NEC

Preliminary User's Manual

μPD780058, 780058Y Subseries

8-bit Single-chip Microcontrollers

μPD780053	μPD780053Y
μPD780054	μPD780054Y
μPD780055	μPD780055Y
μPD780056	μPD780056Y
μPD780058	μPD780058Y
μPD78F0058	μPD78F0058Y

Document No. U12013EJ2V0UM00 (2nd edition)
Date Published February 1998 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: μPD78F0058GC-3B9, 78F0058GC-8BT, 78F0058GK-BE9
μPD78F0058YGC-3B9, 78F0058YGC-8BT, 78F0058YGK-BE9

The customer must judge the need for license:
μPD780053GC-×××-3B9, 780053GC-×××-8BT, 780053GK-×××-BE9
μPD780053YGC-×××-3B9, 780053YGC-×××-8BT, 780053YGK-×××-BE9
μPD780054GC-×××-3B9, 780054GC-×××-8BT, 780054GK-×××-BE9
μPD780054YGC-×××-3B9, 780054YGC-×××-8BT, 780054YGK-×××-BE9
μPD780055GC-×××-3B9, 780055GC-×××-8BT, 780055GK-×××-BE9
μPD780055YGC-×××-3B9, 780055YGC-×××-8BT, 780055YGK-×××-BE9
μPD780056GC-×××-3B9, 780056GC-×××-8BT, 780056GK-×××-BE9
μPD780056YGC-×××-3B9, 780056YGC-×××-8BT, 780056YGK-×××-BE9
μPD780058GC-×××-3B9, 780058GC-×××-8BT, 780058GK-×××-BE9
μPD780058YGC-×××-3B9, 780058YGC-×××-8BT, 780058YGK-×××-BE9

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

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

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.

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

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

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

NEC devices are classified into the following three quality grades:
"Standard", "Special", and "Specific". The Specific quality grade applies only to devices developed based on a customer designated "quality assurance program" for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device before using it in a particular application.

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

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

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

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

Anti-radioactive design is not implemented in this product.

M5 96.5

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.1.
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

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

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

NEC Electronics (France) S.A.
Spain Office
Madrid, Spain
Tel: 01-504-2787
Fax: 01-504-2860

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

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

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

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

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-719-2377
Fax: 02-719-5951

NEC do Brasil S.A.
Cumbica-Guarulhos-SP, Brasil
Tel: 011-6465-6810
Fax: 011-6465-6829

J97. 8

Main Revisions in This Edition

Page	Description
pp. 127 to 131, 137	Change of following block diagrams of ports: **Figures 6-5** and **6-7 P20, P21, and P23 to P26 Block Diagram**, **Figure 6-6** and **6-8 P22 and P27 Block diagram**, **Figure 6-9 P30 to P37 Block Diagram**, and **Figure 6-16 P71 and P72 Block Diagram**
pp. 153	Addition of **Table 7-2 Relationships between CPU Clock and Minimum Instruction Execution Time**
pp. 226, 231	Addition of **Figures 9-10** and **9-13 Square Wave Output Operation Timing**
pp. 291	Connection of Note on BSYE in **Figure 16-5 Serial Bus Interface Control Register Format**
pp. 302	Addition of Caution to **16.4.3 (2) (a) Bus release signal (REL)** and **(b) Command signal (CMD)**
pp. 394	Addition of **(3) MSB/LSB switching as the start bit** to **18.4.2 3-wire serial I/O mode operation**
pp. 416 to 420	Change of **18.4.3 (3) (d) Busy control option**, **(e) Busy & strobe control option**, and **(f) Bit shippage detection function** in old edition to **(4) Synchronization control**, and improvement of explanation
pp. 454	Correction of **Figure 19-11 Receive Error Timing**
pp. 461	Addition of **(3) MSB/LSB switching as the start bit** to **19.4.3 3-wire serial I/O mode**
pp. 463	Addition of **19.4.4 Restrictions in UART mode**
pp. 536	Addition of Note to **26.1 Memory Size Switching Register**
pp. 538	**26.3 Flash Memory Programming** Change of product name of flash programmer from Flashpro to Flashpro II
pp. 559	Addition of **APPENDIX A DIFFERENCES AMONG μPD78054, 78058F, AND 780058 SUBSERIES**
pp. 561 to 573	**APPENDIX B DEVELOPMENT TOOLS** Total revision: Support of in-circuit emulators IE-78K0-NS and IE-78001-R-A
pp. 575 to 576	**APPENDIX C EMBEDDED SOFTWARE** Total revision: Deletion of fuzzy inference development support system
pp. 583	Addition of **APPENDIX E REVISION HISTORY**

The mark ★ shows major revised points.

PREFACE

Readers

This manual has been prepared for user engineers who want to understand the functions of the μPD780058 and 780058Y Subseries and design and develop its application systems and programs.

Purpose

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

Organization

The μPD780058, 780058Y Subseries manual is separated into two parts: this manual and the instruction edition (common to the 78K/0 Series).

μPD780058, 780058Y Subseries User's Manual (This manual)	78K/0 Series User's Manual Instructions
● Pin functions	● CPU functions
● Internal block functions	● Instruction set
● Interrupt	● Explanation of each instruction
● Other on-chip peripheral functions	

How to Read This Manual

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

○ When you want to understand the functions in general:
→ Read this manual in the order of the contents.

○ To know the μPD780058 and 780058Y Subseries instruction function in detail:
→ Refer to the **78K/0 Series User's Manual — Instructions (U12326E)**

○ How to interpret the register format:
→ For the circled bit number, the bit name is defined as a reserved word in RA78K/0, and in CC78K/0, already defined in the header file named sfrbit.h.

○ To learn the function of a register whose register name is known:
→ Refer to **APPENDIX D REGISTER INDEX**.

○ To know the application examples of each function of the μPD780058, 780058Y Subseries:
→ Refer to **78K/0 Series Application Note — Basics (III) (U10182E)** separately available.

Chapter Organization: This manual divides the descriptions for the μPD780058 and 780058Y Subseries into different chapters as shown below. Read only the chapters related to the device you use.

Chapter		μPD780058 Subseries	μPD780058Y Subseries
Chapter 1	Outline (μPD780058 Subseries)	√	—
Chapter 2	Outline (μPD780058Y Subseries)	—	√
Chapter 3	Pin Function (μPD780058 Subseries)	√	—
Chapter 4	Pin Function (μPD780058Y Subseries)	—	√
Chapter 5	CPU Architecture	√	√
Chapter 6	Port Functions	√	√
Chapter 7	Clock Generator	√	√
Chapter 8	16-Bit Timer/Event Counter	√	√
Chapter 9	8-Bit Timer/Event Counters	√	√
Chapter 10	Watch Timer	√	√
Chapter 11	Watchdog Timer	√	√
Chapter 12	Clock Output Control Circuit	√	√
Chapter 13	Buzzer Output Control Circuit	√	√
Chapter 14	A/D Converter	√	√
Chapter 15	D/A Converter	√	√
Chapter 16	Serial Interface Channel 0 (μPD780058 Subseries)	√	—
Chapter 17	Serial Interface Channel 0 (μPD780058Y Subseries)	—	√
Chapter 18	Serial Interface Channel 1	√	√
Chapter 19	Serial Interface Channel 2	√	√
Chapter 20	Real-Time Output Port	√	√
Chapter 21	Interrupt and Test Functions	√	√
Chapter 22	External Device Expansion Function	√	√
Chapter 23	Standby Function	√	√
Chapter 24	Reset Function	√	√
Chapter 25	ROM Correction	√	√
Chapter 26	μPD78F0058, μPD78F0058Y	√	√
Chapter 27	Outline of Instruction Set	√	√

Differences Between μPD780058 and μPD780058Y Subseries:

The μPD780058 and μPD780058Y Subseries are different in the following functions of the serial interface channel 0.

Modes of serial interface channel 0	μPD780058 Subseries	μPD780058Y Subseries
3-wire serial I/O mode	√	√
2-wire serial I/O mode	√	√
SBI (serial bus interface) mode	√	—
I^{2}C (inter IC) bus mode	—	√

√ : Supported

— : Not supported

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

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

● Documents Related to Devices

Document name	Document No.	
	English	Japanese
μPD780058, 780058Y Subseries User's Manual	This manual	U12013J
μPD780053, 780054, 780055, 780056, 780058 Preliminary Product Information	U12182E	U12182J
μPD78F0058 Preliminary Product Information	U12092E	U12092J
μPD780053Y, 780054Y, 780055Y, 780056Y, 780058Y Preliminary Product Information	U12328E	U12328J
μPD78F0058Y Preliminary Product Information	U12324E	U12324E
78K/0 Series User's Manual — Instructions	U12326E	U12326J
78K/0 Series Instruction Table	—	U10903J
78K/0 Series Instruction Set	—	U10904J
μPD780058 Subseries Special Function Register Table	—	To be prepared
μPD780058Y Subseries Special Function Register Table	—	To be prepared
★ 78K/0 Series Application Note — Basics (III)	U10182E	U10182J

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

● **Development Tool Documents (User’s Manuals)**

Document name		Document No.	
		English	Japanese
★ RA78K0 Assembler Package	Operation	U11802E	U11802J
★	Assembly Language	U11801E	U11801J
★	Structured Assembly Language	U11789E	U11789J
RA78K Series Structured Assembler Preprocessor		EEU-1402	U12323J
CC78K/0 C Compiler	Operation	U11517E	U11517J
	Language	U11518E	U11518J
CC78K/0 C Compiler Application Note	Programming know-how	EEA-1208	EEA-618
CC78K Series Library Source File		—	U12322J
★ 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-780308-NS-EM1		To be prepared	To be prepared
EP-780058GC-R		To be prepared	To be prepared
EP-780058GK-R		To be prepared	To be prepared
SM78K0 System Simulator Windows™ Base	Reference	U10181E	U10181J
SM78K Series System Simulator	External component user open interface specifications	U10092E	U10092J
ID78K0 Integrated Debugger EWS Base	Reference	—	U11151J
ID78K0 Integrated Debugger PC Base	Reference	U11539E	U11539J
ID78K0 Integrated Debugger Windows Base	Guide	U11649E	U11649J
★ ID78K0-NS Integrated Debugger PC Base	Reference	To be prepared	U12900J

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

● **Documents for Embedded Software (User’s Manuals)**

Document name		Document No.	
		English	Japanese
78K/0 Series Real-Time OS RX78K/0	Fundamental	U11537E	U11537J
	Installation	U11536E	U11536J
78K/0 Series OS MX78K0	Fundamental	U12257E	U12257J

● **Other Documents**

Document name	Document No.	
	English	Japanese
IC PACKAGE MANUAL	C10943X	
Semiconductor Device Mounting Technology Manual	C10535E	C10535J
Quality Grades on NEC Semiconductor Devices	C11531E	C11531J
NEC Semiconductor Device Reliability/Quality Control System	C10983E	C10983J
★ Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E	C11892J
Guide to Quality Assurance for Semiconductor Devices	MEI-1202	—
Microcomputer Product Series Guide	—	U11416J

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

[MEMO]

CONTENTS

LIST OF FIGURES (1/8)

LIST OF FIGURES (2/8)

LIST OF FIGURES (3/8)

LIST OF FIGURES (4/8)

LIST OF FIGURES (5/8)

LIST OF FIGURES (6/8)

LIST OF FIGURES (7/8)

LIST OF FIGURES (8/8)

LIST OF TABLES (1/3)

LIST OF TABLES (2/3)

LIST OF TABLES (3/3)

CHAPTER 1 OUTLINE (μPD780058 Subseries)

1.1 Features

- On-chip high-capacity ROM and RAM

Item / Part Number	Program Memory		Data Memory		
	Mask ROM	Flash Memory	Internal High-Speed RAM	Internal Buffer RAM	Internal Expansion RAM
μPD780053	24 Kbytes	—	1,024 bytes	32 bytes	None
μPD780054	32 Kbytes	—			
μPD780055	40 Kbytes	—			
μPD780056	48 Kbytes	—			
μPD780058	60 Kbytes	—			1,024 bytes
μPD78F0058	—	60 Kbytes Note 1			1,024 bytes Note 2

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

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

- External memory expansion space: 64 Kbytes
- Minimum instruction execution time changeable from high-speed (0.4 μs: In main system clock 5.0-MHz operation) to ultra-low speed (122 μs: In subsystem clock 32.768-kHz operation)
- Minimum instruction set suited to system control
  - Bit manipulation possible in all address spaces
  - Multiple and divide instructions
- I/O ports: 68 (N-ch open-drain: 4)
- 8-bit resolution A/D converter: 8 channels
- 8-bit resolution D/A converter: 2 channels
- Serial interface: 3 channels
  - 3-wire serial I/O/SBI/2-wire serial I/O mode: 1 channel
  - 3-wire serial I/O mode (On-chip automatic transmit/receive function): 1 channel
  - 3-wire serial I/O/UART mode (On-chip time-division transfer function): 1 channel
- Timer: 5 channels
  - 16-bit timer/event counter: 1 channel
  - 8-bit timer/event counter: 2 channels
  - Watch timer: 1 channel
  - Watchdog timer: 1 channel
- Vectored interrupt sources: 21
- Test inputs: 2
- On-chip two types of clock oscillators (main system clock and subsystem clock)
- Supply voltage: V_{DD} = 1.8 to 5.5 V

1.2 Applications

Car audio systems, cellular phones, pagers, printers, AV equipment, cameras, PPCs, vending machines, etc.

