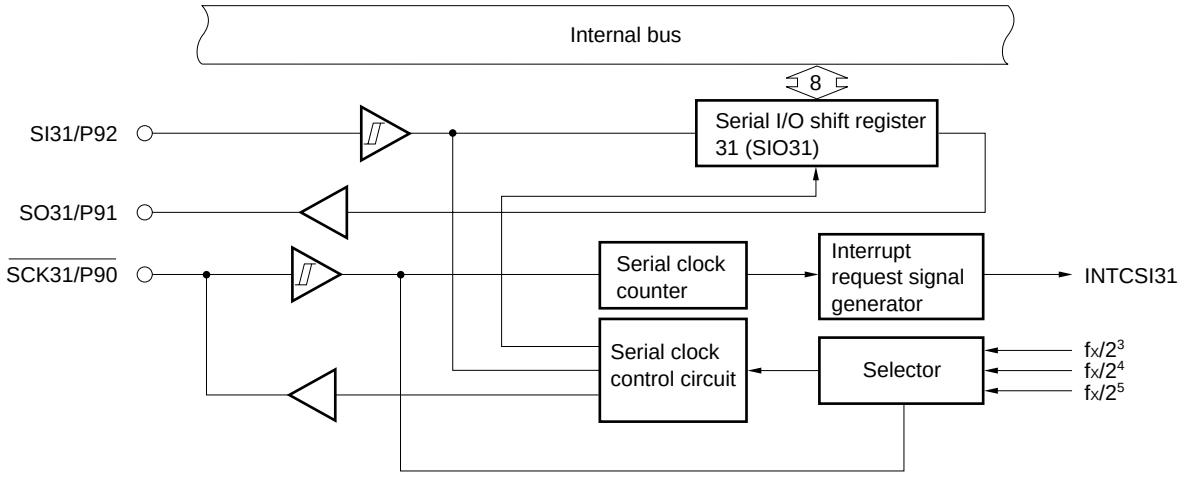
Since simultaneous transmit and receive operations are enabled in 3-wire serial I/O mode, the processing time for data transfers is reduced.

The first bit of the serial transferred 8-bit data is fixed as the MSB.

3-wire serial I/O mode is useful for connection to a peripheral I/O incorporating a clock-synchronous serial interface, or a display controller, etc. For details see **16.4.2 3-wire serial I/O mode**.

Figure 16-1 shows a block diagram of the serial interface (SIO31).

★ **Figure 16-1. Serial Interface (SIO31) Block Diagram**



16.2 Serial Interface (SIO31) Configuration

The serial interface (SIO31) includes the following hardware.

Table 16-1. Serial Interface (SIO31) Configuration

Item	Configuration
Registers	Serial I/O shift register 31 (SIO31)
Control registers	Serial operation mode register 31 (CSIM31)

(1) Serial I/O shift register 31 (SIO31)

This is an 8-bit register that performs parallel-serial conversion and serial transmit/receive (shift operations) synchronized with the serial clock.

SIO31 is set by an 8-bit memory manipulation instruction.

When “1” is set to bit 7 (CSIE31) of the serial operation mode register 31 (CSIM31), a serial operation can be started by writing data to or reading data from SIO31.

When transmitting, data written to SIO31 is output to the serial output (SO31).

When receiving, data is read from the serial input (SI31) and written to SIO31.

$\overline{\text{RESET}}$ input resets SIO31 to 00H.

Caution Do not access SIO31 during a transmit operation unless the access is triggered by a transfer start. (Read operation is disabled when MODE0 = 0 and write operation is disabled when MODE0 = 1.)

16.3 Register to Control Serial Interface (SIO31)

The serial interface (SIO31) uses the following type of register to control functions.

- Serial operation mode register 31 (CSIM31)

(1) Serial operation mode register 31 (CSIM31)

This register is used to enable or disable SIO31's serial clock, operation modes, and specific operations.

CSIM31 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM31 to 00H.

Caution In 3-wire serial I/O mode, set the port mode register (PMXX) as follows. In each case, set the output latch to 0.

- **During serial clock output (master transmission or master reception)**
  Set P90 ($\overline{\text{SCK31}}$) to output mode (PM90 = 0)
- **During serial clock input (slave transmission or slave reception)**
  Set P90 to input mode (PM90 = 1)
- **During transmit/transmit and receive mode**
  Set P91 (SO31) to output mode (PM91 = 0)
- **During receive mode**
  Set P92 (SI31) to input mode (PM92 = 1)

Figure 16-2. Format of Serial Operation Mode Register 31 (CSIM31)

Address: FFB1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM31	CSIE31	0	0	0	0	MODE1	SCL311	SCL310

CSIE31	Enable/disable specification for SIO31		
	Shift register operation	Serial counter	Port
0	Operation stop	Clear	Port function[Note 1]
1	Operation enable	Count operation enable	Serial function + port function[Note 2]

MODE1	Transfer operation modes and flags		
	Operation mode	Transfer start trigger	SO31 output
0	Transmit/transmit and receive mode	Write to SIO31	Normal output
1	Receive-only mode	Read from SIO31	Fixed at low level

SCL311	SCL310	Clock selection
0	0	External clock input to $\overline{\text{SCK31}}$
0	1	$f_X/2^3$ (1.05 MHz)
1	0	$f_X/2^4$ (524 kHz)
1	1	$f_X/2^5$ (262 kHz)

Notes 1. When CSIE31 = 0 (SIO31 operation stop status), the pins SI31, SO31, and $\overline{SCK31}$ can be used for port functions.

2. When CSIE31 = 1 (SIO31 operation enabled state), the SI31 pin can be used as a port pin if only the send function is used, and the SO31 pin can be used as a port pin if only the receive-only mode is used.

Remarks 1. f_X: main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz.

16.4 Serial Interface (SIO31) Operations

This section explains on two modes of serial interface SIO31.

16.4.1 Operation stop mode

Because the serial transfer is not performed during this mode, the power consumption can be reduced.
In addition, pins can be used as normal I/O ports.

(1) Register settings

Operation stop mode are set by the serial operation mode register 31 (CSIM31).
CSIM31 can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{RESET}$ input sets CSIM0 to 00H.

Address: FFB1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM31	CSIE31	0	0	0	0	MODE1	SCL311	SCL310

CSIE31	SIO31 operation enable/disable specification		
	Shift register operation	Serial counter	Port
0	Operation stop	Clear	Port function**Note 1**
1	Operation enable	Count operation enable	Serial function + port function**Note 2**

Notes 1. When CSIE31 = 0 (SIO31 operation stop status), the pins SI31, SO31, and $\overline{SCK31}$ can be used for port functions.

2. When CSIE31 = 1 (SIO31 operation enabled state), the SI31 pin can be used as a port pin if only the send function is used, and the SO31 pin can be used as a port pin if only the receive-only mode is used.

16.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connection to a peripheral I/O incorporating a clock-synchronous serial interface, a display controller, etc.

This mode executes data transfers via three lines: a serial clock line ($\overline{SCK31}$), serial output line (SO31), and serial input line (SI31).

(1) Register settings

3-wire serial I/O mode is set by the serial operation mode register 31 (CSIM31).

CSIM31 can be set by a 1-bit or 8-bit memory manipulation instructions.

RESET input sets CSIM31 to 00H .

Caution In 3-wire serial I/O mode, set the port mode register (PMXX) as follows. In each case, set he output latch to 0.

- **During serial clock output (master transmission or master reception)**
  Set P90 ($\overline{SCK31}$) to output mode (PM90 = 0)
- **During serial clock input (slave transmission or slave reception)**
  Set P90 to input mode (PM90 = 1)
- **During transmit/transmit and receive mode**
  Set P91 (SO31) to output mode (PM91 = 0)
- **During receive mode**
  Set P92 (SI31) to input mode (PM92 = 1)

Address: FFB1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM31	CSIE31	0	0	0	0	MODE1	SCL311	SCL310

CSIE31	Enable/disable specification for SIO31		
	Shift register operation	Serial counter	Port
0	Operation stop	Clear	Port function[Note 1]
1	Operation enable	Count operation enable	Serial function + port function[Note 2]

MODE1	Transfer operation modes and flags		
	Operation mode	Transfer start trigger	SO31 output
0	Transmit/transmit and receive mode	Write to SIO31	Normal output
1	Receive-only mode	Read from SIO31	Fixed at low level

SCL311	SCL310	Clock selection
0	0	External clock input to $\overline{SCK31}$
0	1	$f_X/2^3$ (1.05 MHz)
1	0	$f_X/2^4$ (524 kHz)
1	1	$f_X/2^5$ (262 kHz)

Notes 1. When CSIE31 = 0 (SIO31 operation stop status), the pins SI31, SO31, and $\overline{\text{SCK31}}$ can be used for port functions.

2. When CSIE31 = 1 (SIO31 operation enabled state), the SI31 pin can be used as a port pin if only the send function is used, and the SO31 pin can be used as a port pin if only the receive-only mode is used.

Remarks 1. f_X: main system clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 8.38 MHz.

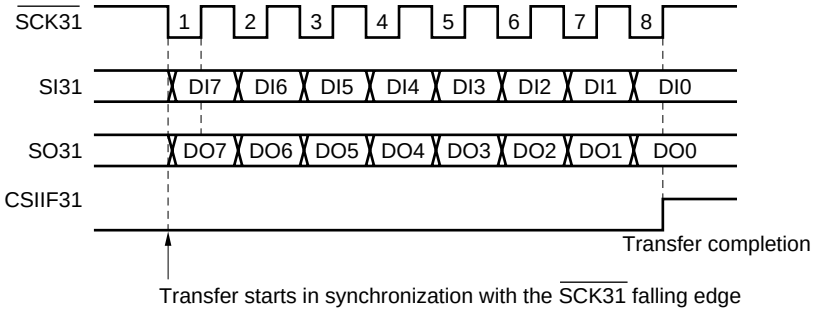
(2) Communication operations

In the three-wire serial I/O mode, data is transmitted and received in 8-bit units. Each bit of data is sent or received in synchronization with the serial clock.

The serial I/O shift register 31 (SIO31) is shifted in synchronization with the falling edge of the serial clock. Transmission data is held in the SO31 latch and is output from the SO31 pin. Data that is received via the SI31 pin in synchronization with the rising edge of the serial clock is latched to SIO31.

Completion of an 8-bit transfer automatically stops operation of SIO31 and sets interrupt request flag (CSIIF31).

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



(3) Transfer start

A serial transfer starts when the following two conditions have been satisfied and transfer data has been set (or read) to serial I/O shift register 31 (SIO31).

- The SIO31 operation control bit (CSIE31) = 1
- After an 8-bit serial transfer, either the internal serial clock is stopped or $\overline{\text{SCK31}}$ is set to high level.
- Transmit/transmit and receive mode
  When CSIE31 = 1 and MODE1 = 0, transfer starts when writing to SIO31.
- Receive-only mode
  When CSIE31 = 1 and MODE1 = 1, transfer starts when reading from SIO31.

Caution After data has been written to SIO31, transfer will not start even if the CSIE31 bit value is set to "1".

Completion of an 8-bit transfer automatically stops the serial transfer operation and interrupt request flag (CSIIF31) is set.

CHAPTER 17 INTERRUPT FUNCTIONS

17.1 Interrupt Function Types

The following three types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally. It does not undergo priority control and is given top priority over all other interrupt requests.

It generates a standby release signal.

One interrupt from the watchdog timer is incorporated as a non-maskable interrupt.

(2) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into a high interrupt priority group and a low interrupt priority group by setting the priority specify flag registers (PR0L, PR0H, PR1L).

Multiple high priority interrupts can be applied to low priority interrupts. If two or more interrupts with the same priority are simultaneously generated, each interrupt has a predetermined priority (see **Table 17-1**).

A standby release signal is generated.

Four external interrupt requests and fourteen internal interrupt requests are incorporated as maskable interrupts.

(3) Software interrupt

This is a vectored interrupt to be generated by executing the BRK instruction. It is acknowledged even in a disabled stated. The software interrupt does not undergo interrupt priority control.

17.2 Interrupt Sources and Configuration

A total of twenty interrupt sources exist among non-maskable, maskable, and software interrupts (see **Table 17-1**).

Table 17-1. Interrupt Source List

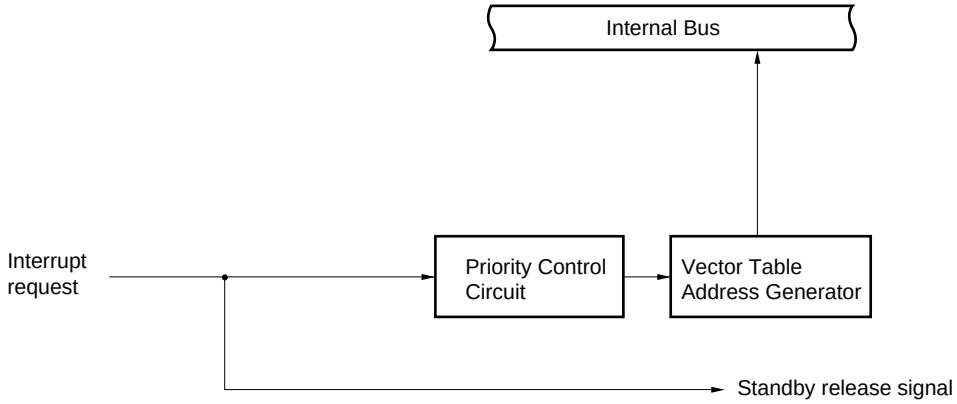
Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Non-maskable	—	INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (with interval timer mode selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTSER0	Serial interface UART0 reception error generation	Internal	000EH	(B)
	6	INTSR0	End of serial interface UART0 reception		0010H	
	7	INTST0	End of serial interface UART0 transmission		0012H	
	8	INTCSI30	End of serial interface SIO30 transfer		0014H	
	9	INTCSI31	End of serial interface SIO31 transfer		0016H	
	10	INTCSI1	End of serial interface SIO1 transfer		0018H	
★	11	INTTM00	Match of 16-bit timer/counter 0 and capture/compare register 00 (CR00) (with CR00 specified to compare register) TI00 valid edge detection (with CR01 specified to capture register)		001AH	
★	12	INTTM01	Match of 16-bit timer/counter 0 and capture/compare register 01 (CR01) (with CR01 specified to compare register) TI01 valid edge detection (with CR00 specified to capture register)		001CH	
	13	INTTM50	Match of 8-bit counter 50 and 8-bit compare register 50		001EH	
	14	INTTM51	Match of 8-bit counter 51 and 8-bit compare register 51		0020H	
	15	INTWTI	Reference time interval signal from watch timer		0022H	
	16	INTWT	Watch timer overflow		0024H	
	17	INTAD0	End of A/D converter conversion		0026H	
Software	—	BRK	BRK instruction execution	—	003EH	(D)

Notes 1. The default priority is the priority applicable when two or more maskable interrupt are generated simultaneously. 0 is the highest priority, and 17 is the lowest.

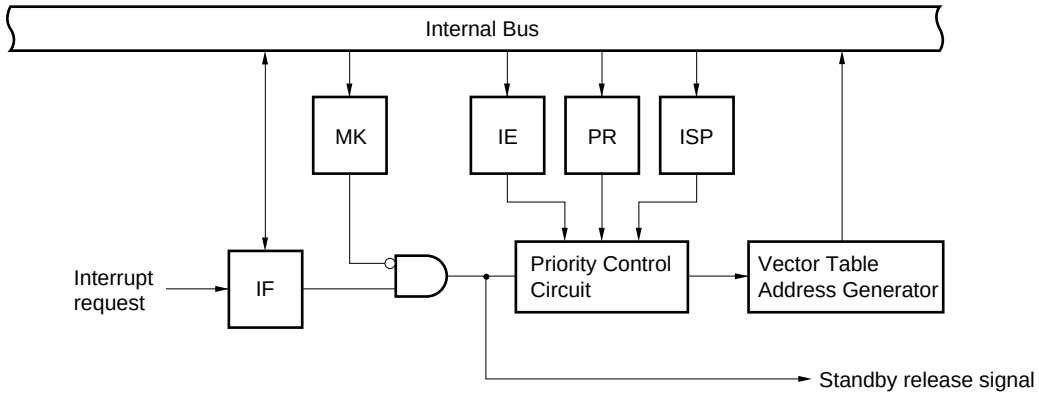
2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 17-1.

Figure 17-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt (INTP0 to INTP3)

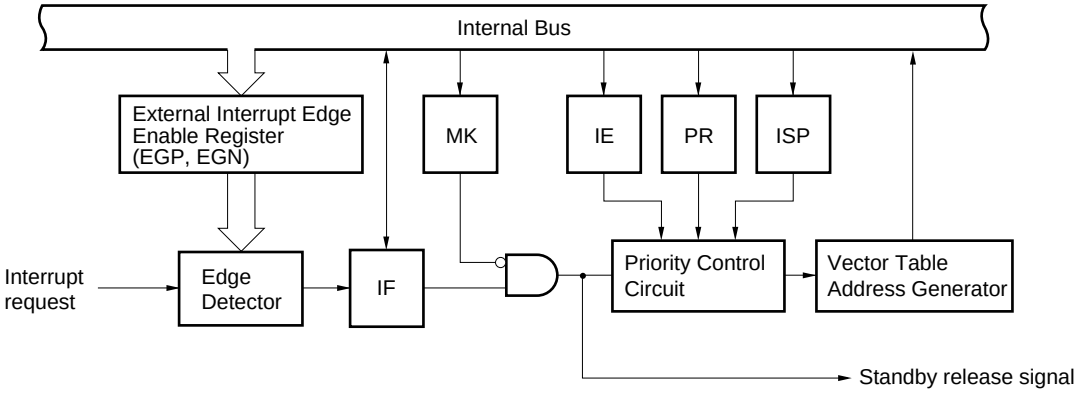
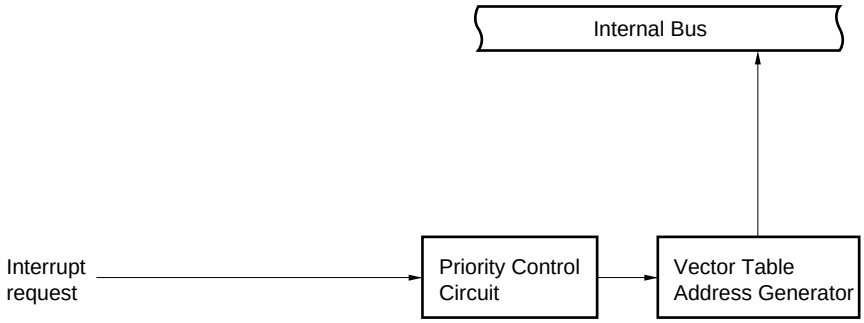


Figure 17-1. Basic Configuration of Interrupt Function (2/2)

(D) Software interrupt



17.3 Interrupt Function Control Registers

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H, IF1L)
- Interrupt mask flag register (MK0L, MK0H, MK1L)
- Priority specify flag register (PR0L, PR0H, PR1L)
- External interrupt rising edge enable flag (EGP)
- External interrupt falling edge enable flag (EGN)
- Program status word (PSW)

Table 17-2 gives a list of interrupt request flags, interrupt mask flags, and priority specify flags corresponding to interrupt request sources.

Table 17-2. Flags Corresponding to Interrupt Request Sources

Interrupt Request	Interrupt Request Flag		Interrupt Mask Flag		Priority Specify Flag	
		Register		Register		Register
INTWDT	WDTIF	IF0L	WDTMK	MK0L	WDTPR	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTSER0	SERIF0		SERMK0		SERPR0	
INTSR0	SRIF0		SRMK0		SRPR0	
INTST0	STIF0		STMK0		STPR0	
INTCSI30	CSIIF30	IF0H	CSIMK30	MK0H	CSIPR30	PR0H
INTCSI31	CSIIF31		CSIMK31		CSIPR31	
INTCSI1	CSIIF1		CSIMK1		CSIPR1	
INTTM00	TMIF00		TMMK00		TMPR00	
INTTM01	TMIF01		TMMK01		TMPR01	
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM51	TMIF51		TMMK51		TMPR51	
INTWTI	WTIIF		WTIMK		WTIPR	
★ INTWT	WTIF	IF1L	WTMK	MK1L	WTPR0	PR1L
INTAD0	ADIF0		ADMK0		ADPR0	

(1) Interrupt request flag registers (IF0L, IF0H, IF1L)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon application of $\overline{\text{RESET}}$ input.

IF0L, IF0H, and IF1L are set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H are combined to form 16-bit register IF0, they are read with a 16-bit memory manipulation instruction.

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

Figure 17-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L)

Address: FFE0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF0L	STIF0	SRIF0	SERIF0	PIF3	PIF2	PIF1	PIF0	WDTIF

Address: FFE1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF0H	WTIIF	TMIF51	TMIF50	TMIF01	TMIF00	CSIIF1	CSIIF31	CSIIF30

Address: FFE2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF1L	0	0	0	0	0	0	ADIF0	WTIF

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Cautions 1. The WDTIF flag is R/W enabled only when the watchdog timer is used as the interval timer. If watchdog timer mode 1 is used, set the WDTIF flag to 0.

2. Be sure to set 0 to IF1L bits 2 through 7.

3. When operating a timer, serial interface, or A/D converter after stand-by release, run it once after clearing an interrupt request flag. An interrupt request flag may be set by noise.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt service and to set standby clear enable/disable.

MK0L, MK0H, and MK1L are set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H are combined to form a 16-bit register, they are set with a 16-bit memory manipulation instruction.

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

Figure 17-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L)

Address: FFE4H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK0L	STMK0	SRMK0	SERMK0	PMK3	PMK2	PMK1	PMK0	WDTMK

Address: FFE5H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK0H	WTIMK	TMMK51	TMMK50	TMMK01	TMMK00	CSIMK1	CSIMK31	CSIMK30

Address: FFE6H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK1L	1	1	1	1	1	1	ADMK0	WTMK

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Cautions 1. If the watchdog timer is used in watchdog timer mode 1, the contents of the WDTMK flag become undefined when read.

2. Because port 0 pins have an alternate function as external interrupt request input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, 1 should be set in the interrupt mask flag before using the output mode.

3. Always set MK1L bits 2 through 7 to 1.

(3) Priority specify flag registers (PR0L, PR0H, PR1L)

The priority specify flag registers are used to set the corresponding maskable interrupt priority orders. PR0L, PR0H, and PR1L are set by a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H are combined to form 16-bit register PR0, they are set with a 16-bit memory manipulation instruction.

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

Figure 17-4. Format of Priority Specify Flag Registers (PR0L, PR0H, PR1L)

Address: FFE8H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR0L	STPR0	SRPR0	SERPR0	PPR3	PPR2	PPR1	PPR0	WDTPR

Address: FFE9H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR0H	WTIPR	TMPR51	TMPR50	TMPR01	TMPR00	CSIPR1	CSIPR31	CSIPR30

Address: FFEAH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
★ PR1L	1	1	1	1	1	1	ADPR0	WTPR0

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Cautions 1. When the watchdog timer is used in the watchdog timer 1 mode, set 1 in the WDTPR flag.

2. Always set PR1L bits 2 through 7 to 1.

(4) External interrupt rising edge enable register (EGP), External interrupt falling edge enable register (EGN)

These registers specify the valid edge for INTP0 to INTP3.

EGP and EGN are set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 17-5. Format of External Interrupt Rising Edge Enable Register (EGP), External Interrupt Falling Edge Enable Register (EGN)

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP	0	0	0	0	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	0	0	0	0	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 3)
0	0	Interrupt disable
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

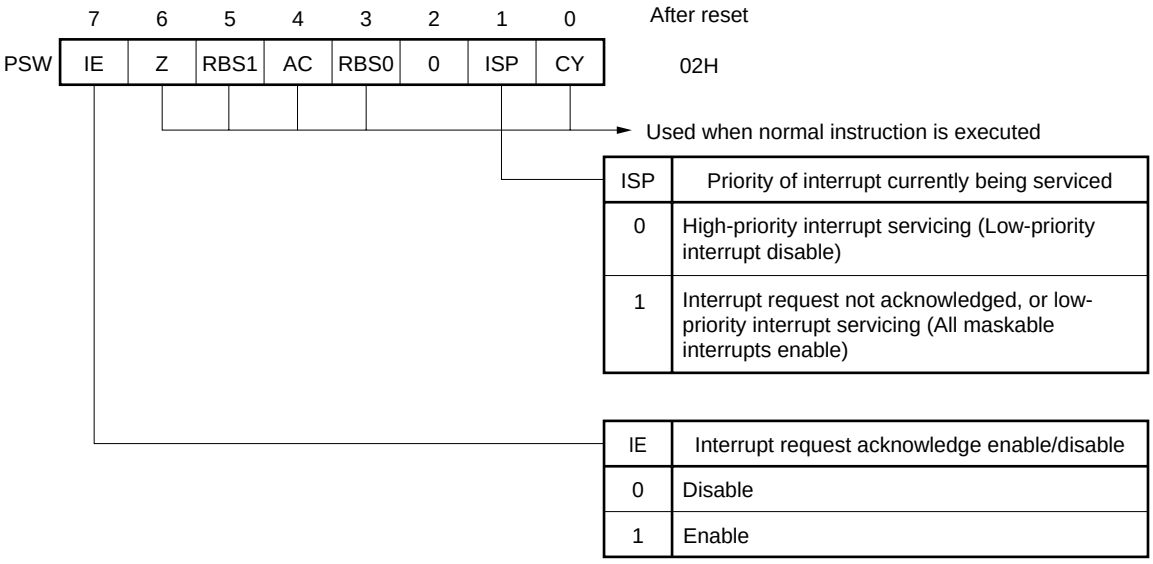
(5) Program status word (PSW)

The program status word is a register to hold the instruction execution result and the current status for an interrupt request. The IE flag to set maskable interrupt enable/disable and the ISP flag to control multiple processing are mapped.

Besides 8-bit read/write, this register can carry out operations with a bit manipulation instruction and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of PSW are automatically saved into a stack and the IE flag is reset to 0. If a maskable interrupt request is acknowledged, the contents of the priority specify flag of the acknowledged interrupt are transferred to the ISP flag. The PSW contents are also saved into the stack with 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 17-6. Program Status Word Format



ISP	Priority of interrupt currently being serviced
0	High-priority interrupt servicing (Low-priority interrupt disable)
1	Interrupt request not acknowledged, or low-priority interrupt servicing (All maskable interrupts enable)

IE	Interrupt request acknowledge enable/disable
0	Disable
1	Enable

17.4 Interrupt Servicing Operations

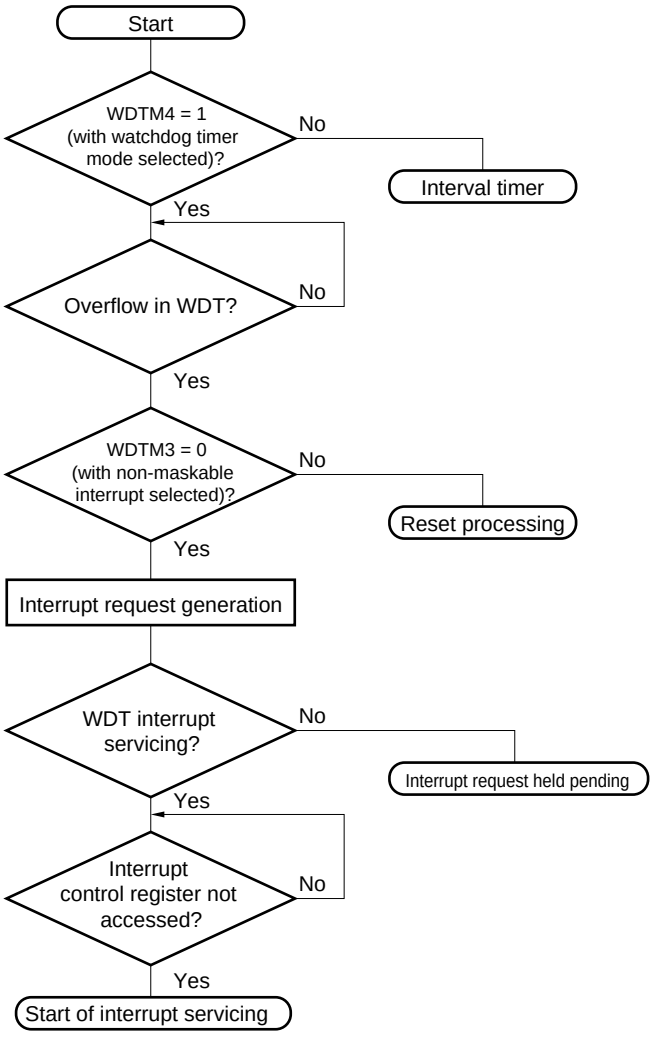
17.4.1 Non-maskable interrupt request acknowledge operation

A non-maskable interrupt request is unconditionally acknowledged even if in an interrupt acknowledge disable state. It does not undergo interrupt priority control and has highest priority over all other interrupts.

If a non-maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag and ISP flag are reset (0), and the contents of the vector table are loaded into PC and branched.

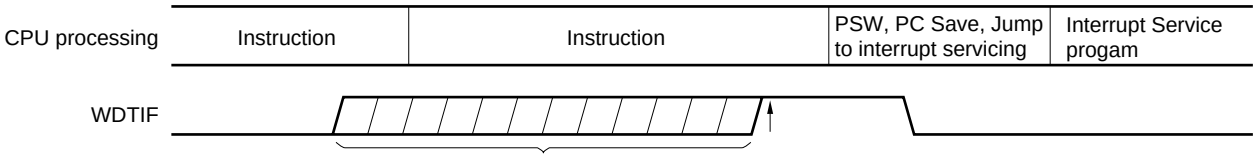
A new non-maskable interrupt request generated during execution of a non-maskable interrupt servicing program is acknowledged after the current execution of the non-maskable interrupt servicing program is terminated (following RETI instruction execution) and one main routine instruction is executed. However, if a new non-maskable interrupt request is generated twice or more during non-maskable interrupt servicing program execution, only one non-maskable interrupt request is acknowledged after termination of the non-maskable interrupt servicing program execution. Figures 17-7, 17-8, and 17-9 show the flowchart of the non-maskable interrupt request generation through acknowledge, acknowledge timing of non-maskable interrupt request, and acknowledge operation at multiple non-maskable interrupt request generation, respectively.

Figure 17-7. Non-Maskable Interrupt Request Generation to Acknowledge Flowchart



WDTM: Watchdog timer mode register
WDT: Watchdog timer

Figure 17-8. Non-Maskable Interrupt Request Acknowledge Timing

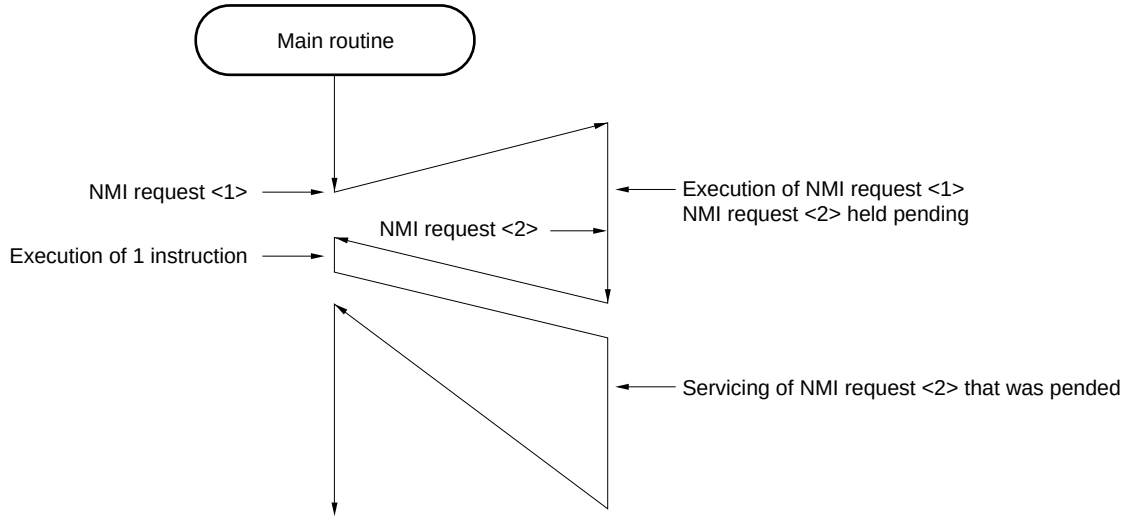


Interrupt request generated during this interval is acknowledged at ↑.

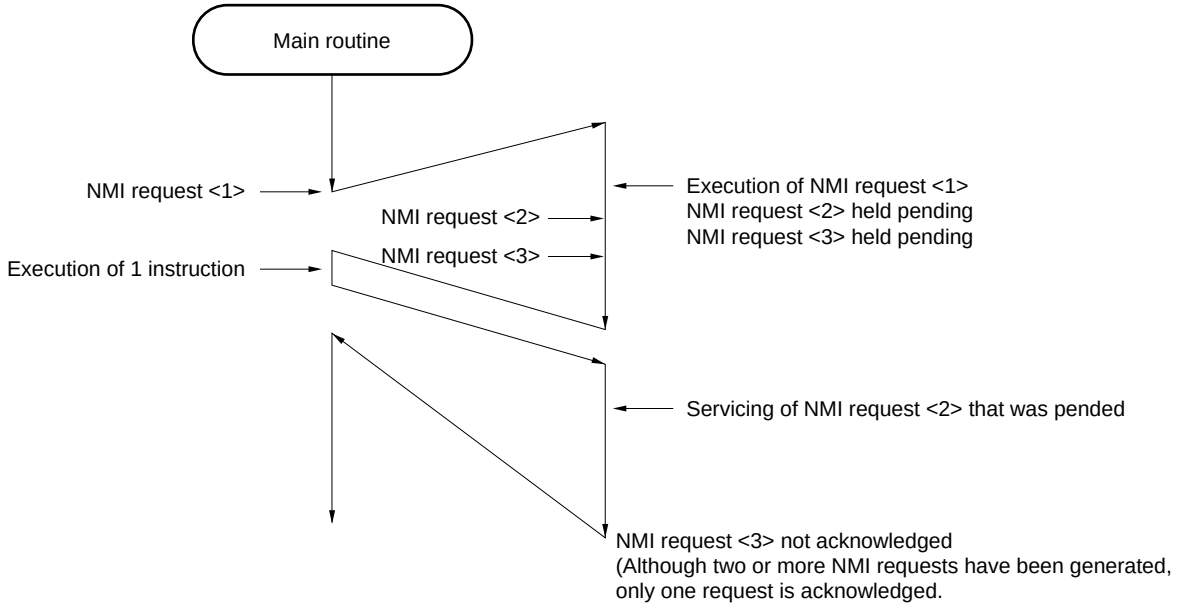
WDTIF: Watchdog timer interrupt request flag

Figure 17-9. Non-Maskable Interrupt Request Acknowledge Operation

(a) If a 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



17.4.2 Maskable interrupt acknowledge operation

A maskable interrupt becomes acknowledgeable when an interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if in the interrupt enable state (when IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request (when the ISP flag is reset to 0). The times from generation of a maskable interrupt request until interrupt servicing is performed are listed in Table 17-3 below.

For the interrupt request acknowledge timing, see Figures 17-11 and 17-12.

Table 17-3. Times from Generation of Maskable Interrupt until Servicing

	Minimum Time	Maximum Time[Note]
When ××PR = 0	7 clocks	32 clocks
When ××PR = 1	8 clocks	33 clocks

Note If an interrupt request is generated just before a divide instruction, the wait time becomes longer.