1.3 Ordering Information

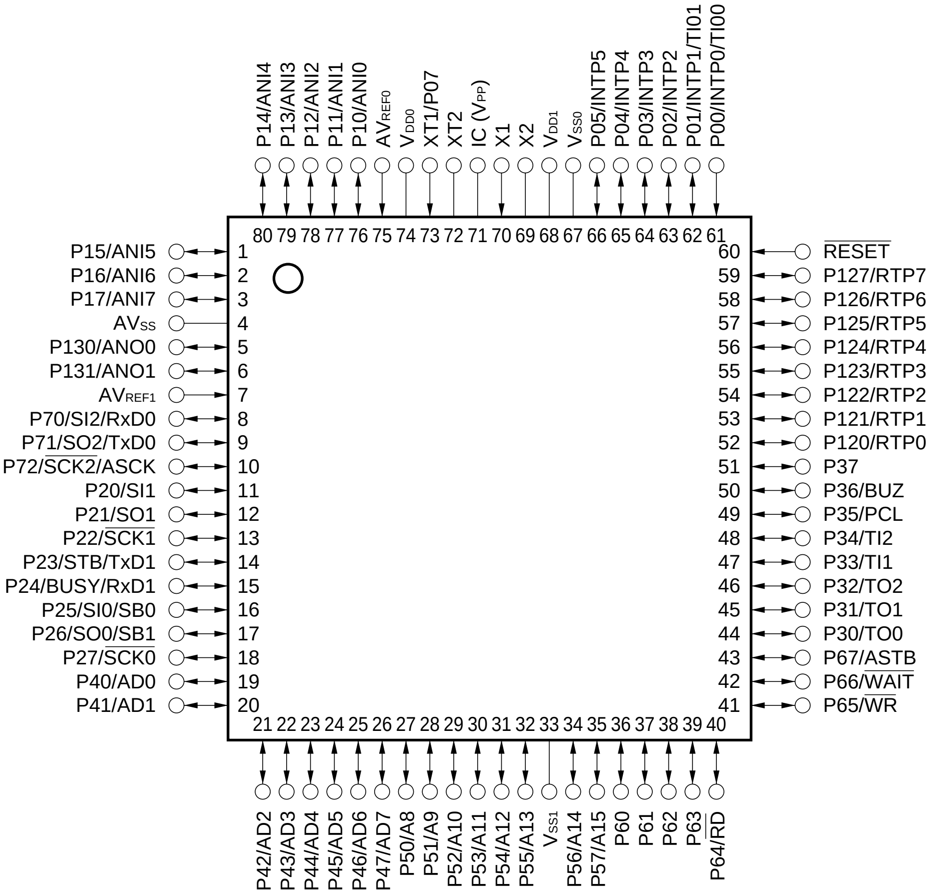
Part Number	Package	Internal ROM
μPD780053GC-××× -3B9	80-pin plastic QFP (14 × 14 mm, Resin thickness: 2.7 mm)	Mask ROM
μPD780053GC-×××-8BT	80-pin plastic QFP (14 × 14 mm, Resin thickness: 1.4 mm)	Mask ROM
μPD780053GK-×××-BE9	80-pin plastic TQFP (Fine pitch) (12 × 12 mm)	Mask ROM
μPD780054GC-×××-3B9	80-pin plastic QFP (14 × 14 mm, Resin thickness: 2.7 mm)	Mask ROM
μPD780054GC-×××-8BT	80-pin plastic QFP (14 × 14 mm, Resin thickness: 1.4 mm)	Mask ROM
μPD780054GK-×××-BE9	80-pin plastic TQFP (Fine pitch) (12 × 12 mm)	Mask ROM
μPD780055GC-×××-3B9	80-pin plastic QFP (14 × 14 mm, Resin thickness: 2.7 mm)	Mask ROM
μPD780055GC-×××-8BT	80-pin plastic QFP (14 × 14 mm, Resin thickness: 1.4 mm)	Mask ROM
μPD780055GK-×××-BE9	80-pin plastic TQFP (Fine pitch) (12 × 12 mm)	Mask ROM
μPD780056GC-×××-3B9	80-pin plastic QFP (14 × 14 mm, Resin thickness: 2.7 mm)	Mask ROM
μPD780056GC-×××-8BT	80-pin plastic QFP (14 × 14 mm, Resin thickness: 1.4 mm)	Mask ROM
μPD780056GK-×××-BE9	80-pin plastic TQFP (Fine pitch) (12 × 12 mm)	Mask ROM
μPD780058GC-×××-3B9	80-pin plastic QFP (14 × 14 mm, Resin thickness: 2.7 mm)	Mask ROM
μPD780058GC-×××-8BT	80-pin plastic QFP (14 × 14 mm, Resin thickness: 1.4 mm)	Mask ROM
μPD780058GK-×××-BE9	80-pin plastic TQFP (Fine pitch) (12 × 12 mm)	Mask ROM
μPD78F0058GC-3B9	80-pin plastic QFP (14 × 14 mm, Resin thickness: 2.7 mm)	Flash memory
μPD78F0058GC-8BT	80-pin plastic QFP (14 × 14 mm, Resin thickness: 1.4 mm)	Flash memory
μPD78F0058GK-BE9	80-pin plastic TQFP (Fine pitch) (12 × 12 mm)	Flash memory

Caution The μPD780053GC, 780054GC, 780055GC, 780056GC, 780058GC, and 78F0058GC are available in two packages. For the package that can be supplied, consult an NEC sales representative.

Remark ××× indicates ROM code suffix.

1.4 Pin Configuration (Top View)

- **80-pin plastic QFP (14 × 14 mm, Resin thickness: 2.7 mm)**
  μPD780053GC-×××-3B9, 780054GC-×××-3B9, 780055GC-×××-3B9,
  μPD780056GC-×××-3B9, 780058GC-×××-3B9, 78F0058GC-3B9
- **80-pin plastic QFP (14 × 14 mm, Resin thickness: 1.4 mm)**
  μPD780053GC-×××-8BT, 780054GC-×××-8BT, 780055GC-×××-8BT,
  μPD780056GC-×××-8BT, 780058GC-×××-8BT, 78F0058GC-8BT
- **80-pin plastic TQFP (Fine pitch) (12 × 12 mm)**
  μPD780053GK-×××-BE9, 780054GK-×××-BE9, 780055GK-×××-BE9,
  μPD780056GK-×××-BE9, 780058GK-×××-BE9, 78F0058GK-BE9



Cautions 1. Be sure to connect IC (Internally Connected) pin to V_{SS0} directly in normal operating mode.

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

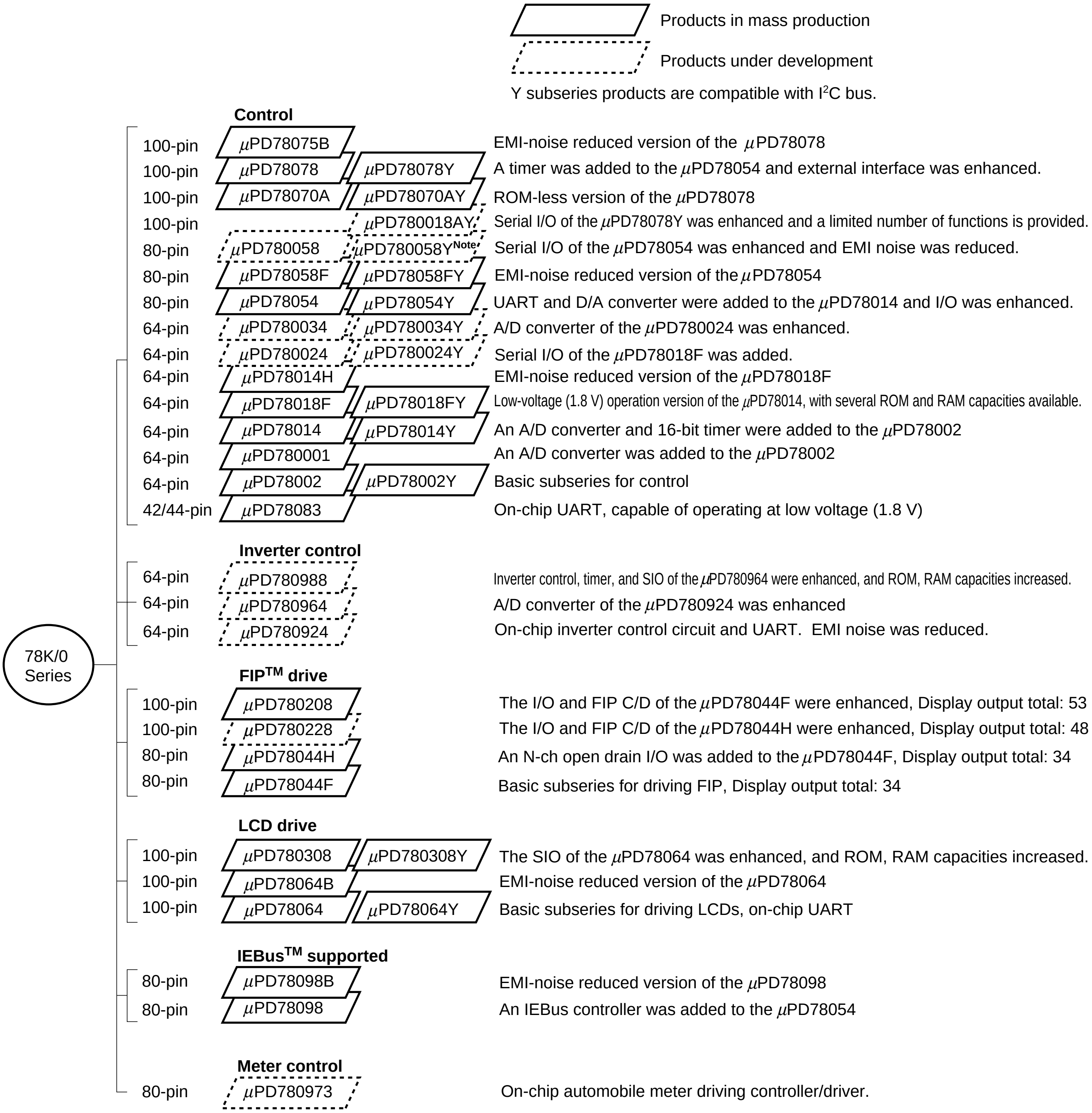
Remarks 1. Pin connection in parentheses is intended for the μPD78F0058.

2. When the μPD780053, 780054, 780055, 780056, or 780058 is used in application fields that require reduction of the noise generated from inside the microcontroller, the implementation of noise reduction measures, such as supplying to V_{DD0} and V_{DD1} individually and connecting V_{SS0} and V_{SS1} to different ground lines, is recommended.

A8 to A15	:	Address Bus	PCL	:	Programmable Clock
AD0 to AD7	:	Address/Data Bus	$\overline{\text{RD}}$	:	Read Strobe
ANI0 to ANI7	:	Analog Input	$\overline{\text{RESET}}$	:	Reset
ANO0, ANO1	:	Analog Output	RTP0 to RTP7	:	Real-Time Output Port
ASCK	:	Asynchronous Serial Clock	RxD0, RxD1	:	Receive Data
ASTB	:	Address Strobe	SB0, SB1	:	Serial Bus
AV_{REF0}, AV_{REF1}	:	Analog Reference Voltage	$\overline{\text{SCK0}}$ to $\overline{\text{SCK2}}$	:	Serial Clock
AV_{SS}	:	Analog Ground	SI0 to SI2	:	Serial Input
BUSY	:	Busy	SO0 to SO2	:	Serial Output
BUZ	:	Buzzer Clock	STB	:	Strobe
IC	:	Internally Connected	TI00, TI01	:	Timer Input
INTP0 to INTP6	:	Interrupt from Peripherals	TI1, TI2	:	Timer Input
P00 to P05, P07	:	Port 0	TO0 to TO2	:	Timer Output
P10 to P17	:	Port 1	TxD0, TxD1	:	Transmit Data
P20 to P27	:	Port 2	V_{DD0}, V_{DD1}	:	Power Supply
P30 to P37	:	Port 3	V_{PP}	:	Programming Power Supply
P40 to P47	:	Port 4	V_{SS0}, V_{SS1}	:	Ground
P50 to P57	:	Port 5	$\overline{\text{WAIT}}$	:	Wait
P60 to P67	:	Port 6	$\overline{\text{WR}}$	:	Write Strobe
P70 to P72	:	Port 7	X1, X2	:	Crystal (Main System Clock)
P120 to P127	:	Port 12	XT1, XT2	:	Crystal (Subsystem Clock)
P130, P131	:	Port 13			

★ 1.5 78K/0 Series Line-up

78K/0 Series product line-up is illustrated below. Part numbers in the boxes indicate subseries names.



Y subseries products are compatible with I²C bus.