Remark 1 clock: $1/f_{CPU}$ (f_{CPU}: CPU clock)

If two or more interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specify flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

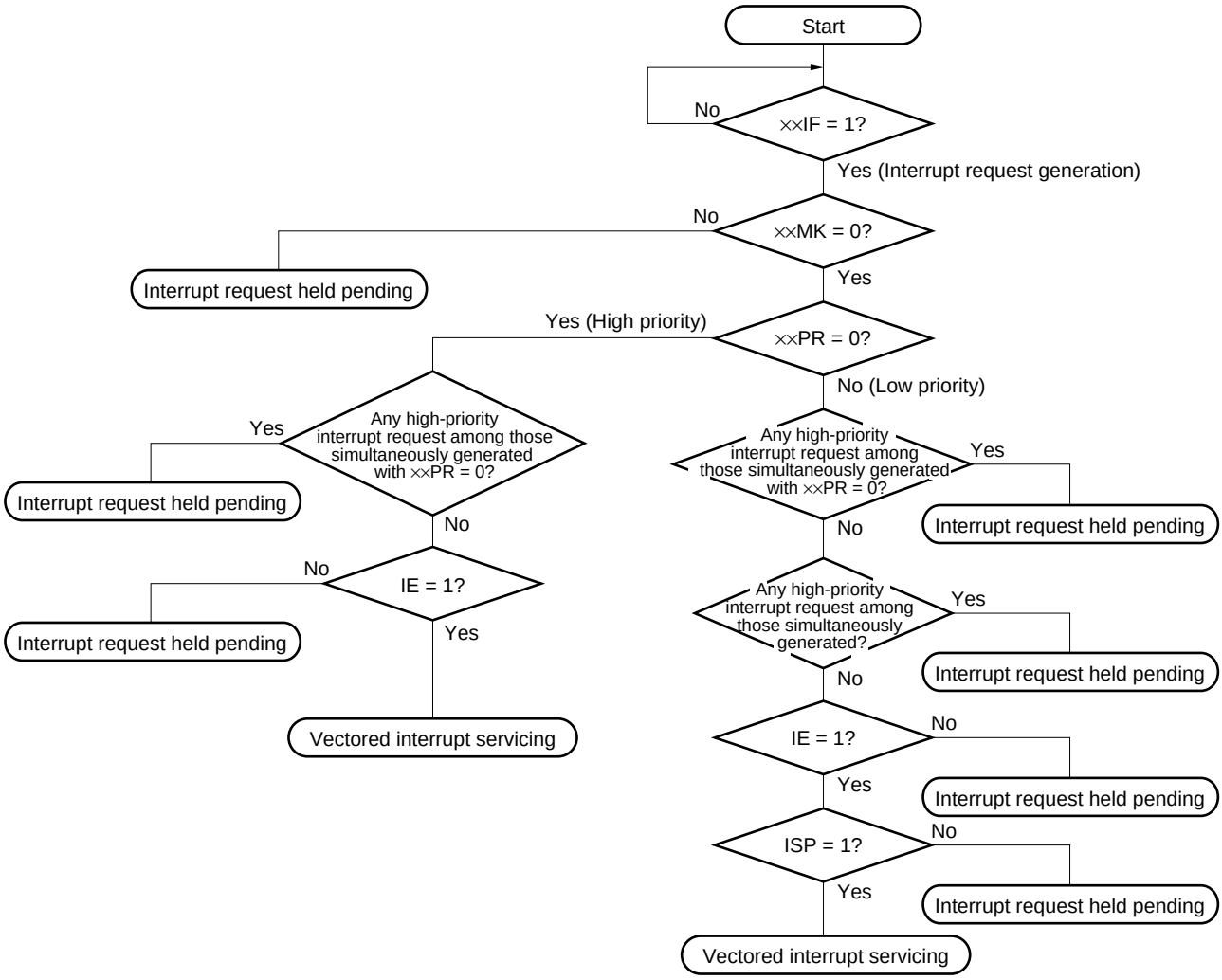
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 17-10 shows the interrupt request acknowledge algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specify flag corresponding to the acknowledged interrupt are transferred to the ISP flag. Further, the vector table data determined for each interrupt request is loaded into PC and branched.

Return from an interrupt is possible with the RETI instruction.

Figure 17-10. Interrupt Request Acknowledge Processing Algorithm



××IF: Interrupt request flag

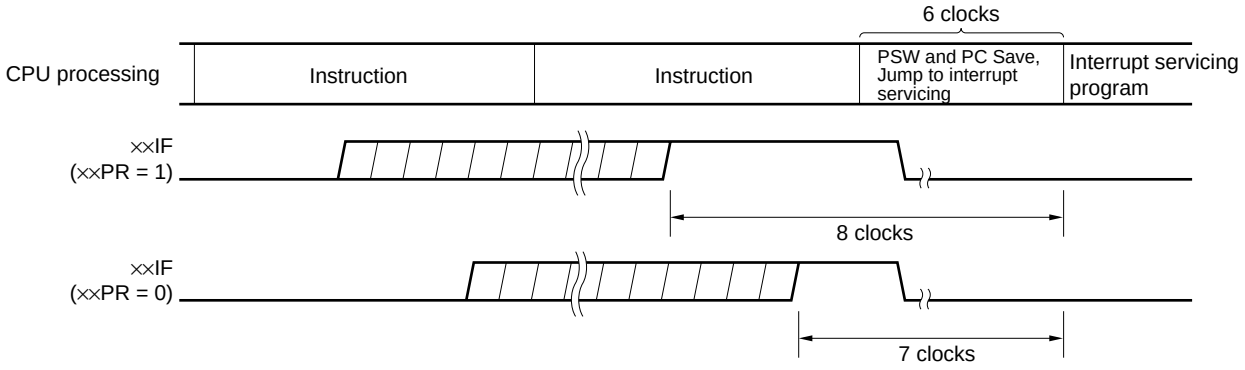
××MK: Interrupt mask flag

××PR: Priority specify flag

IE: Flag that controls acknowledge of maskable interrupt request (1 = Enable, 0 = Disable)

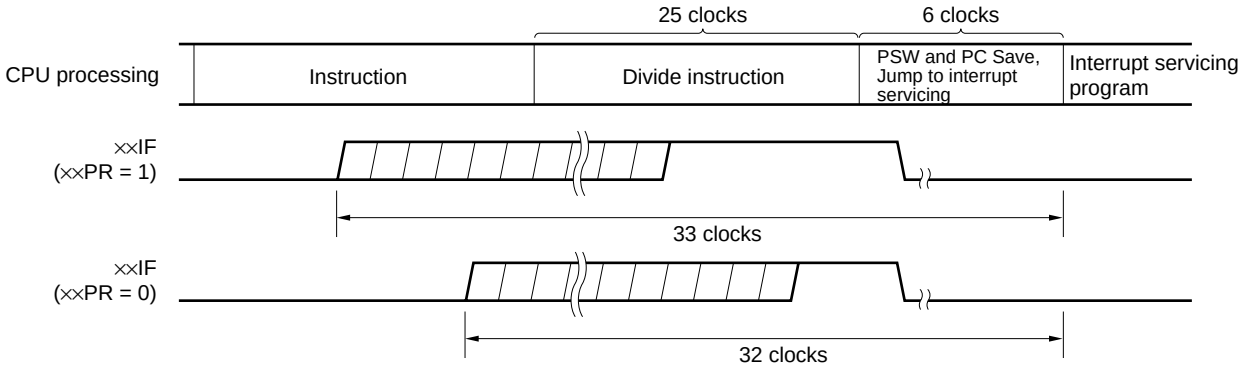
ISP: Flag that indicates the priority level of the interrupt currently being serviced (0 = high-priority interrupt servicing, 1 = No interrupt request received, or low-priority interrupt servicing)

Figure 17-11. Interrupt Request Acknowledge Timing (Minimum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

Figure 17-12. Interrupt Request Acknowledge Timing (Maximum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

17.4.3 Software interrupt request acknowledge operation

A software interrupt acknowledge is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (003EH, 003FH) are loaded into PC and branched.

Return from a software interrupt is possible with the RETB instruction.

Caution Do not use the RETI instruction for returning from the software interrupt.

17.4.4 Multiple interrupt servicing

Multiple interrupts occur when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupts do not occur unless the interrupt request acknowledge enable state is selected (IE = 1) (except non-maskable interrupts). Also, when an interrupt request is received, interrupt requests acknowledge becomes disabled (IE = 0). Therefore, to enable multiple interrupts, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledge.

Moreover, even if interrupts are enabled, multiple interrupts 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 interrupts.

In the interrupt enable 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 of the interrupt disable state or they have a lower priority are held pending. When servicing of the current interrupt ends, the pendent interrupt request is acknowledged following execution of one main processing instruction execution.

Multiple interrupt servicing is not possible during non-maskable interrupt servicing.

Table 17-4 shows interrupt requests enabled for multiple interrupt servicing, and Figure 17-13 shows multiple interrupt examples.

Table 17-4. Interrupt Request Enabled for Multiple Interrupt during Interrupt Servicing

Multiple Interrupt Request / Interrupt being Serviced		Non-Maskable Interrupt Request	Maskable Interrupt Request			
			PR = 0		PR = 1	
			IE = 1	IE = 0	IE = 1	IE = 0
Non-maskable interrupt		×	×	×	×	×
Maskable interrupt	ISP = 0	○	○	×	×	×
	ISP = 1	○	○	×	○	×
Software interrupt		○	○	×	○	×

Remarks 1. ○: Multiple interrupt enable

2. ×: Multiple interrupt disable

3. ISP and IE are flags contained in PSW.

ISP = 0: An interrupt with higher priority is being serviced.

ISP = 1: No interrupt request has been acknowledged, or an interrupt with a lower priority is being serviced.

IE = 0: Interrupt request acknowledge is disabled.

IE = 1: Interrupt request acknowledge is enabled.

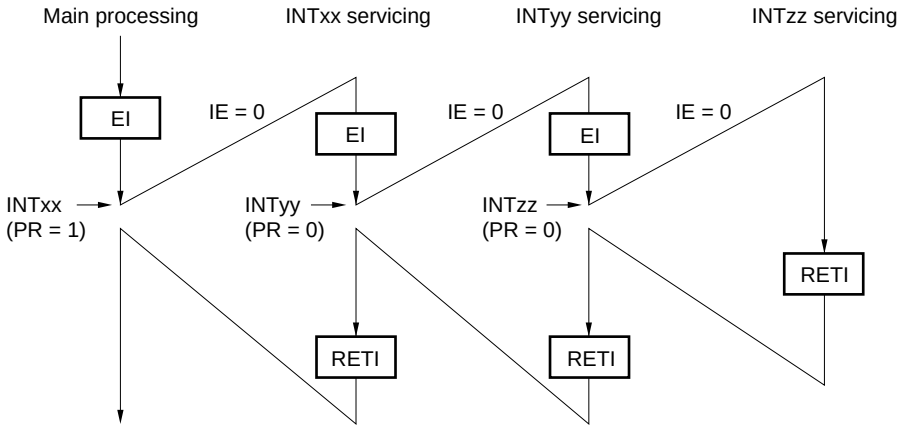
4. PR is a flag contained in PR0L, PR0H, and PR1L.

PR = 0: Higher priority level

PR = 1: Lower priority level

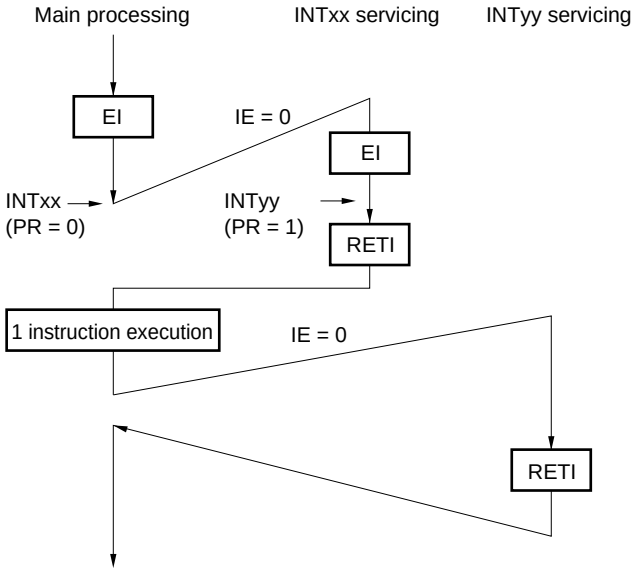
Figure 17-13. Multiple Interrupt Examples (1/2)

Example 1. Multiple interrupts occur twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledge.

Example 2. Multiple interrupt servicing does not occur due to priority control

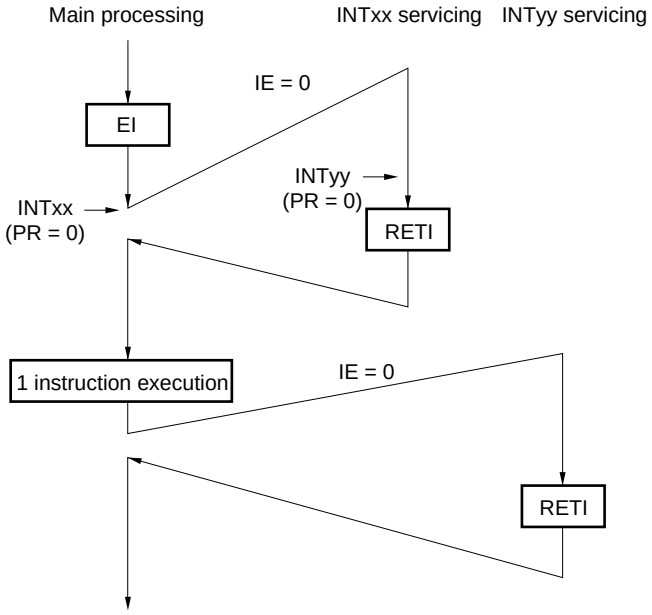


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
PR = 1: Lower priority level
IE = 0: Interrupt request acknowledge disable

Figure 17-13. Multiple Interrupt Examples (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupt is not enabled



Interrupt is not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
IE = 0: Interrupt request acknowledge disabled

17.4.5 Interrupt request hold

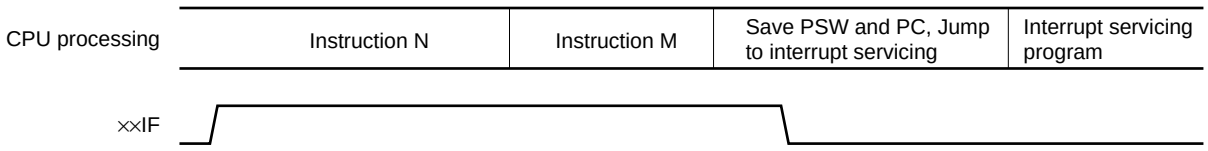
There are instructions where, even if an interrupt request is issued for them while another instruction is executed, request acknowledge is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- 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
- Manipulate instructions for the IF0L, IF0H, IF1L, MK0L, MK0H, MK1L, PR0L, PR0H, PR1L, EGP, and EGN registers.

Caution The BRK instruction is not one of the above-listed interrupt request hold instructions. However, the software interrupt activated by executing the BRK instruction causes the IE flag to be cleared. Therefore, even if a maskable interrupt requests is generated during execution of the BRK instruction, the interrupt request is not acknowledged. However, a non-maskable interrupt request is acknowledged.

Figure 17-14 shows the timing with which interrupt requests are held pending.

Figure 17-14. 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 (instruction request)

CHAPTER 18 EXTERNAL DEVICE EXPANSION FUNCTION

18.1 External Device Expansion Function

The external device expansion function connects external devices to areas other than the internal ROM, RAM, and SFR. Connection of external devices uses ports 4 to 6. Ports 4 to 6 control address/data, read/write strobe, wait, address strobe, etc.

Table 18-1. Pin Functions in External Memory Expansion Mode

Pin Function at External Device Connection		Alternate Function
Name	Function	
AD0 to AD7	Multiplexed address/data bus	P40 to P47
A8 to A15	Address bus	P50 to P57
$\overline{RD}$	Read strobe signal	P64
$\overline{WR}$	Write strobe signal	P65
$\overline{WAIT}$	Wait signal	P66
ASTB	Address strobe signal	P67

Table 18-2. State of Port 4 to 6 Pins in External Memory Expansion Mode

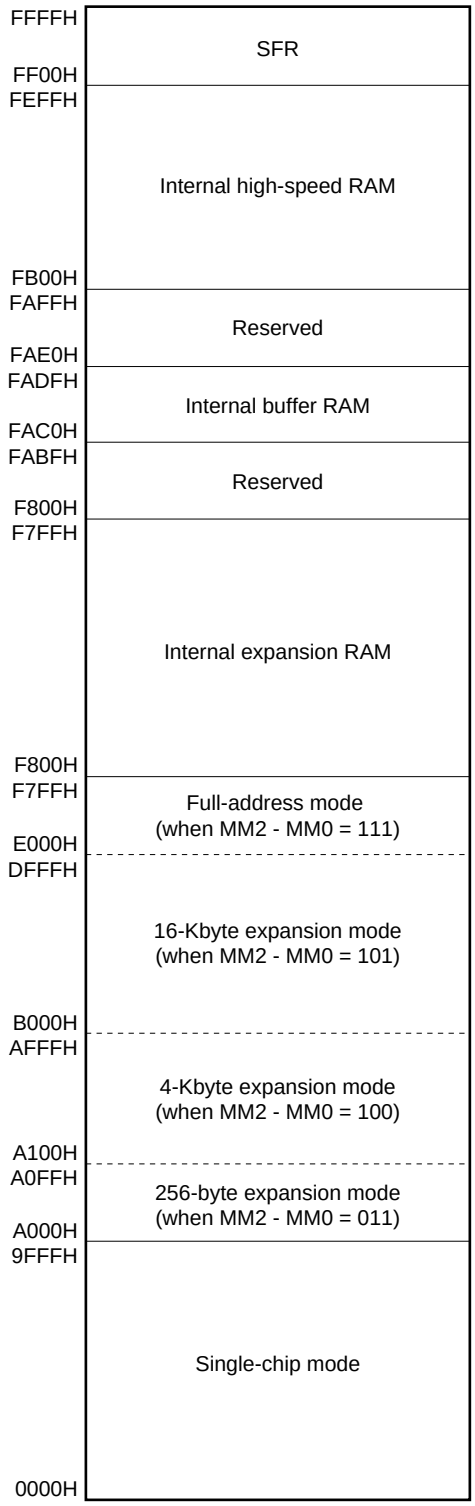
Port / External expansion mode	Port 4	Port 5								Port 6							
	0 to 7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Single-chip mode	Port	Port								Port							
256-byte expansion mode	Address/data	Port								Port				$\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB			
4-Kbyte expansion mode	Address/data	Address				Port				Port				$\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB			
16-Kbyte expansion mode	Address/data	Address						Port		Port				$\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB			
Full-address mode	Address/data	Address								Port				$\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB			

Caution When the external wait function is not used, the $\overline{WAIT}$ pin can be used as a port in all modes.

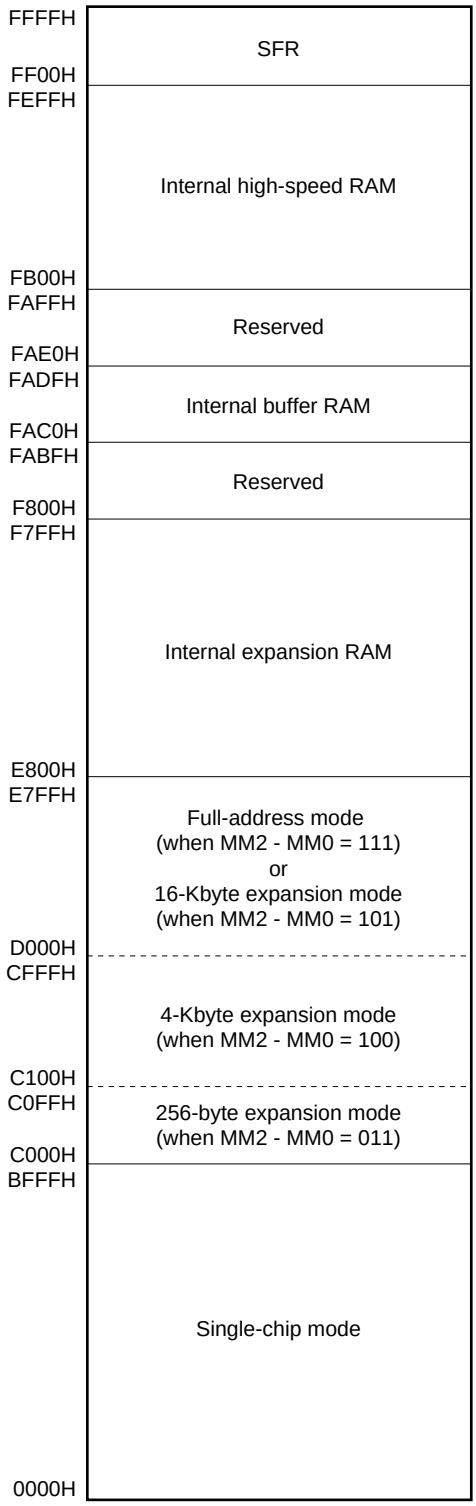
The memory maps when using the external device expansion function are as follows.

Figure 18-1. Memory Map When Using External Device Function

(a) Memory map of μPD780065 and μPD78F0066 when internal ROM (flash memory) size is 40 Kbytes



(b) Memory map of μPD78F0066



18.2 External Device Expansion Function Control Register

The external device expansion function is controlled by the following two types of registers.

- Memory expansion mode register (MEM)
- Memory expansion wait setting register (MM)

(1) Memory expansion mode register (MEM)

MEM sets the external expansion area.

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

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

Figure 18-2. Format of Memory Expansion Mode Register (MEM)

Address: FF47H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MEM	0	0	0	0	0	MM2	MM1	MM0

MM2	MM1	MM0	Single-chip/memory expansion mode selection		P40 to P47, P50 to P57, P64 to P67 pin state				
					P40 to P47	P50 to P53	P54, P55	P56, P57	P64 to P67
0	0	0	Single-chip mode		Port mode				
0	0	1							
0	1	1	Memory expansion mode	256-byte mode	AD0 to AD7	Port mode			P64 = $\overline{\text{RD}}$ P65 = $\overline{\text{WR}}$
1	0	0		4-Kbyte mode		A8 to A11	Port mode		P66 = $\overline{\text{WAIT}}$ P67 = ASTB
1	0	1		16-Kbyte mode			A12, A13	Port mode	
1	1	1		Full-address mode**Note**				A14, A15	
Other than above			Setting prohibited						

Note The full-address mode allows external expansion to the entire 64-Kbyte address space except for the internal ROM, RAM, SFR areas and the reserved areas.

(2) Memory expansion wait setting register (MM)

MM sets the number of waits.

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

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

Figure 18-3. Format of Memory Expansion Wait Setting Register (MM)

Address: FFF8H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
MM	0	0	PW1	PW0	0	0	0	0

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

18.3 External Device Expansion Function Timing

Timing control signal output pins in the external memory expansion mode are as follows.

(1) $\overline{RD}$ pin (Alternate function: P64)

Read strobe output pin. The read strobe output pin is output in data accesses and instruction fetches from external memory.

During internal memory access, the read strobe signal is not output (maintains high level).

(2) $\overline{WR}$ pin (Alternate function: P65)

Write strobe signal output pin. The write strobe signal is output in data access to external memory.

During internal memory access, the write strobe signal is not output (maintains high level)

(3) $\overline{WAIT}$ pin (Alternate function: P66)

External wait signal input pin.

When the external wait is not used, the $\overline{WAIT}$ pin can be used as an input/output port.

During internal memory access, the external wait signal is ignored.

(4) ASTB pin (Alternate function: P67)

Address strobe signal output pin. The address strobe signal is output regardless of data access and instruction fetch from external memory. During internal memory access, the address strobe signal is not output.

(5) AD0 to AD7, A8 to A15 pins (Alternate function: P40 to P47, P50 to P57)

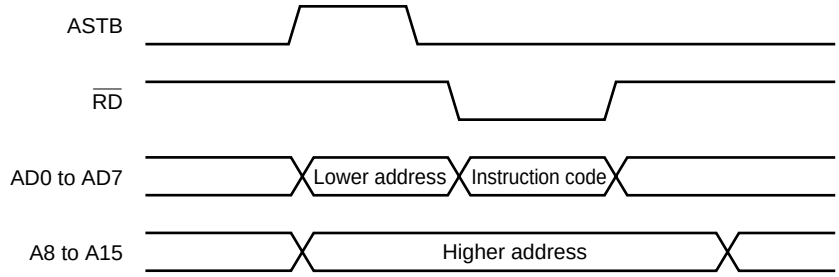
Address/data signal output pins. Valid signal is output or input during data accesses and instruction fetches from external memory.

These signals change even during internal memory access (output values are undefined).

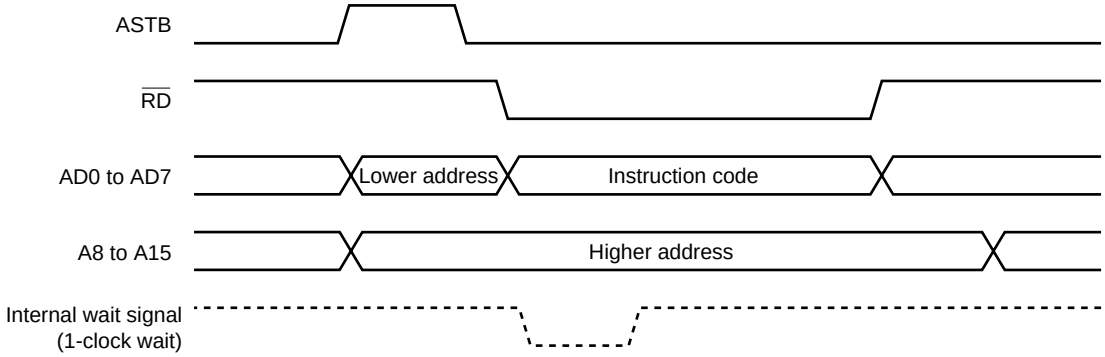
The timing charts are shown in Figures 18-4 to 18-7.

Figure 18-4. Instruction Fetch from External Memory

(a) No wait (PW1, PW0 = 0, 0) setting



(b) Wait (PW1, PW0 = 0, 1) setting



(c) External wait (PW1, PW0 = 1, 1) setting

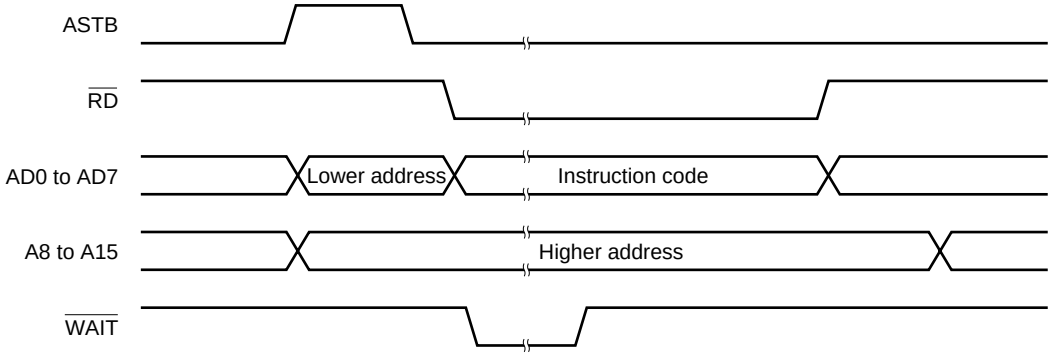
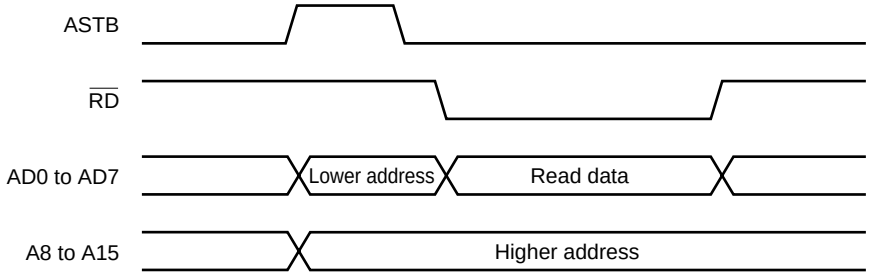
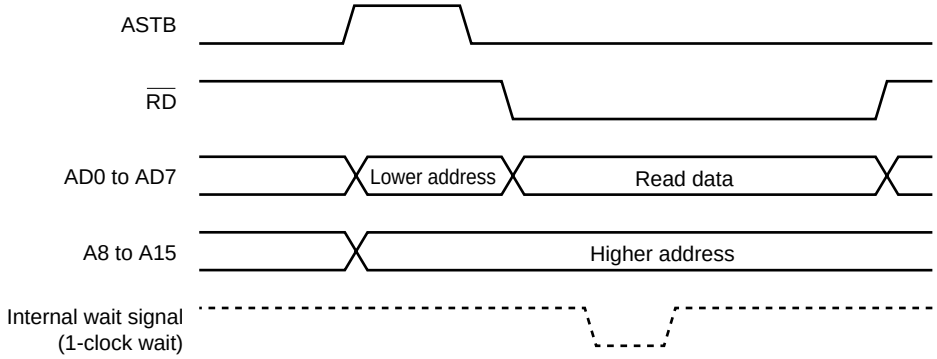


Figure 18-5. External Memory Read Timing

(a) No wait (PW1, PW0 = 0, 0) setting



(b) Wait (PW1, PW0 = 0, 1) setting



(c) External wait (PW1, PW0 = 1, 1) setting

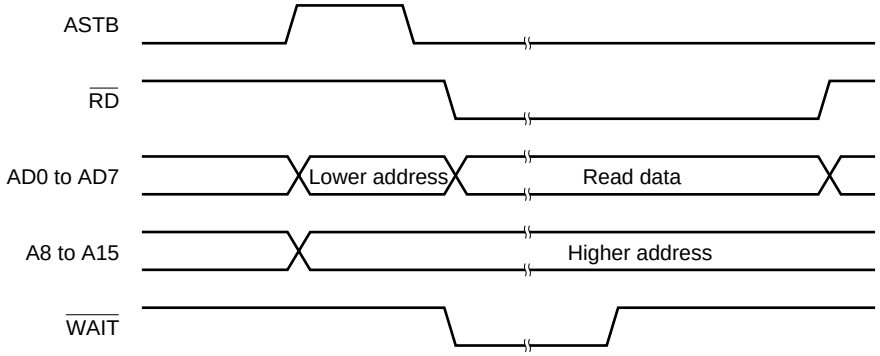
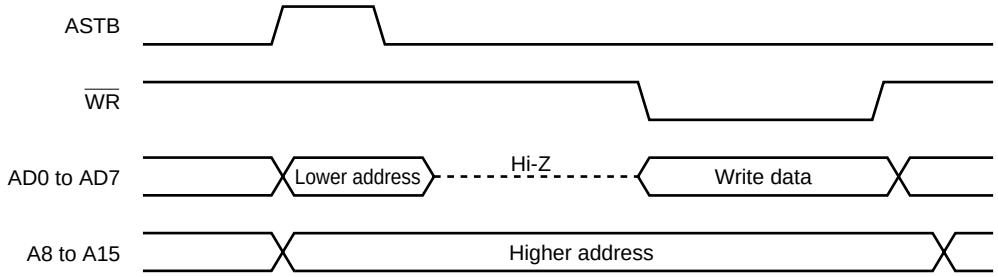
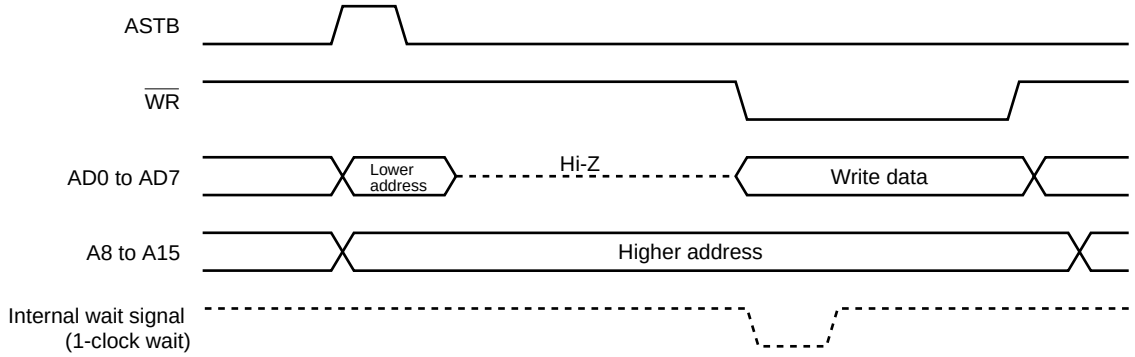


Figure 18-6. External Memory Write Timing

(a) No wait (PW1, PW0 = 0, 0) setting



(b) Wait (PW1, PW0 = 0, 1) setting



(c) External wait (PW1, PW0 = 1, 1) setting

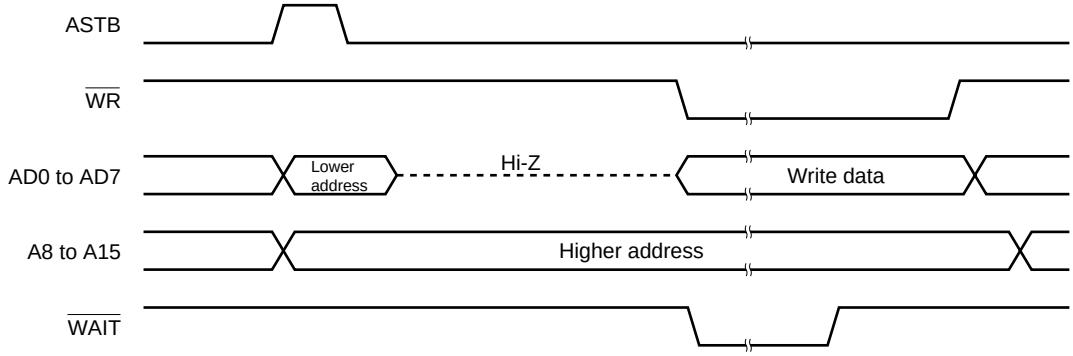
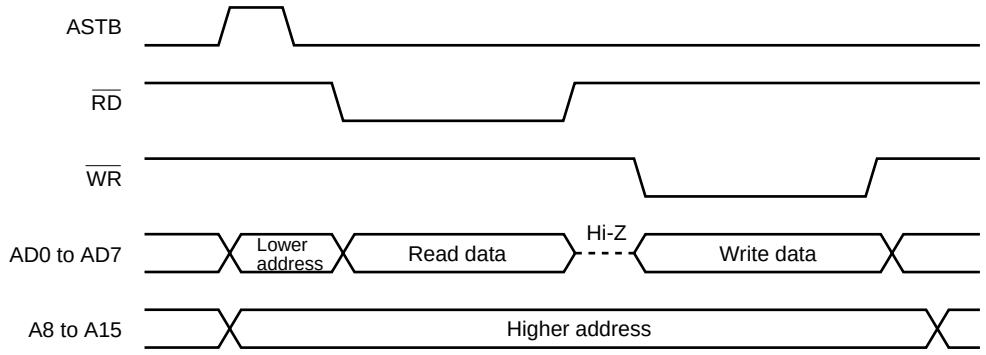
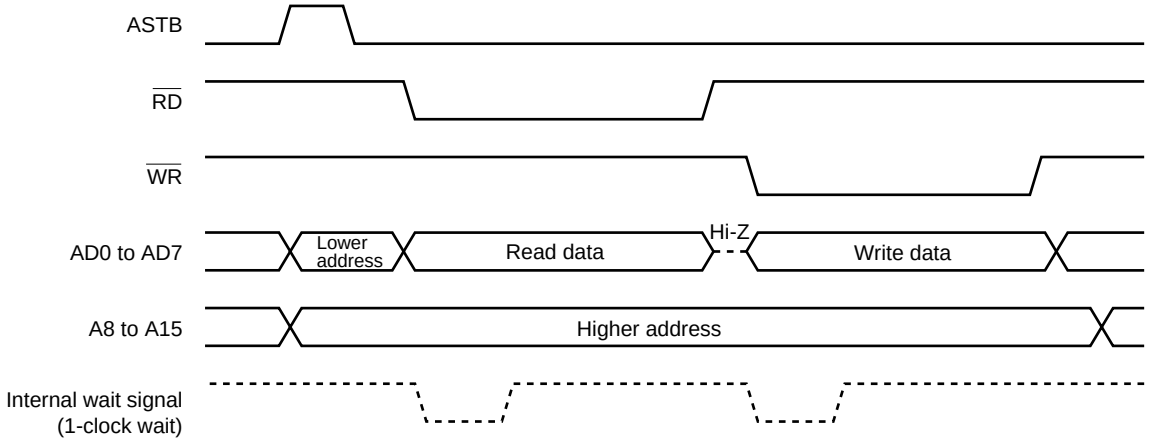


Figure 18-7. External Memory Read Modify Write Timing

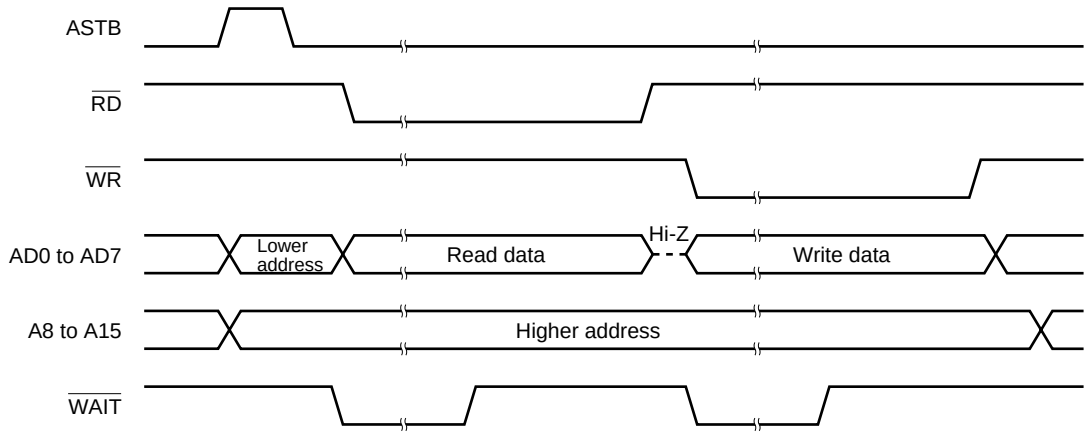
(a) No wait (PW1, PW0 = 0, 0) setting



(b) Wait (PW1, PW0 = 0, 1) setting



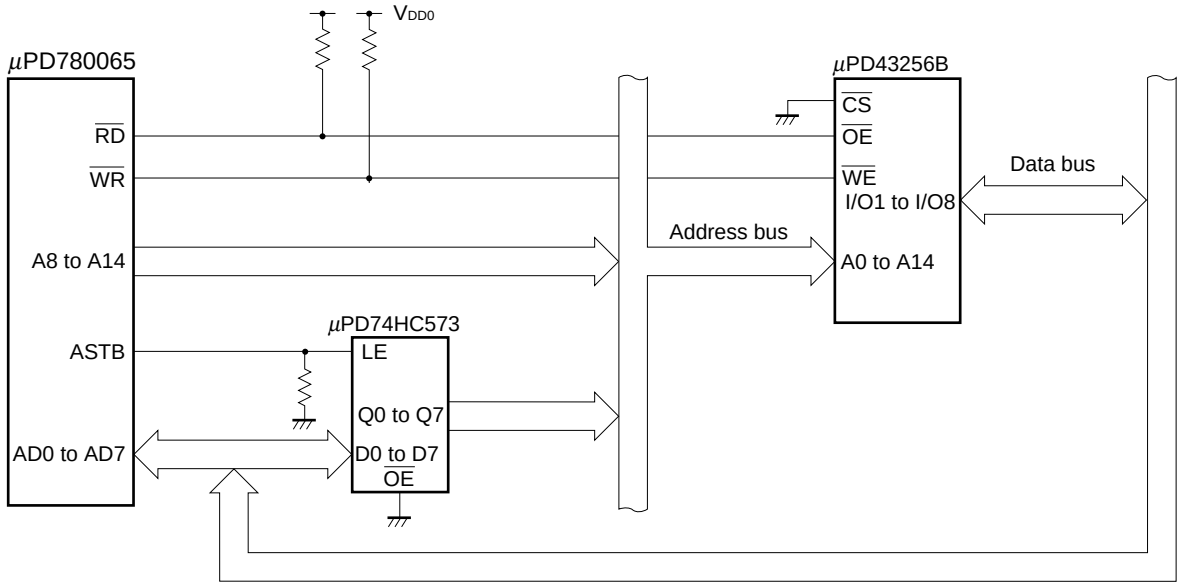
(c) External wait (PW1, PW0 = 1, 1) setting



18.4 Example of Connection with Memory

This section provide an example of connecting the μPD780065 with external memory (in this example, SRAM) in Figure 18-8. In addition, the external device expansion function is used in the full-address mode, and the addresses from 0000H to 9FFFH (40 Kbytes) are allocated for internal ROM, and the addresses after A000H from SRAM.