Control

100-pin	μPD78075B		EMI-noise reduced version of the μPD78078
100-pin	μPD78078	μPD78078Y	A timer was added to the μPD78054 and external interface was enhanced.
100-pin	μPD78070A	μPD78070AY	ROM-less version of the μPD78078
100-pin		μPD780018AY	Serial I/O of the μPD78078Y was enhanced and a limited number of functions is provided.
80-pin	μPD780058	μPD780058Y^Note	Serial I/O of the μPD78054 was enhanced and EMI noise was reduced.
80-pin	μPD78058F	μPD78058FY	EMI-noise reduced version of the μPD78054
80-pin	μPD78054	μPD78054Y	UART and D/A converter were added to the μPD78014 and I/O was enhanced.
64-pin	μPD780034	μPD780034Y	A/D converter of the μPD780024 was enhanced.
64-pin	μPD780024	μPD780024Y	Serial I/O of the μPD78018F was added.
64-pin	μPD78014H		EMI-noise reduced version of the μPD78018F
64-pin	μPD78018F	μPD78018FY	Low-voltage (1.8 V) operation version of the μPD78014, with several ROM and RAM capacities available.
64-pin	μPD78014	μPD78014Y	An A/D converter and 16-bit timer were added to the μPD78002
64-pin	μPD780001		An A/D converter was added to the μPD78002
64-pin	μPD78002	μPD78002Y	Basic subseries for control
42/44-pin	μPD78083		On-chip UART, capable of operating at low voltage (1.8 V)

Inverter control

64-pin	μPD780988	Inverter control, timer, and SIO of the μPD780964 were enhanced, and ROM, RAM capacities increased.
64-pin	μPD780964	A/D converter of the μPD780924 was enhanced
64-pin	μPD780924	On-chip inverter control circuit and UART. EMI noise was reduced.

FIP™ drive

100-pin	μPD780208	The I/O and FIP C/D of the μPD78044F were enhanced, Display output total: 53
100-pin	μPD780228	The I/O and FIP C/D of the μPD78044H were enhanced, Display output total: 48
80-pin	μPD78044H	An N-ch open drain I/O was added to the μPD78044F, Display output total: 34
80-pin	μPD78044F	Basic subseries for driving FIP, Display output total: 34

LCD drive

100-pin	μPD780308	μPD780308Y	The SIO of the μPD78064 was enhanced, and ROM, RAM capacities increased.
100-pin	μPD78064B		EMI-noise reduced version of the μPD78064
100-pin	μPD78064	μPD78064Y	Basic subseries for driving LCDs, on-chip UART

IEBus™ supported

80-pin	μPD78098B	EMI-noise reduced version of the μPD78098
80-pin	μPD78098	An IEBus controller was added to the μPD78054

Meter control

80-pin	μPD780973	On-chip automobile meter driving controller/driver.

Note Under planning

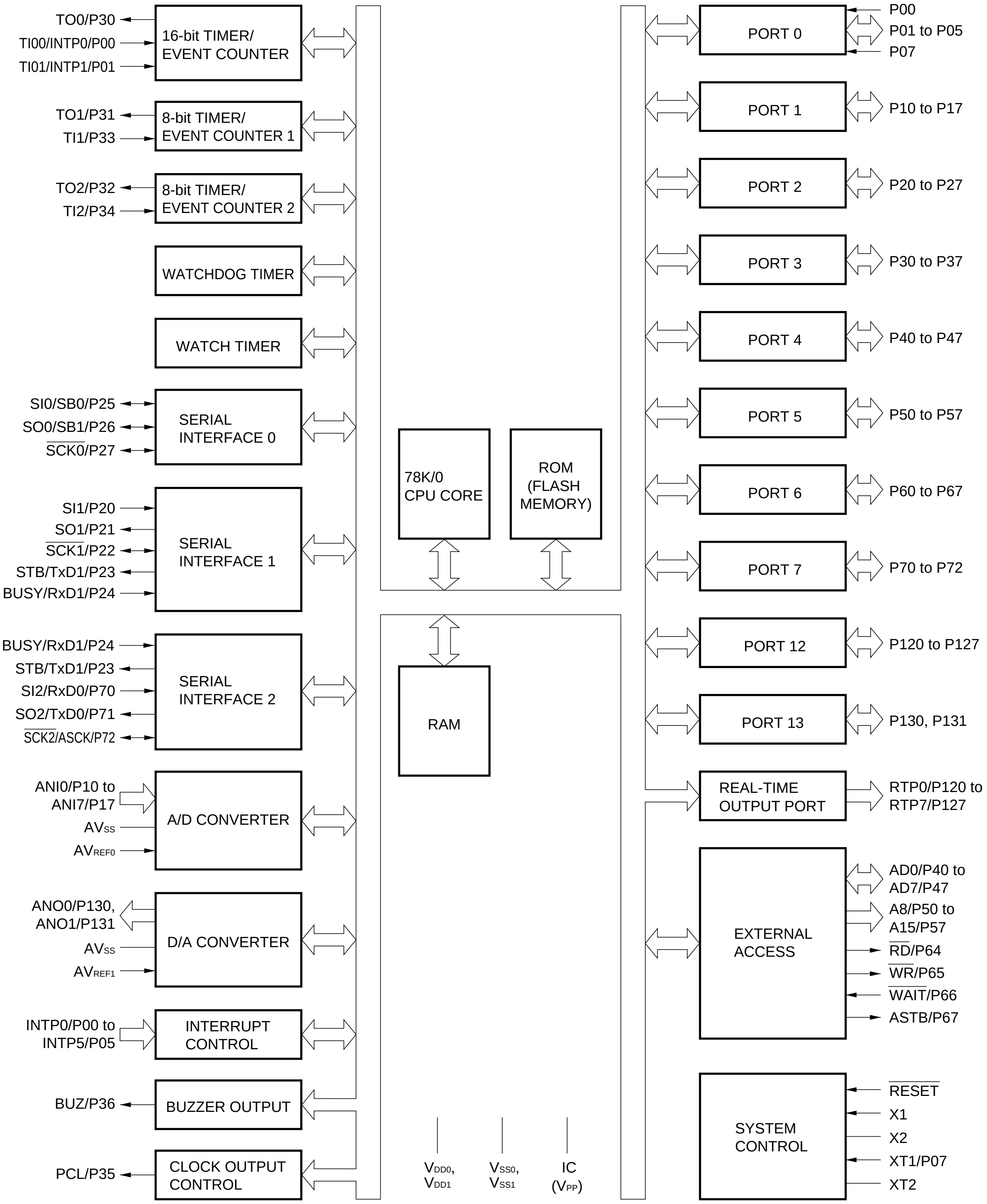
The following lists the main functional differences between subseries products.

Function Subseries Name		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	32K to 40K	4 ch	1 ch	1 ch	1 ch	8 ch	–	2 ch	3 ch (UART: 1 ch)	88	1.8 V	√
	µPD78078	48K to 60K											
	µPD78070A	–									61	2.7 V	
	µPD780058	24K to 60K	2 ch							3 ch (time division UART: 1 ch)	68	1.8 V	
	µPD78058F	48K to 60K								3 ch (UART: 1 ch)	69	2.7 V	
	µPD78054	16K to 60K										2.0 V	
	µPD780034	8K to 32K					–	8 ch	–	3 ch (UART: 1 ch, time division 3-wire: 1 ch)	51	1.8 V	
	µPD780024						8 ch	–					
	µPD78014H									2 ch	53		
	µPD78018F	8K to 60K											
	µPD78014	8K to 32K										2.7 V	
	µPD780001	8K		–	–					1 ch	39		–
	µPD78002	8K to 16K			1 ch		–				53		√
	µPD78083				–		8 ch			1 ch (UART: 1 ch)	33	1.8 V	–
Inverter control	µPD780988	32K to 60K	3 ch	Note 1	–	1 ch	–	8 ch	–	3 ch (UART: 2 ch)	47	4.0 V	√
	µPD780964	8K to 32K		Note 2						2 ch (UART: 2 ch)		2.7 V	
	µPD780924						8 ch	–					
FIP drive	µPD780208	32K to 60K	2 ch	1 ch	1 ch	1 ch	8 ch	–	–	2 ch	74	2.7 V	–
	µPD780228	48K to 60K	3 ch	–	–					1 ch	72	4.5 V	
	µPD78044H	32K to 48K	2 ch	1 ch	1 ch						68	2.7 V	
	µPD78044F	16K to 40K								2 ch			
LCD drive	µPD780308	48K to 60K	2 ch	1 ch	1 ch	1 ch	8 ch	–	–	3 ch (time division UART: 1 ch)	57	2.0 V	–
	µPD78064B	32K								2 ch (UART: 1 ch)			
	µPD78064	16K to 32K											
IEBus supported	µPD78098B	40K to 60K	2 ch	1 ch	1 ch	1 ch	8 ch	–	2 ch	3 ch (UART: 1 ch)	69	2.7 V	√
	µPD78098	32K to 60K											
Meter control	µPD780973	24K to 32K	3 ch	1 ch	1 ch	1 ch	5 ch	–	–	2 ch (UART: 1 ch)	56	4.5 V	–

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

2. 10-bit timer: 1 channel

1.6 Block Diagram



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

2. Pin connection in parentheses is intended for the μPD78F0058.

1.7 Outline of Function

Item	Part Number	μPD780053	μPD780054	μPD780055	μPD780056	μPD780058	μPD78F0058
Internal memory	ROM	Mask ROM					Flash memory
		24 Kbytes	32 Kbytes	40 Kbytes	48 Kbytes	60 Kbytes	60 Kbytes **Note 1**
	High-speed RAM	1,024 bytes					
	Buffer RAM	32 bytes					
	Expansion RAM	None				1,024 bytes	1,024 bytes **Note 2**
Memory space		64 Kbytes					
General register		8 bits × 8 × 4 banks					
Minimum instruction execution time	With main system clock selected	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (@ 5.0-MHz operation)					
	With subsystem clock selected	122 μs (@ 32.768-kHz operation)					
Instruction set		• 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		• Total : 68 • CMOS input : 2 • CMOS I/O : 62 • N-ch open-drain I/O : 4					
A/D converter		8-bit resolution × 8 channels					
D/A converter		8-bit resolution × 2 channels					
Serial interface		• 3-wire serial I/O/SBI/2-wire serial I/O mode selection possible : 1 channel • 3-wire serial I/O mode (on-chip max. 32 bytes auto-transmit/receive function) : 1 channel • 3-wire serial I/O/UART mode (on-chip time-division transfer function) selectable : 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		3: (14-bit PWM output enable: 1)					
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (main system clock: @ 5.0-MHz operation) 32.768 kHz (subsystem clock: @ 32.768-kHz operation)					
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (main system clock: @ 5.0-MHz operation)					

Notes **1.** The capacities of the flash memory can be changed by using the memory switching register (IMS).

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

Part Number Item		μPD780053	μPD780054	μPD780055	μPD780056	μPD780058	μPD78F0058
Vectored interrupt source	Maskable	Internal: 13 External: 6					
	Non-maskable	Internal: 1					
	Software	1					
Test input		Internal: 1 External: 1					
Supply voltage		V_{DD} = 1.8 to 5.5 V					
Operating ambient temperature		T_A = –40 to +85°C					
Package		• 80-pin plastic QFP (14 × 14 mm, Resin thickness : 2.7 mm) • 80-pin plastic QFP (14 × 14 mm, Resin thickness : 1.4 mm) • 80-pin plastic TQFP (Fine pitch) (12 × 12 mm)					

1.8 Mask Options

The mask ROM versions (μPD780053, 780054, 780055, 780056, 780058) provide pull-up resistor mask options which allow users to specify whether to connect a pull-up resistor to a specific port pin when the user places an order for the device production. Using this mask option when pull-up resistors are required reduces the number of components to add to the device, resulting in board space saving.

The mask options provided in the μPD780058 Subseries are shown in Table 1-1.

Table 1-1. Mask Options of Mask ROM Versions

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

CHAPTER 2 OUTLINE (μPD780058Y Subseries)

2.1 Features

- On-chip high-capacity ROM and RAM

Item Part Number	Program Memory		Data Memory		
	Mask ROM	Flash Memory	Internal High-Speed RAM	Internal Buffer RAM	Internal Expansion RAM
μPD780053Y	24 Kbytes	—	1,024 bytes	32 bytes	None
μPD780054Y	32 Kbytes	—			
μPD780055Y	40 Kbytes	—			
μPD780056Y	48 Kbytes	—			
μPD780058Y	60 Kbytes	—			1,024 bytes
μPD78F0058Y	—	60 Kbytes Note 1			1,024 bytes Note 2

Notes **1.** The capacities of flash memory can be changed by means of the memory size switching register (IMS).

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

- External memory expansion space: 64 Kbytes
- Minimum instruction execution time changeable from high-speed (0.4 μs: In main system clock 5.0-MHz operation) to ultra-low speed (122 μs: In subsystem clock 32.768-kHz operation)
- Instruction set suited to system control
  - Bit manipulation possible in all address spaces
  - Multiple and divide instructions
- I/O ports: 68 (N-ch open-drain: 4)
- 8-bit resolution A/D converter: 8 channels
- 8-bit resolution D/A converter: 2 channels
- Serial interface: 3 channels
  - 3-wire serial I/O/2-wire serial I/O/I^{2}C bus mode: 1 channel
  - 3-wire serial I/O mode (On-chip automatic transmit/receive function): 1 channel
  - 3-wire serial I/O/UART mode (On-chip time-division transfer function): 1 channel
- Timer: 5 channels
  - 16-bit timer/event counter: 1 channel
  - 8-bit timer/event counter: 2 channels
  - Watch timer: 1 channel
  - Watchdog timer: 1 channel
- Vectored interrupt sources: 21
- Test inputs: 2
- On-chip two types of clock oscillators (main system clock and subsystem clock)
- Supply voltage: V_{DD} = 1.8 to 5.5 V

2.2 Applications

Car audio systems, cellular phones, pagers, printers, AV equipment, cameras, PPCs, vending machines, etc.