Figure 18-8. Connection Example of μPD780065 and Memory



CHAPTER 19 STANDBY FUNCTION

19.1 Standby Function and Configuration

19.1.1 Standby function

The standby function is designed to decrease power consumption of the system. The following two modes are available.

(1) HALT mode

Halt instruction execution sets the HALT mode. The HALT mode is intended to stop the CPU operation clock. The system clock oscillator continues oscillating. In this mode, current consumption is not decreased as much as in the STOP mode. However, the HALT mode is effective to restart operation immediately upon interrupt request and to carry out intermittent operations such as watch applications.

(2) STOP mode

Stop instruction execution sets the STOP mode. In the STOP mode, the main system clock oscillator stops, stopping the whole system, thereby considerably reducing the CPU power consumption.

Data memory low-voltage hold (down to V_{DD} = 1.6 V) is possible. Thus, the STOP mode is effective to hold data memory contents with ultra-low current consumption. Because this mode can be cleared upon interrupt request, it enables intermittent operations to be carried out.

However, because a wait time is required to secure an oscillation stabilization time after the STOP mode is cleared, select the HALT mode if it is necessary to start processing immediately upon interrupt request.

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The input/output port output latch and output buffer statuses are also held.

Cautions 1. The STOP mode can be used only when the system operates with the main system clock (subsystem clock oscillation cannot be stopped). The HALT mode can be used with either the main system clock or the subsystem clock.

2. When operation is transferred to the STOP mode, be sure to stop the peripheral hardware operation and execute the STOP instruction.

3. The following sequence is recommended for power consumption reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS0) of the A/D converter mode register (ADM0) to 0 to stop the A/D conversion operation, and then execute the HALT or STOP instruction.

19.1.2 Standby function control register

The wait time after the STOP mode is cleared upon interrupt request is controlled with the oscillation stabilization time select register (OSTS).

OSTS is set by an 8-bit memory manipulation instruction.

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

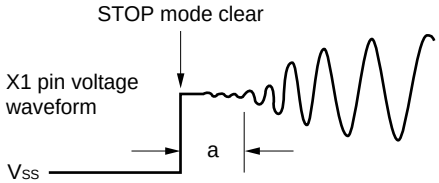
Figure 19-1. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFAH After reset: 04H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time
0	0	0	$2^{12}/f_X$ (488 μs)
0	0	1	$2^{14}/f_X$ (1.95 ms)
0	1	0	$2^{15}/f_X$ (3.91 ms)
0	1	1	$2^{16}/f_X$ (7.81 ms)
1	0	0	$2^{17}/f_X$ (15.6 ms)
Other than above			Setting prohibited

Caution The wait time after the STOP mode is cleared does not include the time (see "a" in the illustration below) from STOP mode clear to clock oscillation start. The time is not included either by $\overline{\text{RESET}}$ input or by interrupt request generation.



Remarks 1. f_X: Main system clock oscillation frequency

2. Values in parentheses are for operation with f_X = 8.38 MHz.

19.2 Standby Function Operations

19.2.1 HALT mode

(1) HALT mode setting and operating statuses

The HALT mode is set by executing the HALT instruction. It can be set with the main system clock or the subsystem clock.

The operating statuses in the HALT mode are described below.

Table 19-1. HALT Mode Operating Statuses

HALT Mode Setting / Item		During HALT Instruction Execution Using Main System Clock		During HALT Instruction Execution Using Subsystem Clock	
		Without Subsystem Clock[Note 1]	With Subsystem Clock[Note 2]	With Main System Clock Oscillation	With Main System Clock Oscillation Stopped
Clock generator		Both main system clock and subsystem clock can be oscillated. Clock supply to CPU stops.			
CPU		Operation stops.			
Port (Output latch)		Status before HALT mode setting is held.			
16-bit timer/event counter		Operable			Operable when TI00 is selected.
8-bit timer/event counter		Operable			Operable when TI50, TI51 are selected as count clock.
Watch timer		Operable when $f_X/2^7$ is selected as count clock	Operable		Operable when f_{XT} is selected as count clock.
Watchdog timer		Operable		Operation stops.	
A/D converter		Stop			
Serial interface		Operable			Operable during external SCK.
External interrupt		Operable			
Bus line during external expansion	AD0 to AD7	High impedance			
	A8 to A15	Status before HALT mode setting is held.			
	ASTB	Low level			
	$\overline{WR}$, $\overline{RD}$	High level			
	$\overline{WAIT}$	High impedance			

Notes 1. Including case when external clock is not supplied.

2. Including case when external clock is supplied.

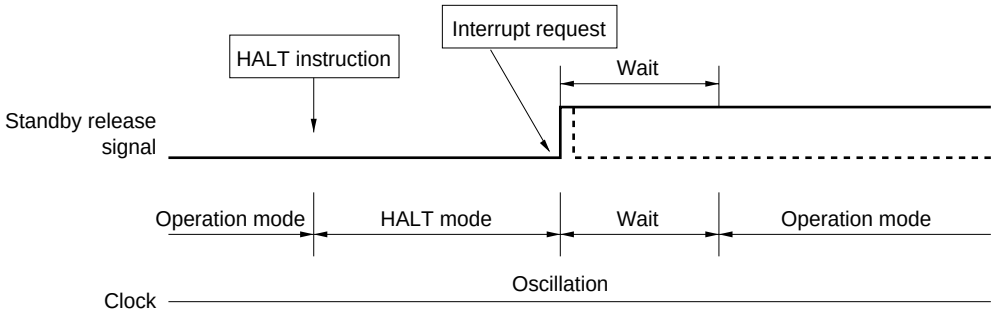
(2) HALT mode clear

The HALT mode can be cleared with the following three types of sources.

(a) Clear upon unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is cleared. If interrupt acknowledge is enabled, vectored interrupt service is carried out. If interrupt acknowledge is disabled, the next address instruction is executed.

Figure 19-2. HALT Mode Clear Upon Interrupt Request Generation



Remarks 1. The broken line indicates the case when the interrupt request which has cleared the standby mode is acknowledged.

2. Wait times are as follows:

- When vectored interrupt service is carried out: 8 or 9 clocks
- When vectored interrupt service is not carried out: 2 or 3 clocks

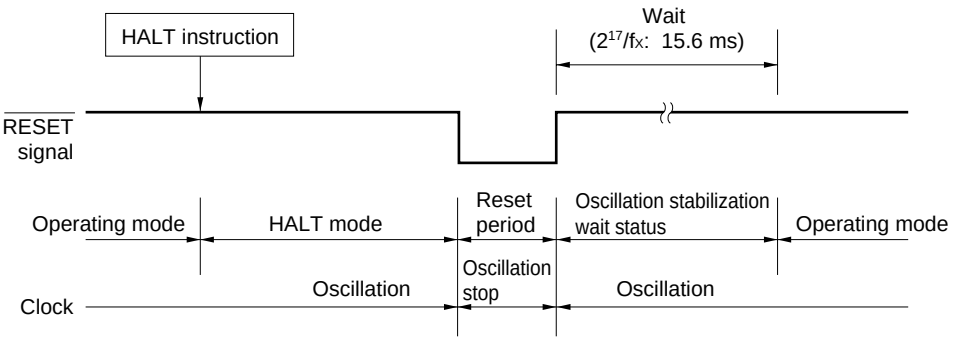
(b) Clear upon non-maskable interrupt request

When an non-maskable interrupt request is generated, the HALT mode is cleared and vectored interrupt service is carried out whether interrupt acknowledge is enabled or disabled.

(c) Clear upon $\overline{\text{RESET}}$ input

When $\overline{\text{RESET}}$ signal is input, HALT mode is released. And, as in the case with normal reset operation, a program is executed after branch to the reset vector address.

Figure 19-3. HALT Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses are for operation with fx = 8.38 MHz.

Table 19-2. Operation after HALT Mode Release

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt service execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt service execution
	1	×	×	×	HALT mode hold
Non-maskable interrupt request	—	—	×	×	Interrupt service execution
$\overline{\text{RESET}}$ input	—	—	×	×	Reset processing

×: don't care

19.2.2 STOP mode

(1) STOP mode setting and operating status

The STOP mode is set by executing the STOP instruction. It can be set only with the main system clock.

Cautions 1. When the STOP mode is set, the X2 pin is internally connected to V_{DD1} via a pull-up resistor to minimize the leakage current at the crystal oscillator. Thus, do not use the STOP mode in a system where an external clock is used for the main system clock.

2. Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction. After the wait set using the oscillation stabilization time select register (OSTS), the operating mode is set.

The operating status in the STOP mode is described in Table 19-3 below.

Table 19-3. STOP Mode Operating Status

STOP Mode Setting / Item		With Subsystem Clock	Without Subsystem Clock
Clock generator		Only main system clock oscillation is stopped.	
CPU		Operation stops.	
Port (Output latch)		Status before STOP mode setting is held.	
16-bit timer/event counter		Operation stops.	
8-bit timer/event counter		Operable only when TI50, TI51 are selected as count clock.	
Watch timer		Operable when f_{XT} is selected as counter clock.	Operation stops.
Watchdog timer		Operation stops.	
Clock output		PCL at low level.	
A/D converter		Operation stops	
Serial interface	Other than UART	Operable only when externally supplied clock is specified as the serial clock.	
	UART	Operation stops. (transmit shift register (TXS0), receive shift register (RX0), and receive buffer register (RXB0) hold the value just before the clock stop.)	
External interrupt		Operable	
Bus line during external expansion	AD0 to AD7	High impedance	
	A8 to A15	Status before STOP mode setting is held.	
	ASTB	Low level	
	$\overline{WR}$, $\overline{RD}$	High level	
	$\overline{WAIT}$	High impedance	

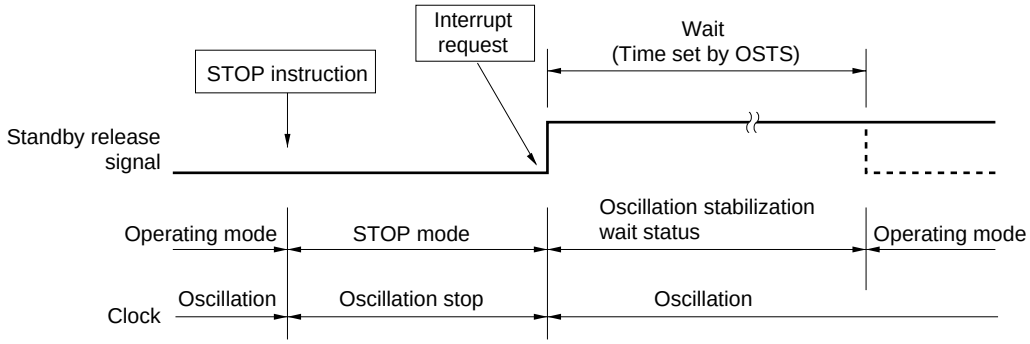
(2) STOP mode release

The STOP mode can be released by the following two types of sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the STOP mode is released. If interrupt acknowledge is enabled after the lapse of oscillation stabilization time, vectored interrupt service is carried out. If interrupt acknowledge is disabled, the next address instruction is executed.

Figure 19-4. STOP Mode Release by Interrupt Request Generation

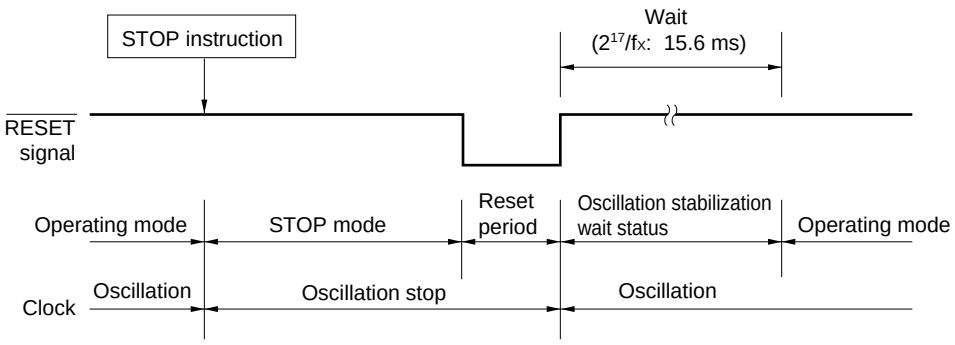


Remark The broken line indicates the case when the interrupt request which has cleared the standby status is acknowledged.

(b) Release by $\overline{\text{RESET}}$ input

The STOP mode is cleared when $\overline{\text{RESET}}$ signal is input, and after the lapse of oscillation stabilization time, reset operation is carried out.

Figure 19-5. STOP Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fx: Main system clock oscillation frequency

2. Values in parentheses are for operation with fx = 8.38 MHz.

Table 19-4. Operation after STOP Mode Release

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt service execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt service execution
	1	×	×	×	STOP mode hold
$\overline{\text{RESET}}$ input	—	—	×	×	Reset processing

×: don't care

CHAPTER 20 RESET FUNCTION

20.1 Reset Function

The following two operations are available to generate the reset function.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer runaway time detection

External reset and internal reset have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H by $\overline{\text{RESET}}$ input. When a low level is input to the $\overline{\text{RESET}}$ pin or the watchdog timer overflows, a reset is applied and each hardware is set to the status shown in Table 20-1. Each pin has high impedance during reset input or during oscillation stabilization time just after reset clear.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is cleared and program execution starts after the lapse of oscillation stabilization time $2^{17}/f_X$. The reset applied by watchdog timer overflow is automatically cleared after a reset and program execution starts after the lapse of oscillation stabilization time $2^{17}/f_X$ (see Figures 20-2 to 20-4).

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
2. During reset input, main system clock oscillation remains stopped but subsystem clock oscillation continues.
3. When the STOP mode is cleared by reset, the STOP mode contents are held during reset input. However, the port pin becomes high-impedance.