2.3 Ordering Information

Part Number	Package	Internal ROM
μPD780053YGC-×××-3B9 Note	80-pin plastic QFP (14 × 14 mm, Resin thickness: 2.7 mm)	Mask ROM
μPD780053YGC-×××-8BT Note	80-pin plastic QFP (14 × 14 mm, Resin thickness: 1.4 mm)	Mask ROM
μPD780053YGK-×××-BE9 Note	80-pin plastic TQFP (Fine pitch) (12 × 12 mm)	Mask ROM
μPD780054YGC-×××-3B9 Note	80-pin plastic QFP (14 × 14 mm, Resin thickness: 2.7 mm)	Mask ROM
μPD780054YGC-×××-8BT Note	80-pin plastic QFP (14 × 14 mm, Resin thickness: 1.4 mm)	Mask ROM
μPD780054YGK-×××-BE9 Note	80-pin plastic TQFP (Fine pitch) (12 × 12 mm)	Mask ROM
μPD780055YGC-×××-3B9 Note	80-pin plastic QFP (14 × 14 mm, Resin thickness: 2.7 mm)	Mask ROM
μPD780055YGC-×××-8BT Note	80-pin plastic QFP (14 × 14 mm, Resin thickness: 1.4 mm)	Mask ROM
μPD780055YGK-×××-BE9 Note	80-pin plastic TQFP (Fine pitch) (12 × 12 mm)	Mask ROM
μPD780056YGC-×××-3B9 Note	80-pin plastic QFP (14 × 14 mm, Resin thickness: 2.7 mm)	Mask ROM
μPD780056YGC-×××-8BT Note	80-pin plastic QFP (14 × 14 mm, Resin thickness: 1.4 mm)	Mask ROM
μPD780056YGK-×××-BE9 Note	80-pin plastic TQFP (Fine pitch) (12 × 12 mm)	Mask ROM
μPD780058YGC-×××-3B9 Note	80-pin plastic QFP (14 × 14 mm, Resin thickness: 2.7 mm)	Mask ROM
μPD780058YGC-×××-8BT Note	80-pin plastic QFP (14 × 14 mm, Resin thickness: 1.4 mm)	Mask ROM
μPD780058YGK-×××-BE9 Note	80-pin plastic TQFP (Fine pitch) (12 × 12 mm)	Mask ROM
μPD78F0058YGC-3B9 Note	80-pin plastic QFP (14 × 14 mm, Resin thickness: 2.7 mm)	Flash-memory
μpD78F0058YGC-8BT Note	80-pin plastic QFP (14 × 14 mm, Resin thickness: 1.4 mm)	Flash-memory
μpD78F0058YGK-BE9 Note	80-pin plastic TQFP (Fine pitch) (12 × 12 mm)	Flash-memory

Note Under planning

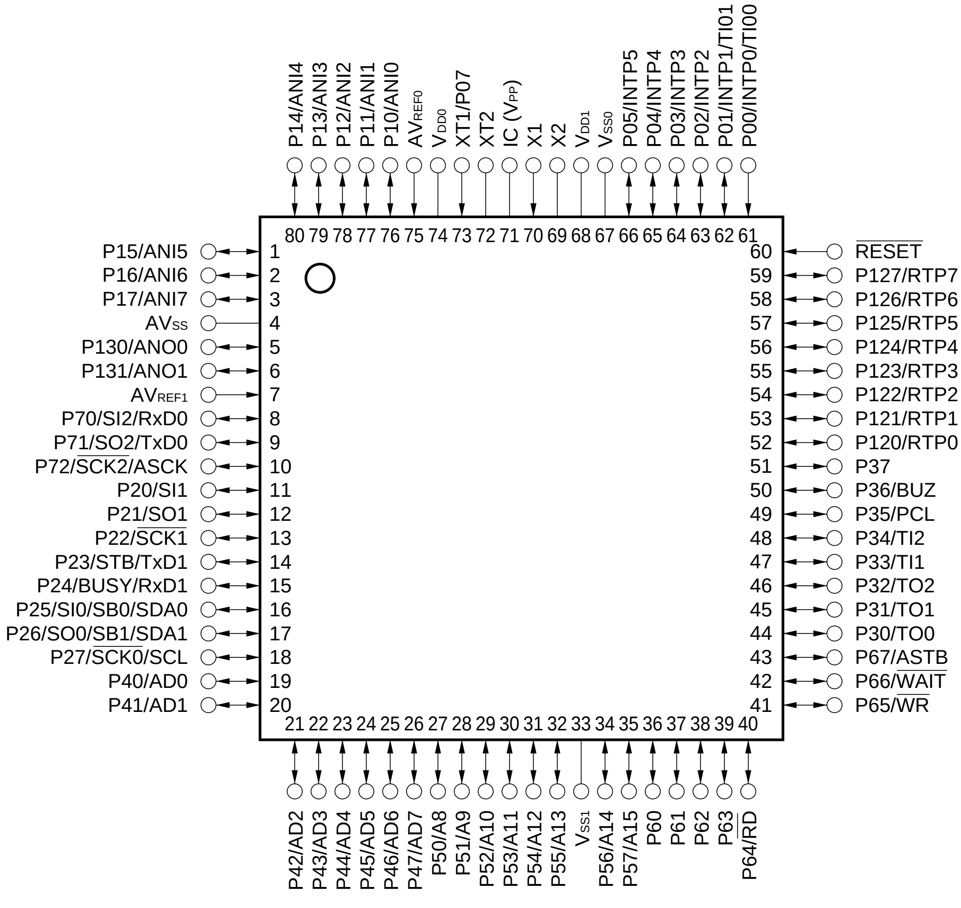
Caution The μPD780053YGC, 780054YGC, 780055YGC, 780056YGC, 780058YGC, and 78F0058YGC are available in two packages. For the package that can be supplied, consult an NEC sales representative.

Remark ××× indicates ROM code suffix.

2.4 Pin Configuration (Top View)

- **80-pin plastic QFP (14 × 14 mm, Resin thickness: 2.7 mm)** Note
  μPD780053YGC-×××-3B9, 780054YGC-×××-3B9, 780055YGC-×××-3B9,
  μPD780056YGC-×××-3B9, 780058YGC-×××-3B9, 78F0058YGC-3B9
- **80-pin plastic QFP (14 × 14 mm, Resin thickness: 1.4 mm)** Note
  μPD780053YGC-×××-8BT, 780054YGC-×××-8BT, 780055YGC-×××-8BT,
  μPD780056YGC-×××-8BT, 780058YGC-×××-8BT, 78F0058YGC-8BT
- **80-pin plastic TQFP (Fine pitch) (12 × 12 mm)** Note
  μPD780053YGK-×××-BE9, 780054YGK-×××-BE9, 780055YGK-×××-BE9,
  μPD780056YGK-×××-BE9, 780058YGK-×××-BE9, 78F0058YGK-BE9

Note Under planning



Cautions **1. Be sure to connect IC (Internally Connected) pin to V_{SS0} directly in normal operating mode.**

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

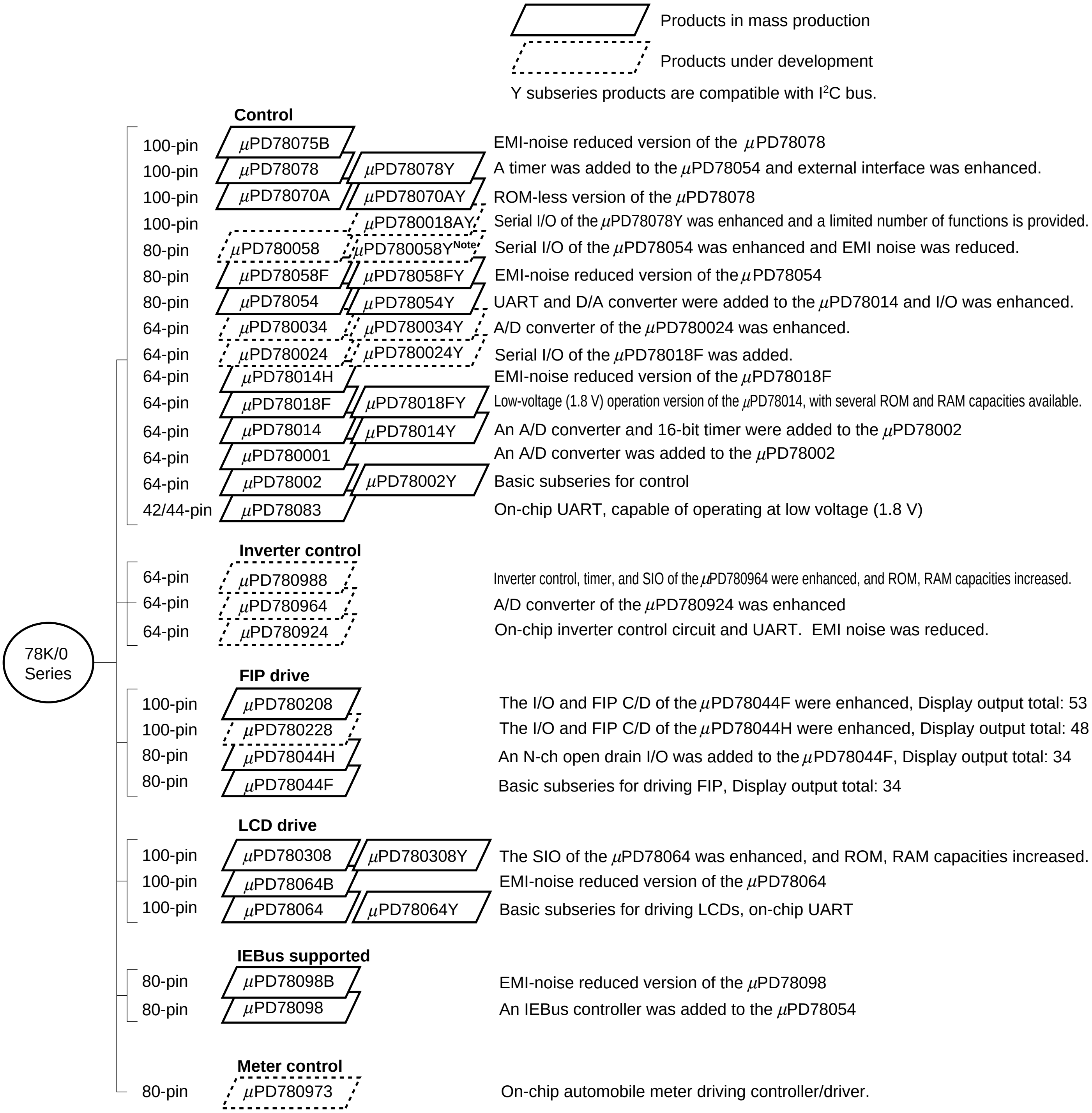
Remarks **1.** Pin connection in parentheses is intended for the μPD78F0058Y.

2. When the μPD780053Y, 780054Y, 780055Y, 780056Y, or 780058Y is used in application fields that require reduction of the noise generated from inside the microcontroller, the implementation of noise reduction measures, such as supplying to V_{DD0} and V_{DD1} individually and connecting V_{SS0} and V_{SS1} to different ground lines, is recommended.

A8 to A15	Address Bus
AD0 to AD7	Address/Data Bus
ANI0 to ANI7	Analog Input
ANO0, ANO1	Analog Output
ASCK	Asynchronous Serial Clock
ASTB	Address Strobe
AV_{REF0}, AV_{REF1}	Analog Reference Voltage
AV_{SS}	Analog Ground
BUSY	Busy
BUZ	Buzzer Clock
IC	Internally Connected
INTP0 to INTP6	Interrupt from Peripherals
P00 to P05, P07	Port 0
P10 to P17	Port 1
P20 to P27	Port 2
P30 to P37	Port 3
P40 to P47	Port 4
P50 to P57	Port 5
P60 to P67	Port 6
P70 to P72	Port 7
P120 to P127	Port 12
P130, P131	Port 13
PCL	Programmable Clock
$\overline{RD}$	Read Strobe
$\overline{RESET}$	Reset
RTP0 to RTP7	Real-Time Output Port
RxD0, RxD1	Receive Data
SB0, SB1	Serial Bus
$\overline{SCK0}$ to $\overline{SCK2}$	Serial Clock
SCL	Serial Clock
SDA0, SDA1	Serial Data
SI0 to SI2	Serial Input
SO0 to SO2	Serial Output
STB	Strobe
TI00, TI01	Timer Input
TI1, TI2	Timer Input
TO0 to TO2	Timer Output
TxD0, TxD1	Transmit Data
V_{DD0}, V_{DD1}	Power Supply
V_{PP}	Programming Power Supply
V_{SS0}, V_{SS1}	Ground
$\overline{WAIT}$	Wait
$\overline{WR}$	Write Strobe
X1, X2	Crystal (Main System Clock)
XT1, XT2	Crystal (Subsystem Clock)

★ 2.5 78K/0 Series Line-up

78K/0 Series product line-up is illustrated below. Part numbers in the boxes indicate subseries names.



Y subseries products are compatible with I^{2}C bus.