Figure 20-1. Reset Function Block Diagram

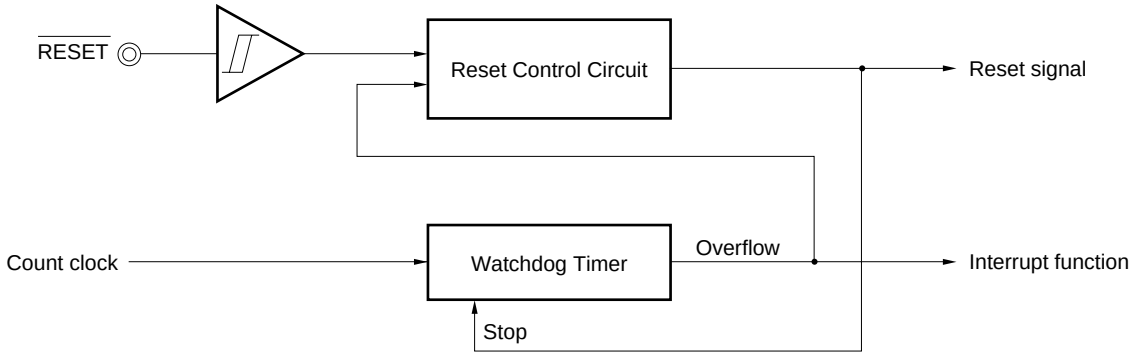


Figure 20-2. Timing of Reset by $\overline{\text{RESET}}$ Input

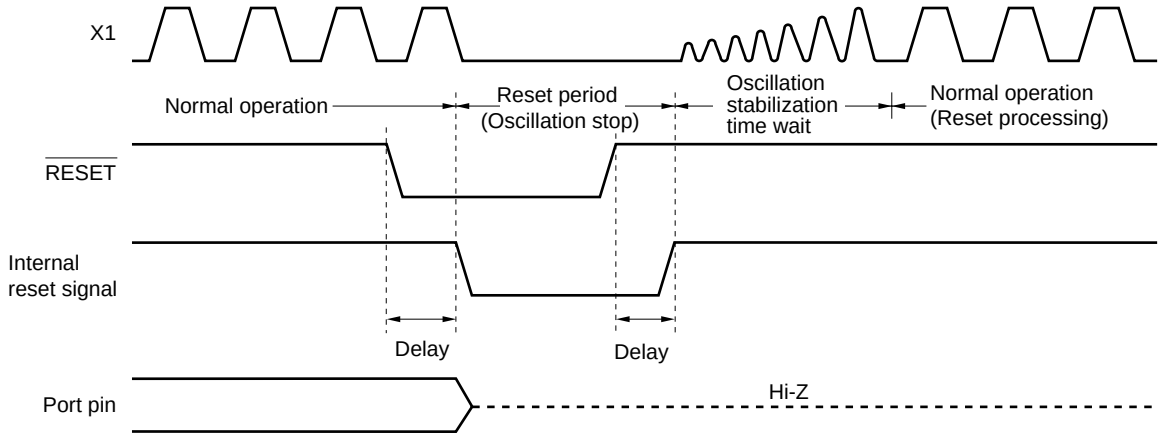


Figure 20-3. Timing of Reset due to Watchdog Timer Overflow

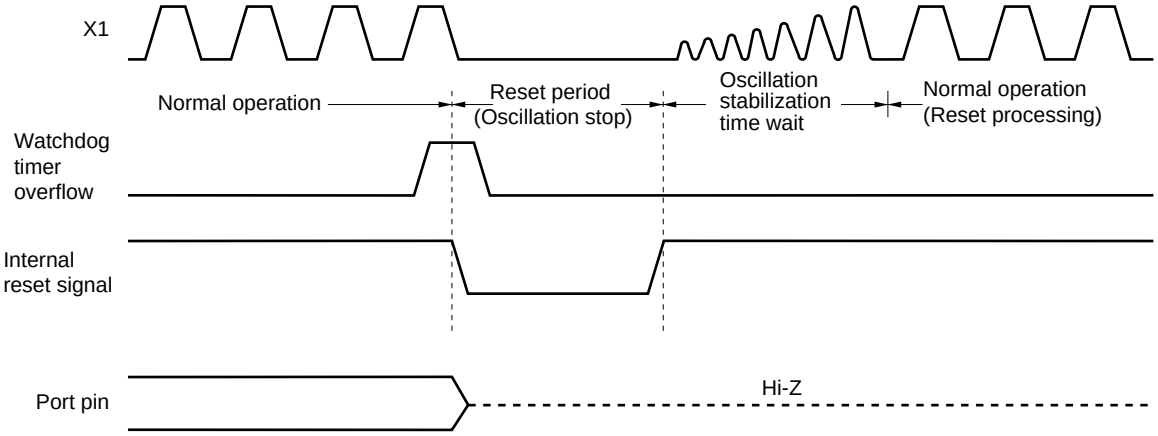


Figure 20-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input

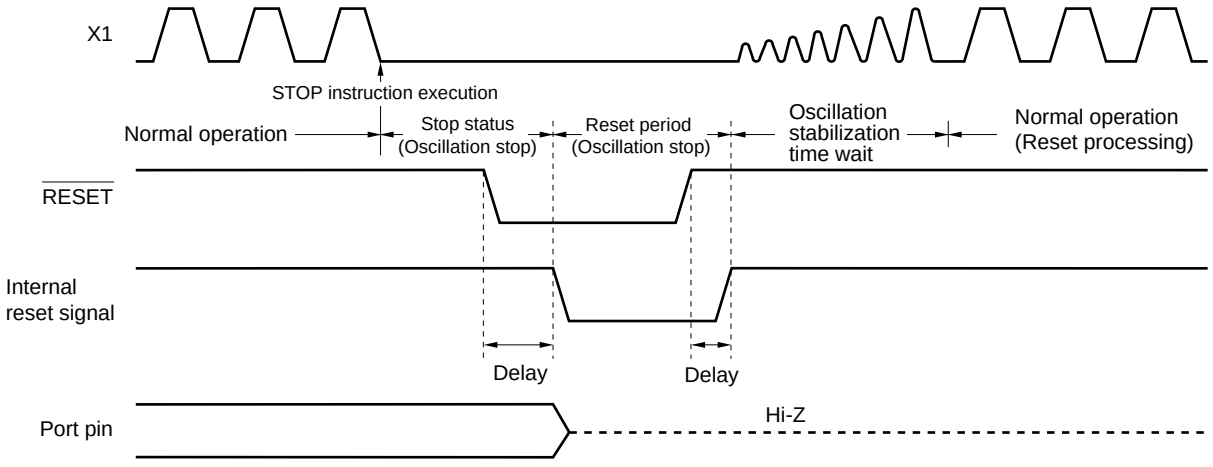


Table 20-1. Hardware Statuses after Reset (1/2)

Hardware		Status After Reset
Program counter (PC)[Note 1]		Contents of reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General register	Undefined[Note 2]
Port (Output latch)	Ports 1 to 3, 7 to 9 (P1 to P3, P7 to P9)	00H
	Ports 4 to 6 (P4 to P6)	Undefined
Port mode registers (PM0, PM2 to PM9)		FFH
Pull-up resistor option registers (PU0, PU2 to PU9)		00H
Processor clock control register (PCC)		04H
Memory size switching register (IMS)		CFH[Note 3]
Internal expansion RAM size switching register (IXS)		0CH[Note 4]
Memory expansion mode register (MEM)		00H
Memory expansion wait setting register (MM)		10H
Oscillation stabilization time selection register (OSTS)		04H
16-bit timer/event counter	Timer register (TM0)	0000H
	Capture/compare register (CR00, CR01)	Undefined
	Prescaler mode register (PRM0)	00H
	Mode control register (TMC0)	00H
	Output control register (TOC0)	00H
8-bit timer/event counter	Timer counters (TM50, TM51)	00H
	Compare registers (CR50, CR51)	Undefined
	Clock selection registers (TCL50, TCL51)	00H
	Mode control register (TMC50, TMC51)	04H
Watch timer	Mode control register (WTM)	00H
Watchdog timer	Clock selection register (WDCS)	00H
	Mode register (WDTM)	00H

Notes 1. During reset input or oscillation stabilization time wait, only the PC contents among the hardware statuses becomes undefined. All other hardware statuses remain unchanged after reset.

2. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

3. Although the initial value is CFH, use the following value to be set for each version.
μPD780065: CAH
μPD78F0066: CCH or value for mask ROM versions

4. Although the initial value is 0CH, always set 04H.

Table 20-1. Hardware Statuses after Reset (2/2)

Hardware		Status After Reset
Clock output controller	Clock output selection register (CKS)	00H
A/D converter	Conversion result registers (ADCR0)	00H
	Mode register (ADM0)	00H
	Analog input channel specification register (ADS0)	00H
Serial interface (UART0)	Asynchronous serial interface mode register (ASIM0)	00H
	Asynchronous serial interface status register (ASIS0)	00H
	Baud rate generator control register (BRGC0)	00H
	Transmit shift register (TXS0)	FFH
	Receive buffer register (RXB0)	
Serial interface (SIO1)	Shift register (SIO1)	Undefined
	Automatic data transmission/reception address pointer (ADTP0)	00H
	Operating mode register (CSIM1)	00H
	Automatic data transmission/reception control register (ADTC0)	00H
	Automatic data transmission/reception interval specification register (ADTI0)	00H
Serial interface (SIO3)	Shift registers (SIO30, SIO31)	Undefined
	Operating mode registers (CSIM30, CSIM31)	00H
Interrupt	Request flag registers (IF0L, IF0H, IF1L)	00H
	Mask flag registers (MK0L, MK0H, MK1L)	FFH
	Priority specify flag registers (PR0L, PR0H, PR1L)	FFH
	External interrupt rising edge enable register (EGP)	00H
	External interrupt falling edge enable register (EGN)	00H

CHAPTER 21 μPD78F0066

The μPD78F0066 is provided as the flash memory version of the μPD780065 Subseries.

The μPD78F0066 replaces the internal mask ROM of the μPD780065 with flash memory to which a program can be written, deleted and overwritten while mounted on the substrate. Table 21-1 lists the differences between the μPD78F0066 and the mask ROM version.

Table 21-1. Differences between μPD78F0066 and Mask ROM Version

Item	μPD78F0066	Mask ROM Version (μPD780065)
Internal ROM configuration	Flash memory	Mask ROM
Internal ROM capacity	48 Kbytes	40 Kbytes
IC pin	None	Available
V_{PP} pin	Available	None
Electrical specifications	Refer to data sheet of each product.	

Caution Flash memory versions and mask ROM versions differ in their noise immunity and noise radiation. If replacing flash memory versions with mask ROM versions when changing from test production to mass production, be sure to perform sufficient evaluation with CS versions (not ES versions) of mask ROM versions.

21.1 Memory Size Switching Register

The μPD78F0066 allows users to select the internal memory capacity using the memory size switching register (IMS) so that the same memory map as that of the μPD780065 with a different size of internal memory capacity can be achieved.

IMS is set by using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IMS to CFH.

★ **Caution The initial value of IMS is CFH (setting prohibited). As the initial program setting, be sure to set the following values.**

μPD780065: CAH

μPD78F0066: CCH or a value supporting the Mask ROM version

Figure 21-1. Format of Memory Size Switching Register (IMS)

★ Address: FFF0H After reset: CFH R/W

Symbol	7	6	5	4	3	2	1	0
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0

RAM2	RAM1	RAM0	Internal high-speed RAM capacity selection
1	1	0	1024 bytes
Other than above			Setting prohibited

ROM3	ROM2	ROM1	ROM0	Internal ROM capacity selection
1	0	1	0	40 Kbytes
1	1	0	0	48 Kbytes
Other than above				Setting prohibited

21.2 Internal Expansion RAM Size Switching Register

The internal expansion RAM size switching register (IXS) is a register used to set internal expansion RAM capacity.
IXS is set by using 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets IXS to 0CH.

Caution Set IXS to 04H as the initial value of the program. The initial value of IXS is 0CH (setting prohibited).

Figure 21-2. Format of Internal Expansion RAM Size Switching Register (IXS)

Address: FFF4H After reset: 0CH R/W

Symbol	7	6	5	4	3	2	1	0
IXS	0	0	0	IXRAM4	IXRAM3	IXRAM2	IXRAM1	IXRAM0

IXRAM4	IXRAM3	IXRAM2	IXRAM1	IXRAM0	Internal expansion RAM capacity selection
0	0	1	0	0	4096 bytes
Other than above					Setting prohibited

★ 21.3 Flash Memory Programming

On-board writing of flash memory (with device mounted on target system) is supported.

On-board writing is done after connecting a dedicated flash programmer (Flashpro II (type FL-PR2), Flashpro III (type FL-PR3, PG-FP3)) to the host machine and target system.

Moreover, writing to flash memory can also be performed using a flash memory writing adapter connected to Flashpro II or Flashpro III.

Remark FL-PR2 and FL-PR3 are products of NAITO DENSEI MACHIDA MFG. CO., LTD.

21.3.1 Selection of transmission method

Writing to flash memory is performed using Flashpro II or Flashpro III and serial communication. Select the transmission method for writing from Table 21-2. For the selection of the transmission method, a format like the one shown in Figure 21-3 is used. The transmission methods are selected with the V_{PP} pulse numbers shown in Table 21-2.

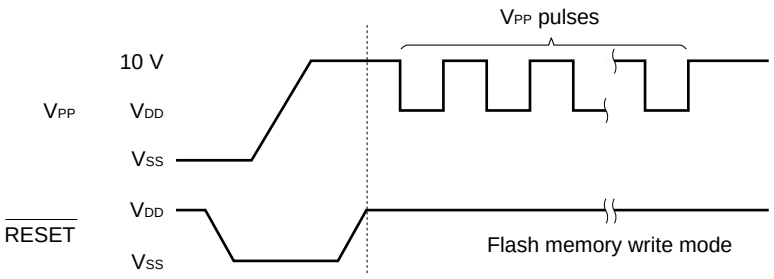
Table 21-2. Transmission Method List

Transmission Method	Number of Channels	Pin Used	Number of V_{PP} Pulses
3-wire serial I/O	2	SI31/P92 SO31/P91 $\overline{\text{SCK31}}$/P90	0
		SI1/P84 SO1/P83 $\overline{\text{SCK1}}$/P82	1
UART	1	RxD0/P73 TxD0/P72	8

Cautions 1. Be sure to select the number of V_{PP} pulses shown in Table 21-2 for the transmission method.

2. If performing write operations to flash memory with the UART transmission method, set the main system clock oscillation frequency to 3 MHz or higher.

Figure 21-3. Format of Transmission Method Selection



21.3.2 Flash memory programming function

Flash memory writing is performed through command and data transmit/receive operations using the selected transmission method. The main functions are listed in Table 21-3.

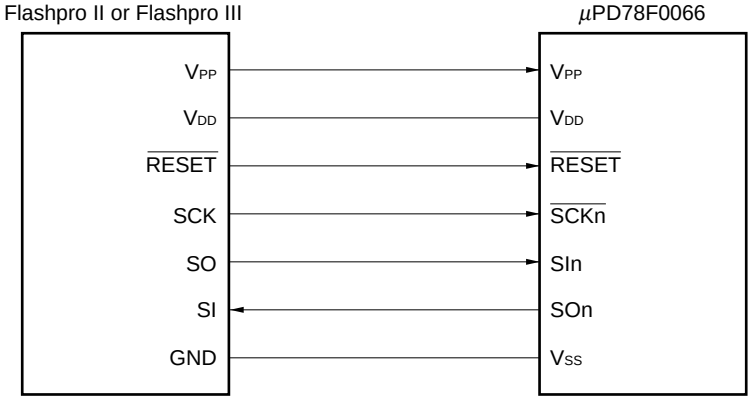
Table 21-3. Main Functions of Flash Memory Programming

Function	Description
Reset	Used to detect write stop and transmission synchronization.
Batch verify	Compares entire memory contents and input data.
Batch delete	Deletes the entire memory contents.
Batch blank check	Checks the deletion status of the entire memory.
High-speed write	Performs writing to flash memory according to write start address and number of write data (bytes).
Continuous write	Performs successive write operations using the data input with high-speed write operation.
Status	Checks the current operation mode and operation end.
Oscillation frequency setting	Inputs the resonator oscillation frequency information.
Delete time setting	Inputs the memory delete time.
Baud rate setting	Sets the transmission rate when the UART method is used.
Silicon signature read	Outputs the device name, memory capacity, and device block information.

21.3.3 Connection of Flashpro II or Flashpro III

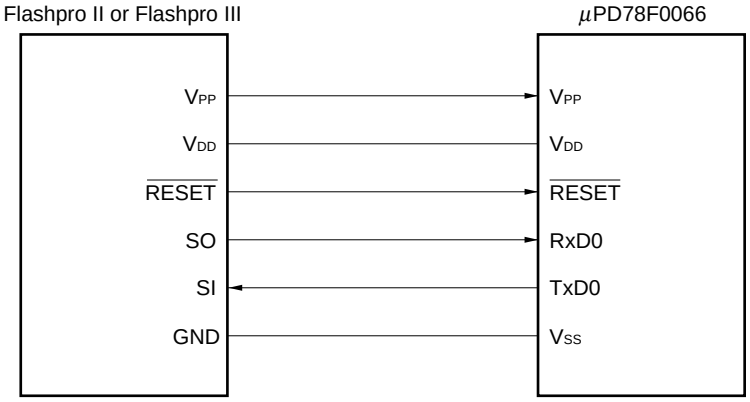
Connection of the Flashpro II or Flashpro III and the μPD78F0066 differs depending on communication method (3-wire serial I/O or UART). Each type of connection is shown in Figures 21-4 and 21-5.

Figure 21-4. Connection of Flashpro II or Flashpro III Using 3-Wire Serial I/O Method



n = 1, 31

Figure 21-5. Connection of Flashpro II or Flashpro III Using UART Method



CHAPTER 22 INSTRUCTION SET

This chapter lists each instruction set of the μPD780065 Subseries in table form. For details of its operation and operation code, refer to the separate document **78K/0 Series User's Manual—Instructions (U12326E)**.

22.1 Symbols Used in Operation List

22.1.1 Operand identifiers and description methods

Operands are described in “Operand” column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for detail). When there are two or more description methods, select one of them. Alphabetic letters in capitals and symbols, #, !, $ and [ ] are key words and must be described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- $: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, $, and [ ] symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for description.

Table 22-1. Operand Identifiers and Description Methods

Identifier	Description Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7),
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special-function register symbol**Note**
sfrp	Special-function register symbol (16-bit manipulatable register even addresses only)**Note**
saddr	FE20H to FF1FH Immediate data or labels
saddrp	FE20H to FF1FH Immediate data or labels (even address only)
addr16	0000H to FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions)
addr11	0800H to 0FFFH Immediate data or labels
addr5	0040H to 007FH Immediate data or labels (even address only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special-function register symbols, refer to **Table 3-3. Special-Function Register List**.

22.1.2 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)

22.1.3 Description of “flag operation” column

(Blank): Not affected
0: Cleared to 0
1: Set to 1
×: Set/cleared according to the result
R: Previously saved value is restored

22.2 Operation List

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	**MOV**	r, #byte	2	4	–	r ← byte	
		saddr, #byte	3	6	7	(saddr) ← byte	
		sfr, #byte	3	–	7	sfr ← byte	
		A, r Note 3	1	2	–	A ← r	
		r, A Note 3	1	2	–	r ← A	
		A, saddr	2	4	5	A ← (saddr)	
		saddr, A	2	4	5	(saddr) ← A	
		A, sfr	2	–	5	A ← sfr	
		sfr, A	2	–	5	sfr ← A	
		A, !addr16	3	8	9 + n	A ← (addr16)	
		!addr16, A	3	8	9 + m	(addr16) ← A	
		PSW, #byte	3	–	7	PSW ← byte	× × ×
		A, PSW	2	–	5	A ← PSW	
		PSW, A	2	–	5	PSW ← A	× × ×
		A, [DE]	1	4	5 + n	A ← (DE)	
		[DE], A	1	4	5 + m	(DE) ← A	
		A, [HL]	1	4	5 + n	A ← (HL)	
		[HL], A	1	4	5 + m	(HL) ← A	
		A, [HL + byte]	2	8	9 + n	A ← (HL + byte)	
		[HL + byte], A	2	8	9 + m	(HL + byte) ← A	
		A, [HL + B]	1	6	7 + n	A ← (HL + B)	
		[HL + B], A	1	6	7 + m	(HL + B) ← A	
		A, [HL + C]	1	6	7 + n	A ← (HL + C)	
		[HL + C], A	1	6	7 + m	(HL + C) ← A	
	XCH	A, r Note 3	1	2	–	A ↔ r	
		A, saddr	2	4	6	A ↔ (saddr)	
		A, sfr	2	–	6	A ↔ (sfr)	
		A, !addr16	3	8	10 + n + m	A ↔ (addr16)	
		A, [DE]	1	4	6 + n + m	A ↔ (DE)	
		A, [HL]	1	4	6 + n + m	A ↔ (HL)	
		A, [HL + byte]	2	8	10 + n + m	A ↔ (HL + byte)	
		A, [HL + B]	2	8	10 + n + m	A ↔ (HL + B)	
		A, [HL + C]	2	8	10 + n + m	A ↔ (HL + C)	

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed.