Control			
100-pin	μPD78075B		EMI-noise reduced version of the μPD78078
100-pin	μPD78078	μPD78078Y	A timer was added to the μPD78054 and external interface was enhanced.
100-pin	μPD78070A	μPD78070AY	ROM-less version of the μPD78078
100-pin		μPD780018AY	Serial I/O of the μPD78078Y was enhanced and a limited number of functions is provided.
80-pin	μPD780058	μPD780058Y[Note]	Serial I/O of the μPD78054 was enhanced and EMI noise was reduced.
80-pin	μPD78058F	μPD78058FY	EMI-noise reduced version of the μPD78054
80-pin	μPD78054	μPD78054Y	UART and D/A converter were added to the μPD78014 and I/O was enhanced.
64-pin	μPD780034	μPD780034Y	A/D converter of the μPD780024 was enhanced.
64-pin	μPD780024	μPD780024Y	Serial I/O of the μPD78018F was added.
64-pin	μPD78014H		EMI-noise reduced version of the μPD78018F
64-pin	μPD78018F	μPD78018FY	Low-voltage (1.8 V) operation version of the μPD78014, with several ROM and RAM capacities available.
64-pin	μPD78014	μPD78014Y	An A/D converter and 16-bit timer were added to the μPD78002
64-pin	μPD780001		An A/D converter was added to the μPD78002
64-pin	μPD78002	μPD78002Y	Basic subseries for control
42/44-pin	μPD78083		On-chip UART, capable of operating at low voltage (1.8 V)
Inverter control			
64-pin	μPD780988		Inverter control, timer, and SIO of the μPD780964 were enhanced, and ROM, RAM capacities increased.
64-pin	μPD780964		A/D converter of the μPD780924 was enhanced
64-pin	μPD780924		On-chip inverter control circuit and UART. EMI noise was reduced.
FIP drive			
100-pin	μPD780208		The I/O and FIP C/D of the μPD78044F were enhanced, Display output total: 53
100-pin	μPD780228		The I/O and FIP C/D of the μPD78044H were enhanced, Display output total: 48
80-pin	μPD78044H		An N-ch open drain I/O was added to the μPD78044F, Display output total: 34
80-pin	μPD78044F		Basic subseries for driving FIP, Display output total: 34
LCD drive			
100-pin	μPD780308	μPD780308Y	The SIO of the μPD78064 was enhanced, and ROM, RAM capacities increased.
100-pin	μPD78064B		EMI-noise reduced version of the μPD78064
100-pin	μPD78064	μPD78064Y	Basic subseries for driving LCDs, on-chip UART
IEBus supported			
80-pin	μPD78098B		EMI-noise reduced version of the μPD78098
80-pin	μPD78098		An IEBus controller was added to the μPD78054
Meter control			
80-pin	μPD780973		On-chip automobile meter driving controller/driver.

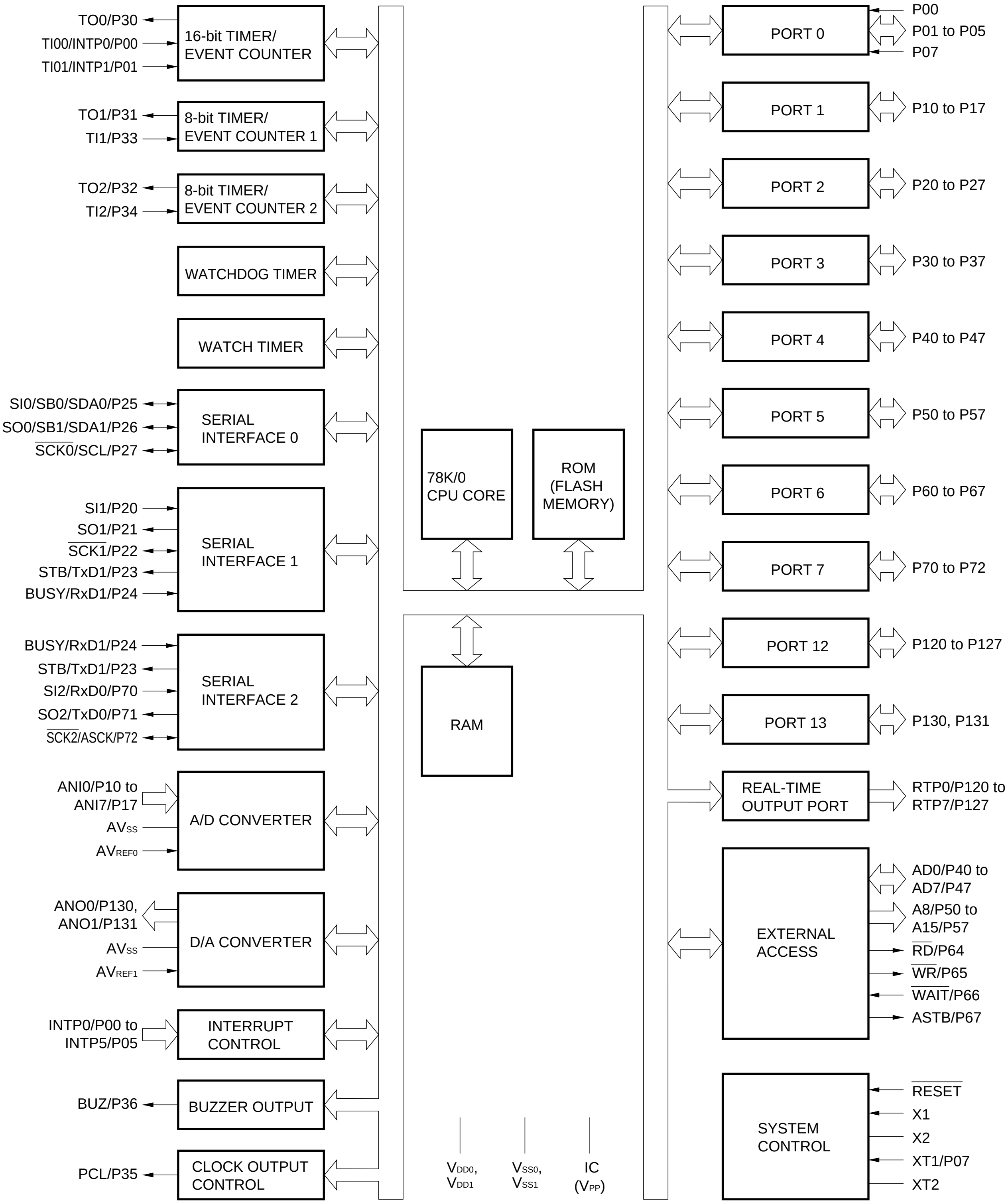
Note Under planning

Major functional differences among the Y subseries are shown below.

Function Subseries Name		ROM Capacity	Serial Interface Configuration	I/O	V$_{DD}$ MIN. Value
Control	μPD78078Y	48 K to 60 K	3-wire/2-wire/I^{2}C : 1 ch 3-wire with auto-transmit/receive function : 1 ch 3-wire/UART : 1 ch	88	1.8 V
	μPD78070AY	—		61	2.7 V
	μPD780018AY	48 K to 60 K	3-wire with auto-transmit/receive function: 1 ch Time-division 3-wire : 1 ch I^{2}C bus (supports multimaster) : 1 ch	88	
	μPD780058Y	24 K to 60 K	3-wire/2-wire/I^{2}C : 1 ch 3-wire with auto-transmit/receive function: 1 ch 3-wire/time-division UART : 1 ch	68	1.8 V
	μPD78058FY	48 K to 60 K	3-wire/2-wire/I^{2}C : 1 ch 3-wire with auto-transmit/receive function : 1 ch 3-wire/UART : 1 ch	69	2.7 V
	μPD78054Y	16 K to 60 K			2.0 V
	μPD780034Y	8 K to 32 K	UART : 1 ch 3-wire : 1 ch I^{2}C bus (supports multimaster) : 1 ch	51	1.8 V
	μPD780024Y				
	μPD78018FY	8 K to 60 K	3-wire/2-wire/I^{2}C : 1 ch 3-wire with auto-transmit/receive function: 1 ch	53	
	μPD78014Y	8 K to 32 K	3-wire/2-wire/SBI/I^{2}C : 1 ch 3-wire with auto-transmit/receive function: 1 ch		2.7 V
	μPD78002Y	8 K to 16 K	3-wire/2-wire/SBI/I^{2}C : 1 ch		
LCD drive	μPD780308Y	48 K to 60 K	3-wire/2-wire/I^{2}C : 1 ch 3-wire/time-division UART : 1 ch 3-wire : 1 ch	57	2.0 V
	μPD78064Y	16 K to 32 K 3-wire/UART	3-wire/2-wire/I^{2}C : 1 ch : 1 ch		

Remark The functions, except for the serial interface, are the same as those of subseries without Y.

2.6 Block Diagram



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

2. Pin connection in parentheses is intended for the μPD78F0058Y.

2.7 Outline of Function

Part Number Item		μPD780053Y Note 1	μPD780054Y Note 1	μPD780055Y Note 1	μPD780056Y Note 1	μPD780058Y Note 1	μPD78F0058Y Note 1
Internal memory	ROM	Mask ROM					Flash memory
		24 Kbytes	32 Kbytes	40 Kbytes	48 Kbytes	60 Kbytes	60 Kbytes Note 2
	High-speed RAM	1,024 bytes					
	Buffer RAM	32 bytes					
	Expansion RAM	None				1,024 bytes	1,024 bytes Note 3
Memory space		64 Kbytes					
General register		8 bits × 8 × 4 banks					
Minimum instruction execution time	With main system clock selected	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (@ 5.0-MHz operation)					
	With subsystem clock selected	122 μs (@ 32.768-kHz operation)					
Instruction set		• 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		• Total : 68 • CMOS input : 2 • CMOS I/O : 62 • N-ch open-drain I/O : 4					
A/D converter		8-bit resolution × 8 channels					
D/A converter		8-bit resolution × 2 channels					
Serial interface		• 3-wire serial I/O/2-wire serial I/O/I^2C bus mode selection possible : 1 channel • 3-wire serial I/O mode (on-chip max. 32-byte auto transmit/receive function) : 1 channel • 3-wire serial I/O/UART mode (on-chip time-division transfer function) selectable : 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		3: (14-bit PWM output enable: 1)					
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (main system clock: @ 5.0-MHz operation) 32.768 kHz (subsystem clock: @ 32.768-kHz operation)					
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (main system clock: @ 5.0-MHz operation)					

Notes 1. Under planning

2. The capacities of the flash memory can be changed by using the memory switching register (IMS).

3. The capacity of the internal expansion RAM can be changed by using the internal expansion RAM size switching register (IXS).

Item	Part Number	μPD780053Y Note	μPD780054Y Note	μPD780055Y Note	μPD780056Y Note	μPD780058Y Note	μPD78F0058Y Note
Vectored interrupt source	Maskable	Internal: 13 External: 6					
	Non-maskable	Internal: 1					
	Software	1					
Test input		Internal: 1 External: 1					
Supply voltage		V_{DD} = 1.8 to 5.5 V					
Operating ambient temperature		T_A = –40 to +85°C					
Package		• 80-pin plastic QFP (14 × 14 mm, Resin thickness: 2.7 mm) • 80-pin plastic QFP (14 × 14 mm, Resin thickness: 1.4 mm) • 80-pin plastic TQFP (Fine pitch) (12 × 12 mm)					

Note Under planning

2.8 Mask Options

The mask ROM versions (μPD780053Y, 780054Y, 780055Y, 780056Y, 780058Y) provide pull-up resistor mask options which allow users to specify whether to connect a pull-up resistor to a specific port pin when the user places an order for the device production. Using this mask option when pull-up resistors are required reduces the number of components to add to the device, resulting in board space saving.

The mask options provided in the μPD780058Y Subseries are shown in Table 2-1.

Table 2-1. Mask Options of Mask ROM Versions

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

3.1 Pin Function List

(1) Port pins (1/2)

Pin Name	Input/Output	Function		After Reset	Alternate Function
P00	Input	Port 0 7-bit input/output port	Input only	Input	INTP0/TI00
P01	Input/ output		Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.	Input	INTP1/TI01
P02					INTP2
P03					INTP3
P04					INTP4
P05					INTP5
P07 Note 1	Input		Input only	Input	XT1
P10 to P17	Input/ output	Port 1 8-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software. Note 2		Input	ANI0 to ANI7
P20	Input/ output	Port 2 8-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.		Input	SI1
P21					SO1
P22					$\overline{\text{SCK1}}$
P23					STB/TxD1
P24					BUSY/RxD1
P25					SI0/SB0
P26					SO0/SB1
P27					$\overline{\text{SCK0}}$
P30	Input/ output	Port 3 8-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.		Input	TO0
P31					TO1
P32					TO2
P33					TI1
P34					TI2
P35					PCL
P36					BUZ
P37					—

Notes **1.** When the P07/XT1 pin is used as an input port, set the bit 6 (FRC) of the processor clock control register (PCC) to 1 (do not use the feedback resistor incorporated in the subsystem clock oscillator).

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

(1) Port pins (2/2)

Pin Name	Input/Output	Function		After Reset	Alternate Function
P40 to P47	Input/ output	Port 4 8-bit input/output port Input/output mode can be specified in 8-bit units. If used as an input port, an on-chip pull-up resistor can be used by software. Test input flag (KRIF) is set to 1 by falling edge detection.		Input	AD0 to AD7
P50 to P57	Input/ output	Port 5 8-bit input/output port LEDs can be driven directly. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.		Input	A8 to A15
P60	Input/ output	Port 6 8-bit input/output port Input/output mode can be specified bit-wise.	N-ch open-drain input/output port. On-chip pull-up resistor can be specified by mask option. (Mask ROM version only). LEDs can be driven directly.	Input	—
P61					
P62					
P63					
P64			If used as an input port, an on-chip pull-up resistor can be used by software.		$\overline{\text{RD}}$
P65					$\overline{\text{WR}}$
P66					$\overline{\text{WAIT}}$
P67					ASTB
P70	Input/ output	Port 7 3-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.		Input	SI2/RxD0
P71					SO2/TxD0
P72					$\overline{\text{SCK2}}$/ASCK
P120 to P127	Input/ output	Port 12 8-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.		Input	RTP0 to RTP7
P130 to P131	Input/ output	Port 13 2-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.		Input	ANO0 to ANO1

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

Pin Name	Input/Output	Function	After Reset	Alternate Function
INTP0	Input	External interrupt request inputs with specifiable valid edges (rising edge, falling edge, both rising and falling edges).	Input	P00/TI00
INTP1				P01/TI01
INTP2				P02
INTP3				P03
INTP4				P04
INTP5				P05
SI0	Input	Serial interface serial data input	Input	P25/SB0
SI1				P20
SI2				P70/RxD
SO0	Output	Serial interface serial data output	Input	P26/SB1
SO1				P21
SO2				P71/TxD
SB0	Input/ output	Serial interface serial data input/output	Input	P25/SI0
SB1				P26/SO0
$\overline{\text{SCK0}}$	Input/ output	Serial interface serial clock input/output	Input	P27
$\overline{\text{SCK1}}$				P22
$\overline{\text{SCK2}}$				P72/ASCK
STB	Output	Serial interface automatic transmit/receive strobe output	Input	P23/TxD1
BUSY	Input	Serial interface automatic transmit/receive busy input	Input	P24/RxD1
RxD0	Input	Asynchronous serial interface serial data input	Input	P70/SI2
RxD1				P24/BUSY
TxD0	Output	Asynchronous serial interface serial data output	Input	P71/SO2
TxD1				P23/STB
ASCK	Input	Asynchronous serial interface serial clock input	Input	P72/$\overline{\text{SCK2}}$
TI00	Input	External count clock input to 16-bit timer (TM0)	Input	P00/INTP0
TI01		Capture trigger signal input to capture register (CR00)		P01/INTP1
TI1		External count clock input to 8-bit timer (TM1)		P33
TI2		External count clock input to 8-bit timer (TM2)		P34
TO0	Output	16-bit timer (TM0) output (also used for 14-bit PWM output)	Input	P30
TO1		8-bit timer (TM1) output		P31
TO2		8-bit timer (TM2) output		P32
PCL	Output	Clock output (for main system clock and subsystem clock trimming)	Input	P35
BUZ	Output	Buzzer output	Input	P36
RTP0 to RTP7	Output	Real-time output port outputting data in synchronization with trigger	Input	P120 to P127

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

Pin Name	Input/Output	Function	After Reset	Alternate Function
AD0 to AD7	Input/output	Low-order address/data bus when expanding external memory	Input	P40 to P47
A8 to A15	Output	High-order address bus when expanding external memory	Input	P50 to P57
$\overline{\text{RD}}$	Output	Strobe signal output for read operation from external memory	Input	P64
$\overline{\text{WR}}$		Strobe signal output for write operation to external memory		P65
$\overline{\text{WAIT}}$	Input	Wait insertion when accessing external memory	Input	P66
ASTB	Output	Strobe output externally latching address information output to ports 4, 5 to access external memory	Input	P67
ANI0 to ANI7	Input	A/D converter analog input	Input	P10 to P17
ANO0, ANO1	Output	D/A converter analog output	Input	P130, P131
AV_{REF0}	Input	A/D converter reference voltage input (also functions as analog power)	—	—
AV_{REF1}	Input	D/A converter reference voltage input	—	—
AV_{SS}	—	A/D converter, D/A converter ground potential. Use the same potential as V_{SS0}.	—	—
$\overline{\text{RESET}}$	Input	System reset input	—	—
X1	Input	Crystal connection for main system clock oscillation	—	—
X2	—		—	—
XT1	Input	Crystal connection for subsystem clock oscillation	Input	P07
XT2	—		—	—
V_{DD0}	—	Positive power supply for ports	—	—
V_{SS0}	—	Ground potential for ports	—	—
V_{DD1}	—	Positive power supply (except ports and analog block)	—	—
V_{SS1}	—	Ground potential (except ports and analog block)	—	—
V_{PP}	—	High-voltage application for program write/verify. Connect directly to V_{SS0} in normal operating mode.	—	—
IC	—	Internally connected. Connect directly to V_{SS0}.	—	—

3.2 Description of Pin Functions

3.2.1 P00 to P05, P07 (Port 0)

This is a 7-bit input/output port. Besides serving as an input/output port, it functions as an external interrupt request input, an external count clock input to the timer, a capture trigger signal input, and crystal connection for subsystem oscillation.

The following operating modes can be specified bit-wise.

(1) Port mode

P00 and P07 function as an input-only port and P01 to P05 function as an input/output ports.

P01 to P05 can be specified for an input or output port bit-wise with a port mode register 0 (PM0). When it is used as an input port, on-chip pull-up resistors can be used to them by defining the pull-up resistor option register L (PUOL).

(2) Control mode

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

(a) INTP0 to INTP5

INTP0 to INTP5 are external interrupt request input pins which can specify valid edges (rising edge, falling edge, and both rising and falling edges). INTP0 or INTP1 become 16-bit timer/event counter capture trigger signal input pin with a valid edge input.

(b) TI00

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

(c) TI01

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

(d) XT1

Crystal connect pin for subsystem clock oscillation

3.2.2 P10 to P17 (Port 1)

This is an 8-bit input/output port. Besides serving as an input/output port, it functions as an A/D converter analog input.

The following operating modes can be specified bit-wise.

(1) Port mode

This port functions as an 8-bit input/output port.

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

(2) Control mode

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

3.2.3 P20 to P27 (Port 2)

This is an 8-bit input/output port. Besides serving as an input/output port, it functions as data input/output to/from the serial interface, clock input/output, automatic transmit/receive busy input, and strobe output functions.

The following operating modes can be specified bit-wise.

(1) Port mode

This port functions as an 8-bit input/output port. It can be specified bit-wise as an input or output port with port mode register 2 (PM2). When it is used as an input port, on-chip pull-up resistors can be used to them by defining the pull-up resistor option register L (PUOL).

(2) Control mode

This port functions as serial interface data input/output, clock input/output, automatic transmit/receive busy input, and strobe output functions.

(a) SI0, SI1, SO0, SO1

Serial data input/output pins of serial interface

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

Serial clock input/output pins of serial interface

(c) SB0 and SB1

NEC standard serial bus interface input/output pins

(d) BUSY

Automatic transmit/receive busy input pins of serial interface

(e) STB

Automatic transmit/receive strobe output pins of serial interface

(f) RxD1, TxD1

Serial interface serial data input/output pins of asynchronous serial interface

Caution When this port is used as a serial interface pin, the I/O and output latches must be set according to the function the user requires. For the setting, refer to Figure 16-4 Serial Operation Mode Register 0 Format, Figure 18-3 Serial Operation Mode Register 1 Format, and Table 19-2 Serial Interface Channel 2 Operating Mode Settings.

3.2.4 P30 to P37 (Port 3)

This is an 8-bit input/output port. Beside serving as an input/output port, it functions as timer input/output, clock output and buzzer output.

The following operating modes can be specified bit-wise.

(1) Port mode

This port functions as an 8-bit input/output port. It can be specified bit-wise as an input or output port with port mode register 3 (PM3). When it is used as an input port, on-chip pull-up resistors can be used by defining the pull-up resistor option register L (PUOL).

(2) Control mode

This port functions as timer input/output, clock output, and buzzer output.

(a) TI1 and TI2

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

(b) TO0 to TO2

Timer output pins

(c) PCL

Clock output pin

(d) BUZ

Buzzer output pin

3.2.5 P40 to P47 (Port 4)

This is an 8-bit input/output port. Besides serving as an input/output port, it functions as an address/data bus.

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

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

(1) Port mode

This port functions as an 8-bit input/output port. They can be specified in 8-bit units for an input or output port by using the memory expansion mode register (MM). When it is used as an input port, on-chip pull-up resistors can be used by defining the pull-up resistor option register L (PUOL).

(2) Control mode

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

3.2.6 P50 to P57 (Port 5)

This is an 8-bit input/output port. Besides serving as an input/output port, it functions as an address bus.

Port 5 can drive LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

This port functions as an 8-bit input/output port. It can be specified bit-wise as an input/output port with port mode register 5 (PM5). When it is used as an input port, on-chip pull-up resistors can be used by defining the pull-up resistor option register L (PUOL).

(2) Control mode

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

3.2.7 P60 to P67 (Port 6)

This is an 8-bit input/output port. Besides serving as an input/output port, it is used for control in external memory expansion mode. P60 to P63 can be driven LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

This port functions as an 8-bit input/output port. It can be specified bit-wise as an input or output port with port mode register 6 (PM6).

P60 to P63 are N-ch open drain outputs. Mask ROM version can contain pull-up resistors with the mask option. When P64 to P67 are used as an input port, on-chip pull-up resistor can be used by defining the pull-up resistor option register L (PUOL).

(2) Control mode

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

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

3.2.8 P70 to P72 (Port 7)

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

The following operating modes can be specified bit-wise.

(1) Port mode

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

(2) Control mode

This port functions as serial interface data input/output and clock input/output.

(a) SI2, SO2

Serial data input/output pins of serial interface

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

Serial clock input/output pin of serial interface

(c) RxD0, TxD0

Serial data input/output pins of asynchronous serial interface

(d) ASCK

Serial clock input/output pin of asynchronous serial interface

Caution **When this port is used as a serial interface pin, the I/O and output latches must be set according to the function the user requires.**
For the setting, see the operation mode setting list in Table 19-2 Serial Interface Channel 2.

3.2.9 P120 to P127 (Port 12)

This is an 8-bit input/output port. Besides serving as an input/output port, it functions as a real-time output port.

The following operating modes can be specified bit-wise.

(1) Port mode

This port functions as an 8-bit input/output port. It can be specified bit-wise as an input or output port with port mode register 12 (PM12). When they are used as an input port, on-chip pull-up resistors can be used by defining the pull-up resistor option register H (PUOH).

(2) Control mode

This port functions as a real-time output port (RTP0 to RTP7) outputting data in synchronization with a trigger.

3.2.10 P130 and P131 (Port 13)

This is a 2-bit input/output port. Besides serving as an input/output port, it is used for a D/A converter analog output. The following operating modes can be specified bit-wise.

(1) Port mode

This port functions as a 2-bit input/output port. It can be specified bit-wise as an input or output port with port mode register 13 (PM13). When it is used as an input port, on-chip pull-up resistors can be used by defining the pull-up resistor option register H (PUOH).

(2) Control mode

This port allows D/A converter analog outputs (ANO0 and ANO1).

Caution When only either one of the D/A converter channels is used with AV_{REF1} < V_{DD0}, the other pins that are not used as analog outputs must be set as follows:

- **Set PM13x bit of the port mode register 13 (PM13) to 1 (input mode) and connect the pin to V_{SS0}.**
- **Set PM13x bit of the port mode register 13 (PM13) to 0 (output mode) and the output latch to 0, to output low level from the pin.**

3.2.11 AV_{REF0}

An A/D converter reference voltage input pin. This pin also serves as an analog power supply pin. Supply power to this pin when the A/D converter is used.

When the A/D converter is not used, use the same voltage that of the V_{SS0} pin.

3.2.12 AV_{REF1}

A D/A converter reference voltage input pin.

When the D/A converter is not used, use the same voltage that of the V_{DD0} pin.

3.2.13 AV_{SS}

This is a ground voltage pin of A/D converter and D/A converter. Always use the same voltage as that of the V_{SS0} pin even when the A/D converter or D/A converter is not used.

3.2.14 $\overline{\text{RESET}}$

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

3.2.15 X1 and X2

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

3.2.16 XT1 and XT2

Crystal resonator connect pins for subsystem clock oscillation.

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

3.2.17 V_{DD0}, V_{DD1}

V_{DD0} is the positive power supply pin for ports.

V_{DD1} is the positive power supply pin for blocks other than port and analog blocks.

3.2.18 V_{SS0}, V_{SS1}

V_{SS0} is the ground potential pin for ports.

V_{SS1} is the ground potential pin for blocks other than port and analog blocks.

3.2.19 V_{PP} (Flash memory version only)

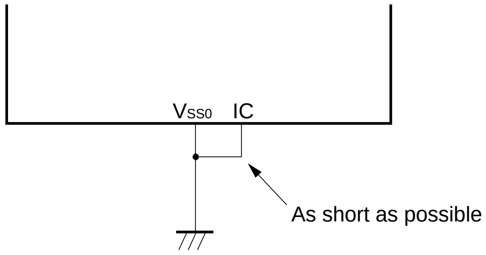
High-voltage apply pin for flash memory programming mode setting and program write/verify. Connect directly to V_{SS0} pin in normal operating mode.