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	**MOVW**	rp, #word	3	6	–	rp ← word			
		saddrp, #word	4	8	10	(saddrp) ← word			
		sfrp, #word	4	–	10	sfrp ← word			
		AX, saddrp	2	6	8	AX ← (saddrp)			
		saddrp, AX	2	6	8	(saddrp) ← AX			
		AX, sfrp	2	–	8	AX ← sfrp			
		sfrp, AX	2	–	8	sfrp ← AX			
		AX, rp **Note 3**	1	4	–	AX ← rp			
		rp, AX **Note 3**	1	4	–	rp ← AX			
		AX, !addr16	3	10	12 + 2n	AX ← (addr16)			
		!addr16, AX	3	10	12 + 2m	(addr16) ← AX			
	XCHW	AX, rp **Note 3**	1	4	–	AX ↔ rp			
8-bit operation	**ADD**	A, #byte	2	4	–	A, CY ← A + byte	×	×	×
		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 + n	A, CY ← A + (addr16)	×	×	×
		A, [HL]	1	4	5 + n	A, CY ← A + (HL)	×	×	×
		A, [HL + byte]	2	8	9 + n	A, CY ← A + (HL + byte)	×	×	×
		A, [HL + B]	2	8	9 + n	A, CY ← A + (HL + B)	×	×	×
		A, [HL + C]	2	8	9 + n	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 + n	A, CY ← A + (addr16) + CY	×	×	×
		A, [HL]	1	4	5 + n	A, CY ← A + (HL) + CY	×	×	×
		A, [HL + byte]	2	8	9 + n	A, CY ← A + (HL + byte) + CY	×	×	×
		A, [HL + B]	2	8	9 + n	A, CY ← A + (HL + B) + CY	×	×	×
		A, [HL + C]	2	8	9 + n	A, CY ← A + (HL + C) + CY	×	×	×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Only when rp = BC, DE or HL

4. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	**SUB**	A, #byte	2	4	–	A, CY ← A – byte	×	×	×
		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 + n	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	4	5 + n	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	8	9 + n	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9 + n	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9 + n	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 + n	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	4	5 + n	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	8	9 + n	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	8	9 + n	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	8	9 + n	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 + n	A ← A ∧ (addr16)	×		
		A, [HL]	1	4	5 + n	A ← A ∧ [HL]	×		
		A, [HL + byte]	2	8	9 + n	A ← A ∧ [HL + byte]	×		
		A, [HL + B]	2	8	9 + n	A ← A ∧ [HL + B]	×		
		A, [HL + C]	2	8	9 + n	A ← A ∧ [HL + C]	×		

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit operation	**OR**	A, #byte	2	4	–	A ← 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 + n	A ← A ∨ (addr16)	×
		A, [HL]	1	4	5 + n	A ← A ∨ (HL)	×
		A, [HL + byte]	2	8	9 + n	A ← A ∨ (HL + byte)	×
		A, [HL + B]	2	8	9 + n	A ← A ∨ (HL + B)	×
		A, [HL + C]	2	8	9 + n	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 + n	A ← A ⊻ (addr16)	×
		A, [HL]	1	4	5 + n	A ← A ⊻ (HL)	×
		A, [HL + byte]	2	8	9 + n	A ← A ⊻ (HL + byte)	×
		A, [HL + B]	2	8	9 + n	A ← A ⊻ (HL + B)	×
		A, [HL + C]	2	8	9 + n	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 + n	A – (addr16)	× × ×
		A, [HL]	1	4	5 + n	A – (HL)	× × ×
		A, [HL + byte]	2	8	9 + n	A – (HL + byte)	× × ×
		A, [HL + B]	2	8	9 + n	A – (HL + B)	× × ×
		A, [HL + C]	2	8	9 + n	A – (HL + C)	× × ×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	**ADDW**	AX, #word	3	6	–	AX, CY ← AX + word	×	×	×
	SUBW	AX, #word	3	6	–	AX, CY ← AX – word	×	×	×
	CMPW	AX, #word	3	6	–	AX – word	×	×	×
Multiply/ divide	**MULU**	X	2	16	–	AX ← A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C			
Increment/ decrement	**INC**	r	1	2	–	r ← r + 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) + 1	×	×	
	DEC	r	1	2	–	r ← r – 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) – 1	×	×	
	INCW	rp	1	4	–	rp ← rp + 1			
	DECW	rp	1	4	–	rp ← rp – 1			
Rotate	**ROR**	A, 1	1	2	–	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$ time			×
	ROL	A, 1	1	2	–	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$ time			×
	RORC	A, 1	1	2	–	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$ time			×
	ROLC	A, 1	1	2	–	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$ time			×
	ROR4	[HL]	2	10	12 + n + m	$A_{3-0} \leftarrow (HL)_{3-0}$, $(HL)_{7-4} \leftarrow A_{3-0}$, $(HL)_{3-0} \leftarrow (HL)_{7-4}$			
	ROL4	[HL]	2	10	12 + n + m	$A_{3-0} \leftarrow (HL)_{7-4}$, $(HL)_{3-0} \leftarrow A_{3-0}$, $(HL)_{7-4} \leftarrow (HL)_{3-0}$			
BCD adjust	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipulate	**MOV1**	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7 + n	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 + n + m	(HL).bit ← CY			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Bit manipulate	**AND1**	CY, saddr.bit	3	6	7	CY ← CY ∧ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ∧ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ∧ A.bit	×
		CY, PSW.bit	3	–	7	CY ← CY ∧ PSW.bit	×
		CY, [HL].bit	2	6	7 + n	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 + n	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 + n	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 + n + m	(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 + n + m	(HL).bit ← 0	
	SET1	CY	1	2	–	CY ← 1	1
	CLR1	CY	1	2	–	CY ← 0	0
	NOT1	CY	1	2	–	CY ← $\overline{CY}$	×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Call/return	**CALL**	!addr16	3	7	–	$(SP-1) \leftarrow (PC+3)_H$, $(SP-2) \leftarrow (PC+3)_L$, $PC \leftarrow$ addr16, $SP \leftarrow SP-2$	
	CALLF	!addr11	2	5	–	$(SP-1) \leftarrow (PC+2)_H$, $(SP-2) \leftarrow (PC+2)_L$, $PC_{15-11} \leftarrow 00001$, $PC_{10-0} \leftarrow$ addr11, $SP \leftarrow SP-2$	
	CALLT	[addr5]	1	6	–	$(SP-1) \leftarrow (PC+1)_H$, $(SP-2) \leftarrow (PC+1)_L$, $PC_H \leftarrow$ (00000000, addr5 + 1), $PC_L \leftarrow$ (00000000, addr5), $SP \leftarrow SP-2$	
	BRK		1	6	–	$(SP-1) \leftarrow PSW$, $(SP-2) \leftarrow (PC+1)_H$, $(SP-3) \leftarrow (PC+1)_L$, $PC_H \leftarrow$ (003FH), $PC_L \leftarrow$ (003EH), $SP \leftarrow SP-3$, $IE \leftarrow 0$	
	RET		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $SP \leftarrow SP+2$	
	RETI		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP+2)$, $SP \leftarrow SP+3$, $NMIS \leftarrow 0$	R R R
	RETB		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP+2)$, $SP \leftarrow SP+3$	R R R
Stack manipulate	**PUSH**	PSW	1	2	–	$(SP-1) \leftarrow PSW$, $SP \leftarrow SP-1$	
		rp	1	4	–	$(SP-1) \leftarrow rp_H$, $(SP-2) \leftarrow rp_L$, $SP \leftarrow SP-2$	
	POP	PSW	1	2	–	$PSW \leftarrow (SP)$, $SP \leftarrow SP+1$	R R R
		rp	1	4	–	$rp_H \leftarrow (SP+1)$, $rp_L \leftarrow (SP)$, $SP \leftarrow SP+2$	
	MOVW	SP, #word	4	–	10	$SP \leftarrow$ word	
		SP, AX	2	–	8	$SP \leftarrow AX$	
		AX, SP	2	–	8	$AX \leftarrow SP$	
Unconditional branch	**BR**	!addr16	3	6	–	$PC \leftarrow$ addr16	
		$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8	
		AX	2	8	–	$PC_H \leftarrow A$, $PC_L \leftarrow X$	
Conditional branch	**BC**	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if CY = 1	
	BNC	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if CY = 0	
	BZ	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if Z = 1	
	BNZ	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if Z = 0	

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Conditional branch	**BT**	saddr.bit, $addr16	3	8	9	PC ← PC + 3 + jdisp8 if(saddr.bit) = 1	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 1	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1	
		PSW.bit, $addr16	3	–	9	PC ← PC + 3 + jdisp8 if PSW.bit = 1	
		[HL].bit, $addr16	3	10	11 + n	PC ← PC + 3 + jdisp8 if (HL).bit = 1	
	BF	saddr.bit, $addr16	4	10	11	PC ← PC + 4 + jdisp8 if(saddr.bit) = 0	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 0	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 0	
		PSW.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if PSW. bit = 0	
		[HL].bit, $addr16	3	10	11 + n	PC ← PC + 3 + jdisp8 if (HL).bit = 0	
	BTCLR	saddr.bit, $addr16	4	10	12	PC ← PC + 4 + jdisp8 if(saddr.bit) = 1 then reset(saddr.bit)	
		sfr.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit	
		PSW.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	× × ×
		[HL].bit, $addr16	3	10	12 + n + m	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit	
	DBNZ	B, $addr16	2	6	–	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0	
		C, $addr16	2	6	–	C ← C –1, then PC ← PC + 2 + jdisp8 if C ≠ 0	
		saddr. $addr16	3	8	10	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if(saddr) ≠ 0	
CPU control	**SEL**	RBn	2	4	–	RBS1, 0 ← n	
	NOP		1	2	–	No Operation	
	EI		2	–	6	IE ← 1(Enable Interrupt)	
	DI		2	–	6	IE ← 0(Disable Interrupt)	
	HALT		2	6	–	Set HALT Mode	
	STOP		2	6	–	Set STOP Mode	

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to internal ROM program.

3. n is the number of waits when external memory expansion area is read from.

4. m is the number of waits when external memory expansion area is written to.

22.3 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, HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

First Operand \ Second Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

First Operand \ Second Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

[MEMO]

APPENDIX A DEVELOPMENT TOOLS

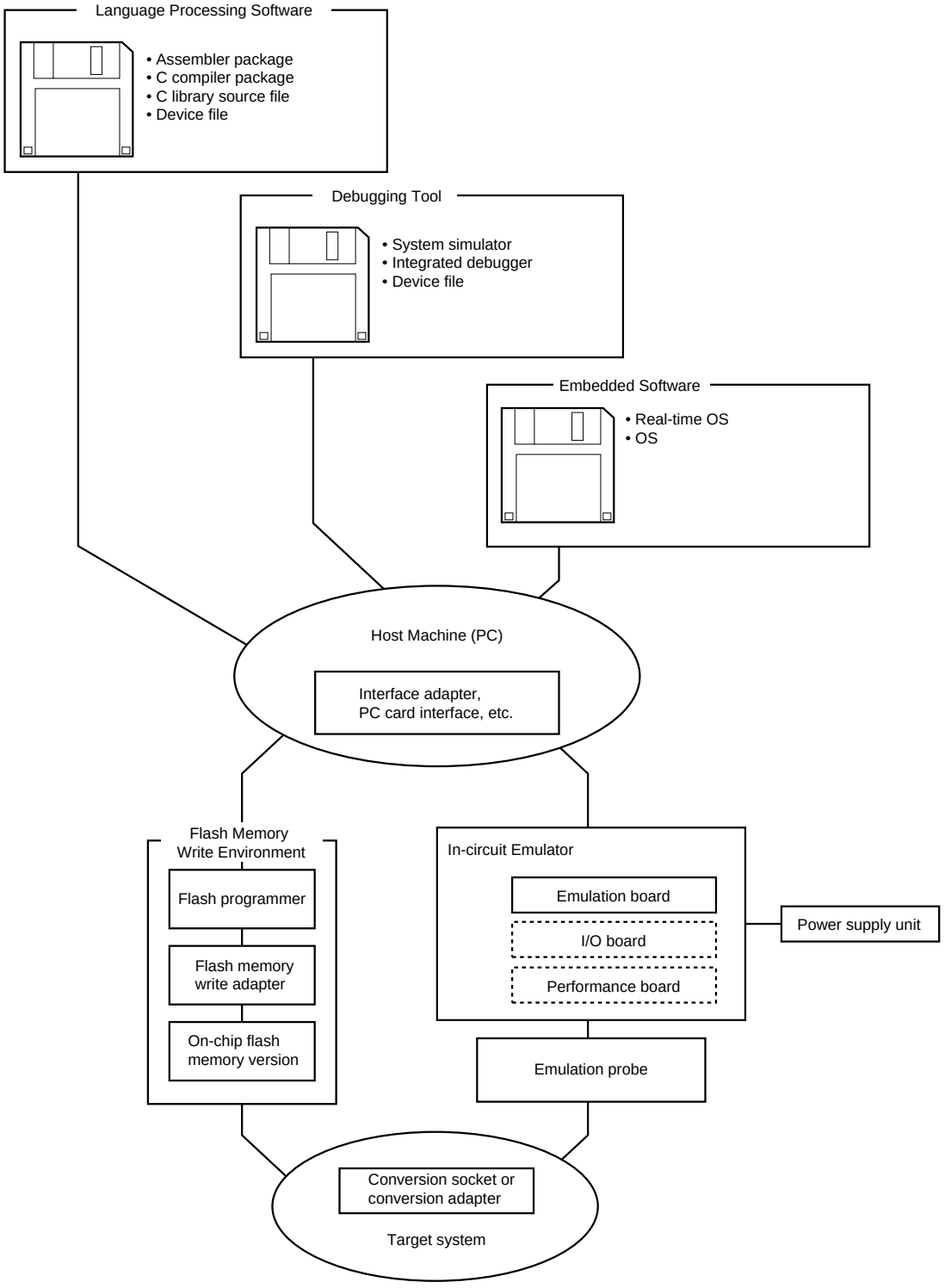
The following shows development tools necessary for the development of systems that employ the μPD780065 Subseries.

★ • Support for PC98-NX series
Unless otherwise specified, products supported by IBM PC/AT™ and compatibles can be used for the PC98-NX series. When using the PC98-NX series, refer to the explanations of IBM PC/AT and compatibles.

★ • Windows
Unless otherwise specified, “Windows” indicates following OSs.
  - Windows 3.1
  - Windows95
  - WindowsNT™ Ver. 4.0

Figure A-1. Development Tool Configuration (1/2)

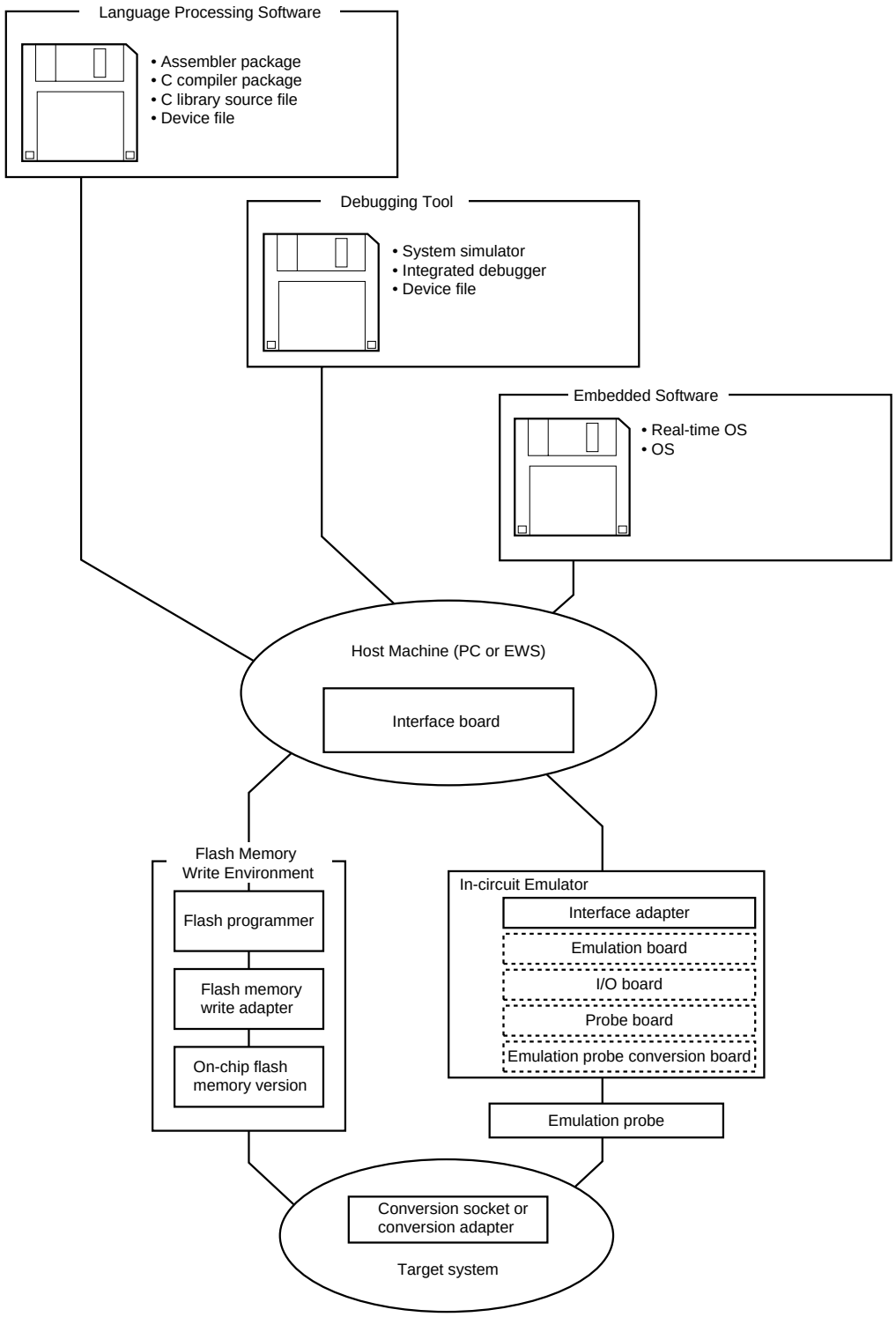
(1) When using the in-circuit emulator IE-78K0-NS



Remark Items in broken-line boxes differ according to the development environment. Refer to **A.3.1. Hardware**.

Figure A-1. Development Tool Configuration (2/2)

(2) When using the in-circuit emulator IE-78001-R-A



Remark Items in broken-line boxes differ according to the development environment. Refer to **A.3.1. Hardware**.

A.1 Language Processing Software

RA78K/0 Assembler Package	This assembler converts programs written in mnemonics into an 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 an optical device file (DF780066). **<Precaution when using RA78K/0 in PC environment>** This assembler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part Number: μS××××RA78K0
CC78K/0 C Compiler Package	This compiler converts programs written in C language into object codes executable with a microcontroller. This compiler should be used in combination with an optical assembler package and device file. **<Precaution when using RA78K/0 in PC environment>** This C compiler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part Number: μS××××CC78K0
★ DF780066[Note] Device File	This file contains information peculiar to the device. This device file should be used in combination with an optical tool (RA78K/0, CC78K/0, SM78K0, ID78K0-NS, and ID78K0). Corresponding OS and host machine differ depending on the tool to be used with.
	Part Number: μS××××DF780066
CC78K/0-L C Library Source File	This is a source file of functions configuring the object library included in the C compiler package (CC78K/0). This file is required to match the object library included in C compiler package to the customer's specifications.
	Part Number: μS××××CC78K0-L

Note The DF780066 can be used in common with the RA78K/0, CC78K/0, SM78K0, ID78K0-NS, and ID78K0. The DF780066 is under development.