3.2.20 IC (Mask ROM version only)

The IC (Internally Connected) pin is provided to set the test mode to check the μPD780058 Subseries at delivery. Connect it directly to the V_{SS0} with the shortest possible wire in the normal operating mode.

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

○ Connect IC pins to V_{SS0} pins directly.



3.3 Input/output Circuits and Recommended Connection of Unused Pins

Table 3-1 shows the input/output circuit types of pins and the recommended connection of for unused pins. Refer to Figure 3-1 for the configuration of the input/output circuit of each type.

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

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
P00/INTP0/TI00	2	Input	Connect to V_{SS0}.
P01/INTP1/TI01	8-C	Input/output	Connect independently via a resistor to V_{SS0}.
P02/INTP2			
P03/INTP3			
P04/INTP4			
P05/INTP5			
P07/XT1	16	Input	Connect to V_{DD0}.
P10/ANI0 to P17/ANI7	11-D	Input/output	Connect independently via a resistor to V_{DD0} or V_{SS0}.
P20/SI1	8-C		
P21/SO1	5-H		
P22/$\overline{\text{SCK1}}$	8-C		
P23/STB/TxD1	5-H		
P24/BUSY/RxD1	8-C		
P25/SI0/SB0	10-B		
P26/SO0/SB1			
P27/$\overline{\text{SCK0}}$			
P30/TO0	5-H		
P31/TO1			
P32/TO2			
P33/TI1	8-C		
P34/TI2			
P35/PCL	5-H		
P36/BUZ			
P37			
P40/AD0 to P47/AD7	5-N	Input/output	Connect independently via a resistor to V_{DD0}.
P50/A8 to P57/A15	5-H	Input/output	Connect independently via a resistor to V_{DD0} or V_{SS0}.

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

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
P60 to P63 (Mask ROM version)	13-J	Input/output	Connect independently via a resistor to V_{DD0}.
P60 to P63 (Flash memory version)	13-K		
P64/$\overline{RD}$	5-H	Input/output	Connect independently via a resistor to V_{DD0} or V_{SS0}.
P65/$\overline{WR}$			
P66/$\overline{WAIT}$			
P67/ASTB			
P70/SI2/RxD0	8-C		
P71/SO2/TxD0	5-H		
P72/$\overline{SCK2}$/ASCK	8-C		
P120/RTP0 to P127/RTP7	5-H		
P130/ANO0, P131/ANO1	12-C	Input/output	Connect independently via a resistor to V_{SS0}.
$\overline{RESET}$	2	Input	—
XT2	16	—	Leave open
AV_{REF0}	—		Connect to V_{SS0}.
AV_{REF1}			Connect to V_{DD0}.
AV_{SS}			Connect to V_{SS0}.
IC (Mask ROM version)			Connect directly to V_{SS0}.
V_{PP} (Flash memory version)			

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

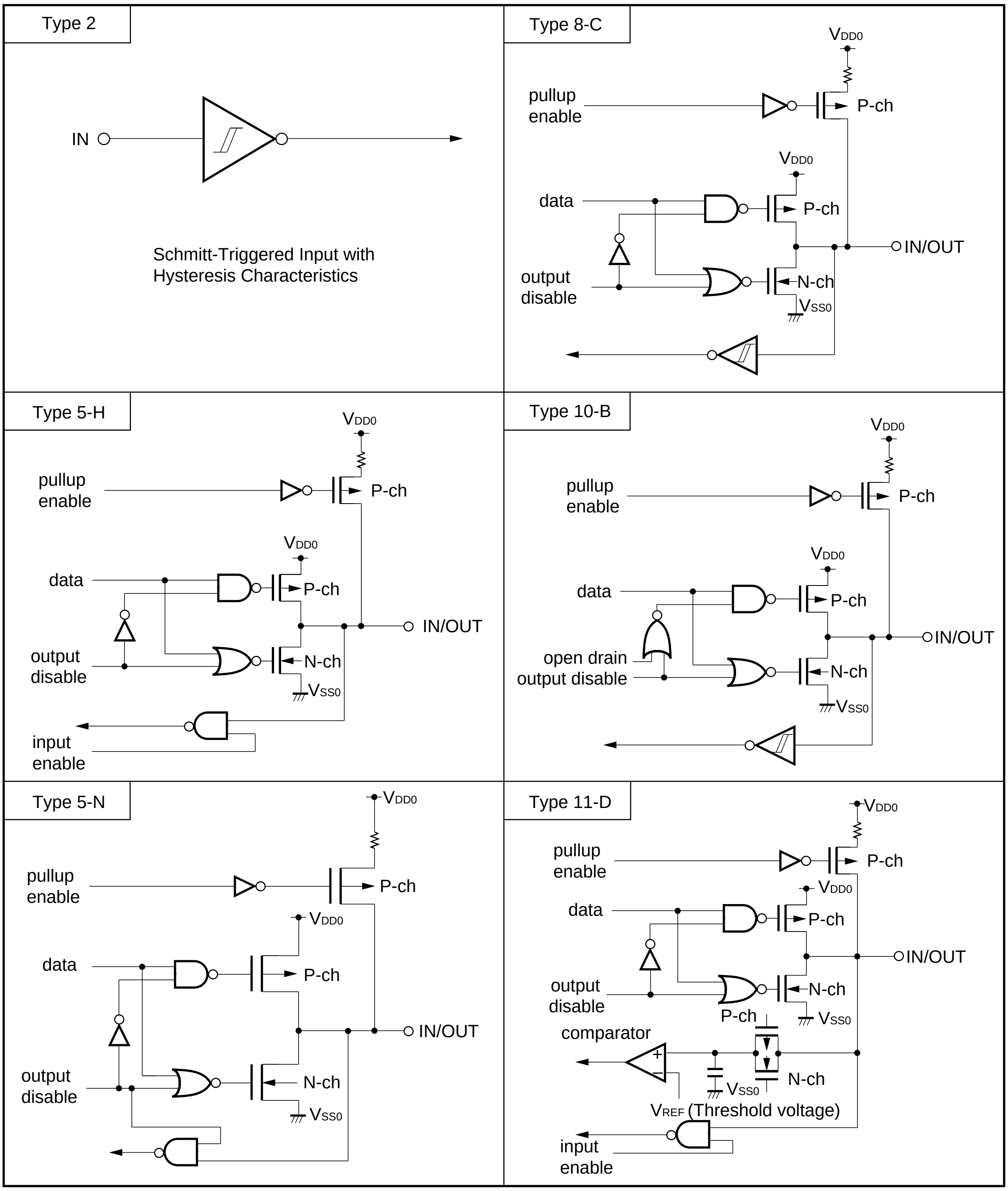
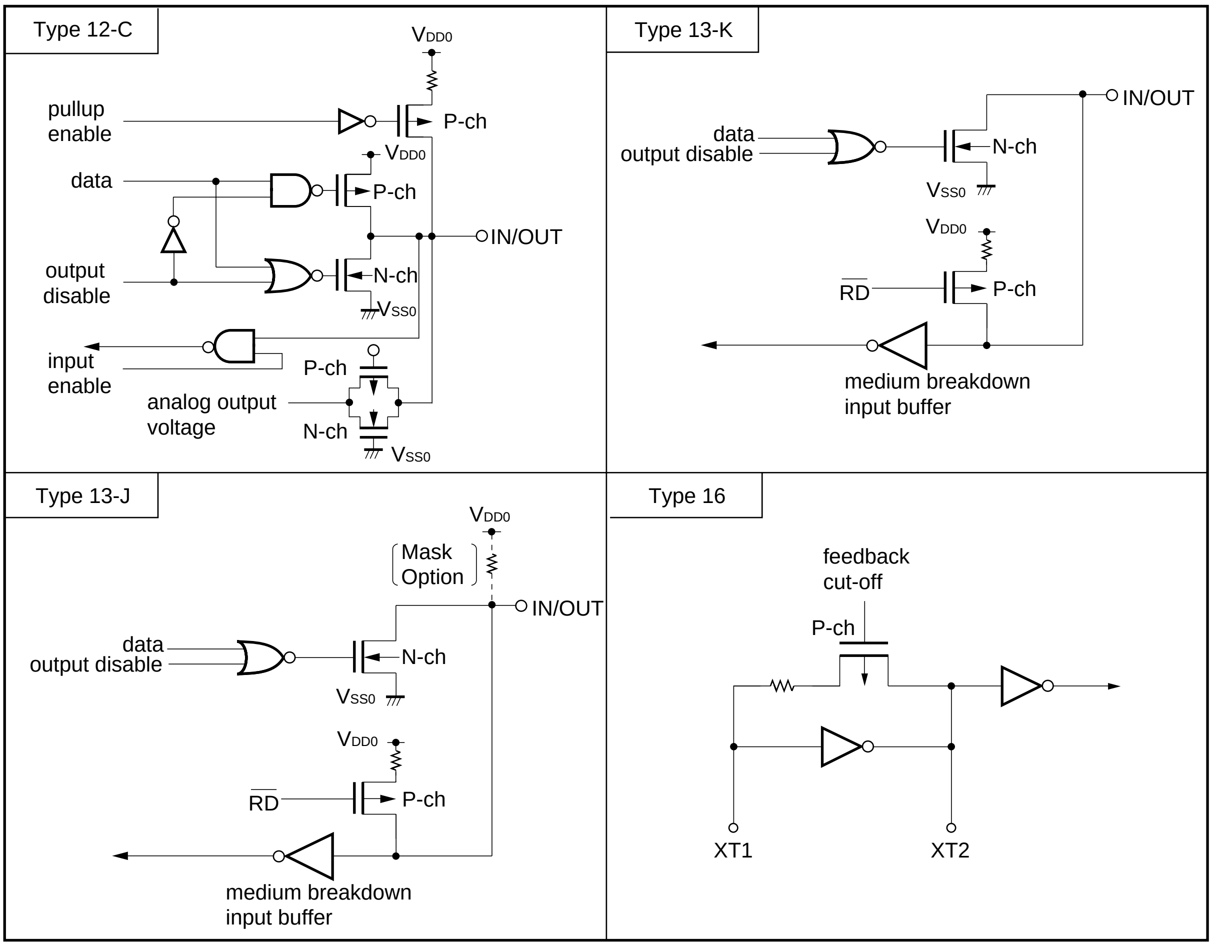


Figure 3-1. Pin Input/Output Circuit List (2/2)



[MEMO]

CHAPTER 4 PIN FUNCTION (μPD780058Y Subseries)

4.1 Pin Function List

(1) Port pins (1/2)

Pin Name	Input/Output	Function		After Reset	Alternate Function
P00	Input	Port 0 7-bit input/output port	Input only	Input	INTP0/TI00
P01	Input/ output		Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.		INTP1/TI01
P02					INTP2
P03					INTP3
P04					INTP4
P05					INTP5
P07 Note 1	Input		Input only	Input	XT1
P10 to P17	Input/ output	Port 1 8-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software Note 2.		Input	ANI0 to ANI7
P20	Input/ output	Port 2 8-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.			SI1
P21					SO1
P22					$\overline{\text{SCK1}}$
P23					STB/TxD1
P24					BUSY/RxD1
P25					SI0/SB0/SDA0
P26					SO0/SB1/SDA1
P27					$\overline{\text{SCK0}}$/SCL
P30	Input/ output	Port 3 8-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.		Input	TO0
P31					TO1
P32					TO2
P33					TI1
P34					TI2
P35					PCL
P36					BUZ
P37					—

Notes **1.** When the P07/XT1 pin is used as an input port, set the bit 6 (FRC) of the processor clock control register (PCC) to 1 (do not use the feedback resistor incorporated in the subsystem clock oscillator).