★ **Remark** ××××in the part number differs depending on the host machine and OS used.

μS××××RA78K0
μS××××CC78K0
μS××××DF780066
μS××××CC78K0-L

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	Windows (Japanese version) Note	3.5-inch 2HD FD
AB13	IBM PC/AT™ and compatibles	Windows (Japanese version) Note	3.5-inch 2HC FD
BB13		Windows (English version) Note	
3P16	HP9000 series 700™	HP-UX™ (Rel. 10.10)	DAT (DDS)
3K13	SPARCstation™	SunOS™ (Rel. 4.1.4), Solaris (Rel. 2.5.1)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS™ (RISC)	NEWS-OS™ (Rel. 6.1)	3.5-inch 2HC FD

Note Can also be operated in DOS environment.

★ A.2 Flash Memory Writing Tools

Flashpro II (type FL-PR2) Flashpro III (type FL-PR3, PG-FP3) Flash programmer	Flash programmer dedicated to microcontrollers with on-chip flash memory.
FA-80GC**Note** Flash memory writing adapter	Flash memory writing adapter used connected to the Flashpro II or Flashpro III. 80-pin plastic QFP (GC-8BT type)

Note Under development

Remark FL-PR2, FL-PR3 and FA-80GC are products of NAITO DENSEI MACHIDA MFG. CO., LTD.
(TEL: +81-44-822-3813)

A.3 Debugging Tools

A.3.1 Hardware (1/2)

(1) When using the in-circuit emulator IE-78K0-NS

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 corresponds to integrated debugger (ID78K0-NS). This emulator should be used in combination with power supply unit, emulation probe, and interface adapter which is required to connect this emulator to the host machine.
★ IE-78K0-NS-PA[Note] Performance board	This board is used for extending the IE-78K0-NS functions, and is used connected to the IE-78K0-NS. 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-70000-MC-PS-B Power supply unit	This adapter is used for supplying power from a receptacle of 100-V to 240-V AC.
IE-70000-98-IF-C Interface adapter	This adapter is required when using the PC-9800 series PC (except notebook type) as the IE-78K0-NS host machine (C bus supported).
IE-70000-CD-IF-A PC card interface	This is PC card and interface cable required when using notebook PC as the IE-78K0-NS host machine (PCMCIA socket supported).
IE-70000-PC-IF-C Interface adapter	This adapter is required when using the IBM PC and compatibles as the IE-78K0-NS host machine (ISA bus supported).
★ IE-70000-PCI-IF Interface adapter	This adapter is required when using the PC with an on-chip PCI bus as the IE-78K0-NS host machine.
IE-780066-NS EM4[Note] Emulation board	This board emulates the operations of the peripheral hardware peculiar to a device. It should be used in combination with an in-circuit emulator and I/O board.
IE-78K0-NS-P01 I/O board	A board mounted with FPGA.
NP-80GC Emulation probe	This probe is used to connect the in-circuit emulator to the target system and is designed for 80-pin plastic QFP (GC-8BT type).
EV-9200GC-80 Conversion socket (Refer to **Figures A-2 and A-3**)	This conversion socket connects the NP-80GC to the target system board designed to mount a 80-pin plastic QFP (GC-8BT type).

Note Under development

Remarks 1. NP-80GC is a product of NAITO DENSEI MACHIDA MFG. CO., LTD.
(TEL: +81-44-822-3813)

2. EV-9200GC-80 is sold in five units.

A.3.1 Hardware (2/2)

(2) When using the in-circuit emulator IE-78001-R-A

IE-78001-R-A In-circuit emulator		The in-circuit emulator serves to debug hardware and software when developing application systems using a 78K/0 Series product. It corresponds to integrated debugger (ID78K0). This emulator should be used in combination with emulation probe and interface adapter, which is required to connect this emulator to the host machine.
IE-7000-98-IF-C Interface adapter		This adapter is required when using the PC-9800 series computer (except notebook type) as the IE-78001-R-A host machine (C bus supported).
IE-70000-PC-IF-C Interface adapter		This adapter is required when using the IBM PC/AT and compatibles as the IE-78001-R-A host machine (ISA bus supported).
★ IE-70000-PCI-IF Interface adapter		This adapter is required when using the PC with an on-chip PCI bus as the IE-78001-R-A host machine.
IE-78000-R-SV3 Interface adapter		This is adapter and cable required when using an EWS computer as the IE-78001-R-A host machine, and is used connected to the board in the IE-78000-R-A.
IE-780066-NS-EM4**Note** Emulation board		This board emulates the operations of the peripheral hardware peculiar to a device. It should be used in combination with an in-circuit emulator, I/O board, and emulation probe conversion board.
	IE-78K0-NS-P01 I/O board	A board mounted with FPGA
	IE-78K0-R-EX1 Emulation probe conversion board	This board is required when using the IE-780066-NS-EM4 + IE-78K0-NS-P01 on the IE-78001-R-A.
EP-78230GC-R Emulation probe		This probe is used to connect the in-circuit emulator to the target system and is designed for 80-pin plastic QFP (GC-8BT type).
	EV-9200GC-80 conversion socket (Refer to **Figures A-2 and A-3**)	This conversion socket connects the EP-78230GC-R to the target system board designed to mount a 80-pin plastic QFP (GC-8BT type).

Note Under development

Remark EV-9200GC-64 is sold in five units.

A.3.2 Software (1/2)

SM78K0 System Simulator	This system simulator is used to perform debugging at C source level or assembler level while simulating the operation of the target system on a host machine. This simulator runs on Windows. Use of the SM78K0 allows the execution of application logical testing and performance testing on an independent basis from hardware development without having to use an in-circuit emulator, thereby providing higher development efficiency and software quality. The SM78K0 should be used in combination with the optical device file (DF780066).
	Part Number: μS××××SM78K0

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SM78K0

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	Windows (Japanese version)	3.5-inch 2HD FD
AB13	IBM PC/AT and compatibles	Windows (Japanese version)	3.5-inch 2HC FD
BB13		Windows (English version)	

A.3.2 Software (2/2)

ID78K0-NS Integrated Debugger (supporting in-circuit emulator IE-78K0-NS)	This debugger is a control program to debug 78K/0 Series microcontrollers. It adopts a graphical user interface, which is equivalent visually and operationally to Windows or OSF/Motif™. It also has an enhanced debugging function for C language programs, and thus trace results can be displayed on screen in C-language level by using the windows integration function which links a trace result with its source program, disassembled display, and memory display. In addition, by incorporating function modules such as task debugger and system performance analyzer, the efficiency of debugging programs, which run on real-time OSs can be improved. It should be used in combination with the optional device file (DF7800066).
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××××ID78K0-NS

××××	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)	3.5-inch 2HC FD
BB13		Windows (English version)	

μS××××ID78K0

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	Windows (Japanese version)	3.5-inch 2HD FD
AB13	IBM PC/AT and its compatibles	Windows (Japanese version)	3.5-inch 2HC FD
BB13		Windows (English version)	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel. 4.1.4), Solaris (Rel. 2.5.1)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS™ (RISC)	NEWS-OS (Rel. 6.1)	3.5-inch 2HC FD

A.4 System Upgrade from Former In-circuit Emulator for 78K/0 Series to IE-78001-R-A

If you already have a former in-circuit emulator for 78K/0 Series microcontrollers (IE-78000-R or IE-78000-R-A), that in-circuit emulator can operate as an equivalent to the IE-78001-R-A by replacing its internal break board with the IE-78001-R-BK.

Table A-1. System-up Method from Former In-circuit Emulator for 78K/0 Series to the IE-78001-R-A

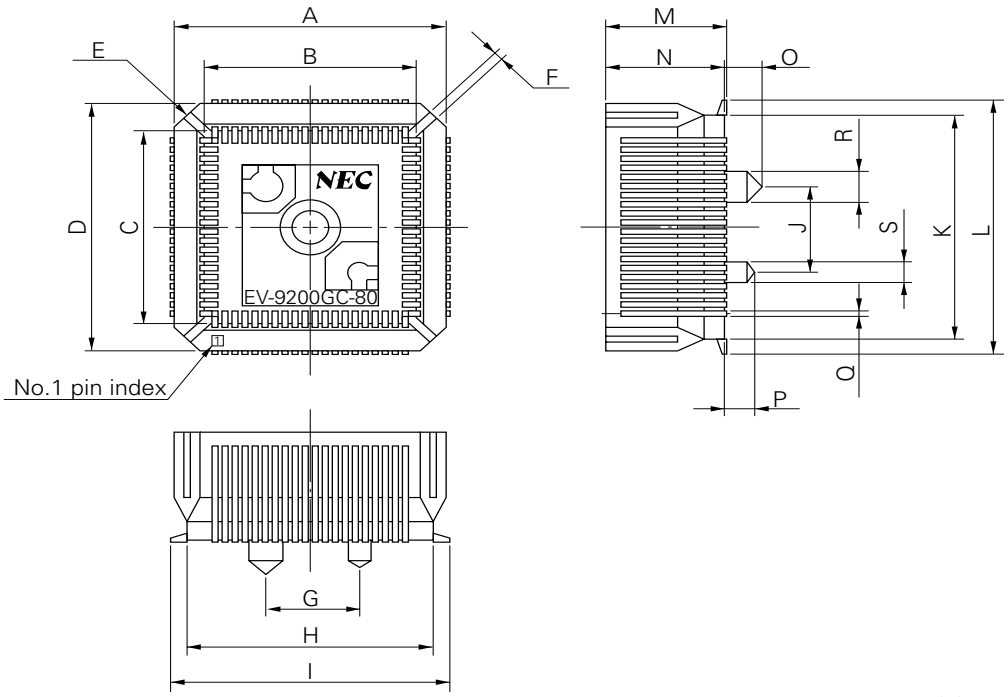
In-circuit Emulator Owned	In-circuit Emulator Cabinet System-up[Note]	Board to be Purchased
IE-78000-R	Required	IE-78001-R-BK
IE-78000-R-A	Not required	

Note For system-up of a cabinet, send your in-circuit emulator to NEC.

Conversion Socket Drawing (EV-9200GC-80) and Footprints

Figure A-2. EV-9200GC-80 Drawing (for reference only)

Based on EV-9200GC-80
(1) Package drawing (in mm)

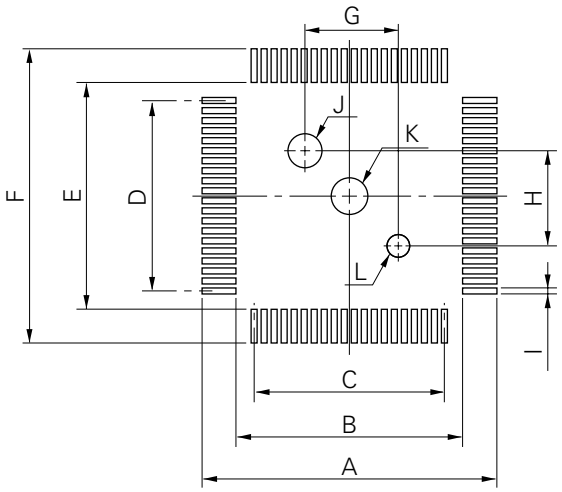


EV-9200GC-80-G1E

ITEM	MILLIMETERS	INCHES
A	18.0	0.709
B	14.4	0.567
C	14.4	0.567
D	18.0	0.709
E	4-C 2.0	4-C 0.079
F	0.8	0.031
G	6.0	0.236
H	16.0	0.63
I	18.7	0.736
J	6.0	0.236
K	16.0	0.63
L	18.7	0.736
M	8.2	0.323
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 A-3. EV-9200GC-80 Footprints (for reference only)

Based on EV-9200GC-80
(2) Pad drawing (in mm)



EV-9200GC-80-P1E

ITEM	MILLIMETERS	INCHES
A	19.7	0.776
B	15.0	0.591
C	$0.65\pm0.02\times19=12.35\pm0.05$	$0.026^{+0.001}_{-0.002}\times0.748=0.486^{+0.003}_{-0.002}$
D	$0.65\pm0.02\times19=12.35\pm0.05$	$0.026^{+0.001}_{-0.002}\times0.748=0.486^{+0.003}_{-0.002}$
E	15.0	0.591
F	19.7	0.776
G	6.0 ± 0.05	$0.236^{+0.003}_{-0.002}$
H	6.0 ± 0.05	$0.236^{+0.003}_{-0.002}$
I	0.35 ± 0.02	$0.014^{+0.001}_{-0.001}$
J	$\phi2.36\pm0.03$	$\phi0.093^{+0.001}_{-0.002}$
K	$\phi2.3$	$\phi0.091$
L	$\phi1.57\pm0.03$	$\phi0.062^{+0.001}_{-0.002}$

Caution Dimensions of mount pad for EV-9200 and that for target device (QFP) may be different in some parts. For the recommended mount pad dimensions for QFP, refer to "SEMICONDUCTOR DEVICE MOUNTING TECHNOLOGY MANUAL" (C10535E).

APPENDIX B EMBEDDED SOFTWARE

For efficient development and maintenance of the μPD780065 Subseries, the following embedded products are available.

Real-Time OS (1/2)

RX78K/0 Real-time OS	RX 78K/0 is a real-time OS conforming to the μITRON specifications. Tool (configurator) for generating nucleus of RX78K/0 and plural information tables is supplied. Used in combination with an optional assembler package (RA78/0) and device file (DF780066). **<Precaution when using RX78K/0 in PC environment>** The real-time OS is a DOS-based application. It should be used in the DOS Prompt when using in Windows.
	Part number: μS××××RX78013-ΔΔΔΔ

Caution When purchasing the RX78K/0, fill in the purchase application form in advance and sign the user agreement.

Remark ×××× and 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)[Note]	3.5-inch 2HD FD
AB13	IBM PC/AT and compatibles	Windows (Japanese version)[Note]	3.5-inch 2HC FD
BB13		Windows (English version)[Note]	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel. 4.1.4), Solaris (Rel. 2.5.1)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS (RISC)	NEWS-OS (Rel. 6.1)	3.5-inch 2HC FD

Note Can also be operated in DOS environment.

Real-Time OS (2/2)

MX78K0	MX78K/0 is an OS for μITRON specification subsets. A nucleus for the MX78K/0 is OS also included as a companion product. This manages tasks, events, and time. In the task management, determining the task execution order and switching from task to the next task are performed. **<Precaution when using MX78K/0 in PC environment>** The MX78K/0 is a DOS-based application. It should be used in the DOS Prompt when using in Windows.
	Part number: μS××××MX78K0-ΔΔΔ

Remark ×××× and ΔΔΔ in the part number differ depending on the host machine and OS used.

μS××××MX78K0-ΔΔΔ

ΔΔΔ	Product Outline	Maximum Number for Use in Mass Production
001	Evaluation object	Use in preproduction stages.
××	Mass-production object	Use in mass production stages.
S01	Source program	Only the users who purchased mass-production objects are allowed to purchase this program.

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	Windows (Japanese version)[Note]	3.5-inch 2HD FD
AB13	IBM PC/AT and compatibles	Windows (Japanese version)[Note]	3.5-inch 2HC FD
BB13		Windows (English version)[Note]	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel. 4.1.4), Solaris (Rel. 2.5.1)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS (RISC)	NEWS-OS (Rel. 6.1)	3.5-inch 2HC FD

Note Can also be operated in DOS environment.

[MEMO]

APPENDIX C REGISTER INDEX

C.1 Register Index (In Alphabetical Order with Respect to Register Names)

[P]

[R]

[S]

[T]

[W]

C.2 Register Index (In Alphabetical Order with Respect to Register Symbol)

[P]

[R]

[S]

[T]

[W]

★ APPENDIX D REVISION HISTORY

The following shows major revisions up to now.

Edition	Major Revisions from Previous Edition	Revised Chapters
2nd	Modification of description of on-chip pull-up resistor specification	CHAPTER 2 PIN FUNCTION
	Modification of description of TI00 pin	
	Addition of input/output circuit type of each pin and pin input/output circuit figures	
	Modification of caution on register initial setting for memory space	CHAPTER 3 CPU ARCHITECTURE
	Modification of register symbols ADTC → ADTC0, ADTP → ADTP0, ADTI → ADTI0	
	Modification of setting of on-chip pull-up resistor to not depend on input/output mode	CHAPTER 4 PORT FUNCTION
	Addition of PPG output column and modification of number of interval timers	CHAPTER 6 16-BIT TIMER/EVENT COUNTER
	Addition of clear by OSPT bit in block diagram of TM0	
	Addition of CR01 column in table of pin valid edges and capture triggers	
	Addition of OSPT bit caution	
	Addition and modification of cautions on external clock and capture trigger (addition of description of sampling clock for noise elimination)	
	Modification of input buffer to schmitt triggered input in block diagram of UART	CHAPTER 13 SERIAL INTERFACE (UART0)
	Modification of register symbols and bit names • Automatic data transmit/receive address pointer (ADTP0) • Automatic data transmit/receive control register (ADTC0) • Automatic data transmit/receive interval register (ADTI0) • Bit name of serial operation mode register 1 (CSM1)	CHAPTER 14 SERIAL INTERFACE (SIO1)
	Deletion of direction control circuit in block diagrams of SIO30, SIO31	CHAPTER 15 SERIAL INTERFACE (SIO30) CHAPTER 16 SERIAL INTERFACE (SIO31)
	Addition of description of capture register specification to INTTM00, INTTTM01 triggers	CHAPTER 17 INTERRUPT FUNCTIONS
	Modification of interrupt flag name (WTPR → WTPR0)	
	Addition of a caution on initial setting of memory size switching register (IMS)	CHAPTER 21 μPD78F0066
	Addition of Flashpro III as a flash writing tool	
	Addition of description of Windows for supporting PC98-NX series, modification of supported OS version, addition of Solaris to OSs, addition of performance board IE-78K0-NS-PA, and interface adapter IE-70000-PCI-IF	APPENDIX A DEVELOPMENT TOOLS

[MEMO]

NEC

Facsimile Message

From:

Name

Company

Tel. FAX

Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: 1-800-729-9288
1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Technical Documentation Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6465-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: 02-528-4411

Taiwan
NEC Electronics Taiwan Ltd.
Fax: 02-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: 044-548-7900

I would like to report the following error/make the following suggestion:

Document title: _____

Document number: _____ Page number: _____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	❑	❑	❑	❑
Technical Accuracy	❑	❑	❑	❑
Organization	❑	❑	❑	❑