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

(1) Port pins (2/2)

<table>
<tr><th>Pin Name</th><th>Input/Output</th><th colspan="2">Function</th><th>After Reset</th><th>Alternate Function</th></tr>
<tr><td>P40 to P47</td><td>Input/output</td><td colspan="2">Port 4
8-bit input/output port
Input/output mode can be specified in 8-bit units.
If used as an input port, an on-chip pull-up resistor can be used by software.
Test input flag (KRIF) is set to 1 by falling edge detection.</td><td>Input</td><td>AD0 to AD7</td></tr>
<tr><td>P50 to P57</td><td>Input/output</td><td colspan="2">Port 5
8-bit input/output port
LEDs can be driven directly.
Input/output mode can be specified bit-wise.
If used as an input port, an on-chip pull-up resistor can be used by software.</td><td>Input</td><td>A8 to A15</td></tr>
<tr><td>P60</td><td rowspan="8">Input/output</td><td rowspan="8">Port 6
8-bit input/output port
Input/output mode can be specified bit-wise.</td><td rowspan="4">N-ch open-drain input/output port.
On-chip pull-up resistor can be specified by mask option.
(Mask ROM version only).
LEDs can be driven directly.</td><td rowspan="8"></td><td rowspan="4">—</td></tr>
<tr><td>P61</td></tr>
<tr><td>P62</td></tr>
<tr><td>P63</td></tr>
<tr><td>P64</td><td rowspan="4">If used as an input port, an on-chip pull-up resistor can be used by software.</td><td>$\overline{\text{RD}}$</td></tr>
<tr><td>P65</td><td>$\overline{\text{WR}}$</td></tr>
<tr><td>P66</td><td>$\overline{\text{WAIT}}$</td></tr>
<tr><td>P67</td><td>ASTB</td></tr>
<tr><td>P70</td><td rowspan="3">Input/output</td><td colspan="2" rowspan="3">Port 7
3-bit input/output port
Input/output mode can be specified bit-wise.
If used as an input port, an on-chip pull-up resistor can be used by software.</td><td rowspan="3"></td><td>SI2/RxD0</td></tr>
<tr><td>P71</td><td>SO2/TxD0</td></tr>
<tr><td>P72</td><td>$\overline{\text{SCK2}}$/ASCK</td></tr>
<tr><td>P120 to P127</td><td>Input/output</td><td colspan="2">Port 12
8-bit input/output port
Input/output mode can be specified bit-wise.
If used as an input port, an on-chip pull-up resistor can be used by software.</td><td></td><td>RTP0 to RTP7</td></tr>
<tr><td>P130 to P131</td><td>Input/output</td><td colspan="2">Port 13
2-bit input/output port
Input/output mode can be specified bit-wise.
If used as an input port, an on-chip pull-up resistor can be used by software.</td><td></td><td>ANO0 to ANO1</td></tr>
</table>

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

Pin Name	Input/Output	Function	After Reset	Alternate Function
INTP0	Input	External interrupt request inputs with specifiable valid edges (rising edge, falling edge, both rising and falling edges).	Input	P00/TI00
INTP1				P01/TI01
INTP2				P02
INTP3				P03
INTP4				P04
INTP5				P05
SI0	Input	Serial interface serial data input	Input	P25/SB0/SDA0
SI1				P20
SI2				P70/RxD
SO0	Output	Serial interface serial data output	Input	P26/SB1/SDA1
SO1				P21
SO2				P71/TxD
SB0	Input/ output	Serial interface serial data input/output	Input	P25/SI0/SDA0
SB1				P26/SO0/SDA1
SDA0				P25/SI0/SB0
SDA1				P26/SO0/SB1
$\overline{\text{SCK0}}$	Input/ output	Serial interface serial clock input/output	Input	P27/SCL
$\overline{\text{SCK1}}$				P22
$\overline{\text{SCK2}}$				P72/ASCK
SCL				P27/$\overline{\text{SCK0}}$
STB	Output	Serial interface automatic transmit/receive strobe output	Input	P23/TxD1
BUSY	Input	Serial interface automatic transmit/receive busy input	Input	P24/RxD1
RxD0	Input	Asynchronous serial interface serial data input	Input	P70/SI2
RxD1				P24/BUSY
TxD	Output	Asynchronous serial interface serial data output	Input	P71/$\overline{\text{SO2}}$
TxD1				P23/STB
ASCK	Input	Asynchronous serial interface serial clock input	Input	P72/$\overline{\text{SCK2}}$
TI00	Input	External count clock input to 16-bit timer (TM0)	Input	P00/INTP0
TI01		Capture trigger signal input to capture register (CR00)		P01/INTP1
TI1		External count clock input to 8-bit timer (TM1)		P33
TI2		External count clock input to 8-bit timer (TM2)		P34
TO0	Output	16-bit timer (TM0) output (also used for 14-bit PWM output)	Input	P30
TO1		8-bit timer (TM1) output		P31
TO2		8-bit timer (TM2) output		P32
PCL	Output	Clock output (for main system clock and subsystem clock trimming)	Input	P35
BUZ	Output	Buzzer output	Input	P36
RTP0 to RTP7	Output	Real-time output port outputting data in synchronization with trigger	Input	P120 to P127

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

Pin Name	Input/Output	Function	After Reset	Alternate Function
AD0 to AD7	Input/output	Low-order address/data bus when expanding external memory	Input	P40 to P47
A8 to A15	Output	High-order address bus when expanding external memory	Input	P50 to P57
$\overline{RD}$	Output	Strobe signal output for read operation from external memory	Input	P64
$\overline{WR}$		Strobe signal output for write operation to external memory		P65
$\overline{WAIT}$	Input	Wait insertion when accessing external memory	Input	P66
ASTB	Output	Strobe output externally latching address information output to ports 4, 5 to access external memory	Input	P67
ANI0 to ANI7	Input	A/D converter analog input	Input	P10 to P17
ANO0, ANO1	Output	D/A converter analog output	Input	P130, P131
AV_{REF0}	Input	A/D converter reference voltage input (also functions as analog power)	—	—
AV_{REF1}	Input	D/A converter reference voltage input	—	—
AV_{SS}	—	A/D converter, D/A converter ground potential. Use the same potential as V_{SS0}.	—	—
$\overline{RESET}$	Input	System reset input	—	—
X1	Input	Crystal connection for main system clock oscillation	—	—
X2	—		—	—
XT1	Input	Crystal connection for subsystem clock oscillation	Input	P07
XT2	—		—	—
V_{DD0}	—	Positive power supply for ports	—	—
V_{SS0}	—	Ground potential for ports	—	—
V_{DD1}	—	Positive power supply (except ports and analog block)	—	—
V_{SS1}	—	Ground potential (except ports and analog block)	—	—
V_{PP}	—	High-voltage application for program write/verify. Connect directly to V_{SS0} in normal operating mode.	—	—
V_{SS}	—	Ground potential	—	—
IC	—	Internally connected. Connect directly to V_{SS0}.	—	—

4.2 Description of Pin Functions

4.2.1 P00 to P05, P07 (Port 0)

This is a 7-bit input/output port. Besides serving as an input/output port, it functions as an external interrupt request input, an external count clock input to the timer, a capture trigger signal input, and crystal connection for subsystem oscillation.

The following operating modes can be specified bit-wise.

(1) Port mode

P00 and P07 function as an input-only port and P01 to P05 function as an input/output ports.

P01 to P05 can be specified for an input or output port bit-wise with a port mode register 0 (PM0). When it is used as an input port, on-chip pull-up resistors can be used to them by defining the pull-up resistor option register L (PUOL).

(2) Control mode

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

(a) INTP0 to INTP5

INTP0 to INTP5 are external interrupt request input pins which can specify valid edges (rising edge, falling edge, and both rising and falling edges). INTP0 and INTP1 become a 16-bit timer/event counter capture trigger signal input pin with a valid edge input.

(b) TI00

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

(c) TI01

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

(d) XT1

Crystal connect pin for subsystem clock oscillation

4.2.2 P10 to P17 (Port 1)

This is an 8-bit input/output port. Besides serving as an input/output port, it functions as an A/D converter analog input.

The following operating modes can be specified bit-wise.

(1) Port mode

This port functions as an 8-bit input/output port.

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

(2) Control mode

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

4.2.3 P20 to P27 (Port 2)

This is an 8-bit input/output port. Besides serving as an input/output port, it functions as data input/output to/from the serial interface, clock input/output, automatic transmit/receive busy input, and strobe output functions.

The following operating modes can be specified bit-wise.

(1) Port mode

This port functions as an 8-bit input/output port. It can be specified bit-wise as an input or output port with port mode register 2 (PM2). When it is used as an input port, on-chip pull-up resistors can be used to them by defining the pull-up resistor option register L (PUOL).

(2) Control mode

This port functions as serial interface data input/output, clock input/output, automatic transmit/receive busy input, and strobe output functions.

(a) SI0, SI1, SO0, SO1, SB0, SB1, SDA0, SDA1

Serial data input/output pins of serial interface

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

Serial clock input/output pins of serial interface

(c) BUSY

Automatic transmit/receive busy input pins of serial interface

(d) STB

Automatic transmit/receive strobe output pins of serial interface

(e) RxD1, TxD1

Serial interface serial data input/output pins of asynchronous serial interface

Caution When this port is used as a serial interface pin, the I/O and output latches must be set according to the function the user requires. For the setting, refer to Figure 17-4 Serial Operation Mode Register 0 Format, Figure 18-3 Serial Operation Mode Register 1 Format, and Table 19-2 Serial Interface Channel 2 Operating Mode Settings.

4.2.4 P30 to P37 (Port 3)

This is an 8-bit input/output port. Beside serving as an input/output port, it functions as timer input/output, clock output, and buzzer output.

The following operating modes can be specified bit-wise.

(1) Port mode

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

(2) Control mode

This port functions as timer input/output, clock output, and buzzer output.

(a) TI1 and TI2

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

(b) TO0 to TO2

Timer output pins

(c) PCL

Clock output pin

(d) BUZ

Buzzer output pin

4.2.5 P40 to P47 (Port 4)

This is an 8-bit input/output port. Besides serving as an input/output port, it functions as an address/data bus.

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

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

(1) Port mode

This port functions as an 8-bit input/output port. They can be specified in 8-bit units for an input or output port by using the memory expansion mode register (MM). When it is used as an input port, on-chip pull-up resistors can be used by defining the pull-up resistor option register L (PUOL).

(2) Control mode

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

4.2.6 P50 to P57 (Port 5)

This is an 8-bit input/output port. Besides serving as an input/output port, it functions as an address bus.

Port 5 can drive LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

This port functions as an 8-bit input/output port. It can be specified bit-wise as an input/output port with port mode register 5 (PM5). When it is used as an input port, on-chip pull-up resistors can be used by defining the pull-up resistor option register L (PUOL).

(2) Control mode

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

4.2.7 P60 to P67 (Port 6)

This is an 8-bit input/output port. Besides serving as an input/output port, it is used for control in external memory expansion mode. P60 to P63 can be driven LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

This port functions as an 8-bit input/output port. It can be specified bit-wise as an input or output port with port mode register 6 (PM6).

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

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

(2) Control mode

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

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

4.2.8 P70 to P72 (Port 7)

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

The following operating modes can be specified bit-wise.

(1) Port mode

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

(2) Control mode

This port functions as serial interface data input/output and clock input/output.

(a) SI2, SO2

Serial data input/output pins of serial interface

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

Serial clock input/output pin of serial interface

(c) RxD0, TxD0

Serial interface serial data input/output pins of asynchronous serial interface

(d) ASCK

Serial clock input/output pin of asynchronous serial interface

Caution When this port is used as a serial interface pin, the I/O and output latches must be set according to the function the user requires.
For the setting, see to the operation mode setting list in Table 19-2 Serial Interface Channel 2.

4.2.9 P120 to P127 (Port 12)

This is an 8-bit input/output port. Besides serving as an input/output port, it functions as a real-time output port.

The following operating modes can be specified bit-wise.

(1) Port mode

This port functions as an 8-bit input/output port. It can be specified bit-wise as an input or output port with port mode register 12 (PM12). When they are used as an input port, on-chip pull-up resistors can be used by defining the pull-up resistor option register H (PUOH).

(2) Control mode

This port functions as a real-time output port (RTP0 to RTP7) outputting data in synchronization with a trigger.

4.2.10 P130 and P131 (Port 13)

This is a 2-bit input/output port. Besides serving as an input/output port, it is used for a D/A converter analog output. The following operating modes can be specified bit-wise.

(1) Port mode

This port functions as a 2-bit input/output port. It can be specified bit-wise as an input or output port with port mode register 13 (PM13). When it is used as an input port, on-chip pull-up resistors can be used by defining the pull-up resistor option register H (PUOH).

(2) Control mode

This port allows D/A converter analog outputs (ANO0 and ANO1).

Caution When only either one of the D/A converter channels is used with $AV_{REF1} < V_{DD0}$, the other pins that are not used as analog outputs must be set as follows:

- **Set PM13x bit of the port mode register 13 (PM13) to 1 (input mode) and connect the pin to V_{SS0}.**
- **Set PM13x bit of the port mode register 13 (PM13) to 0 (output mode) and the output latch to 0, to output low level from the pin.**

4.2.11 AV_{REF0}

An A/D converter reference voltage input pin. This pin also serves as an analog power supply pin. Supply power to this pin when the A/D converter is used.

When the A/D converter is not used, use the same voltage that of the V_{SS0} pin.

4.2.12 AV_{REF1}

A D/A converter reference voltage input pin.

When the D/A converter is not used, use the same voltage that of the V_{DD0} pin.

4.2.13 AV_{SS}

This is a ground voltage pin of A/D converter and D/A converter. Always use the same voltage as that of the V_{SS0} pin even when the A/D converter or D/A converter is not used.

4.2.14 $\overline{\text{RESET}}$

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

4.2.15 X1 and X2

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

4.2.16 XT1 and XT2

Crystal resonator connect pins for subsystem clock oscillation.

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

4.2.17 V_{DD0}, V_{DD1}

V_{DD0} is the positive power supply pin for ports.

V_{DD1} is the positive power supply pin for blocks other than port and analog blocks.

4.2.18 V_{SS0}, V_{SS1}

V_{SS0} is the ground potential pin for ports.

V_{SS1} is the ground potential pin for blocks other than port and analog blocks.

4.2.19 V_{PP} (Flash memory version only)

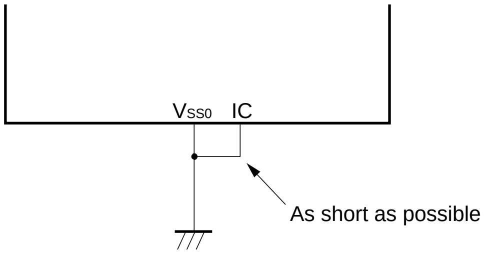
High-voltage apply pin for flash memory programming mode setting and program write/verify. Connect directly to V_{SS0} pin in normal operating mode.

4.2.20 IC (Mask ROM version only)

The IC (Internally Connected) pin is provided to set the test mode to check the μPD780058Y Subseries at delivery. Connect it directly to the V_{SS0} with the shortest possible wire in the normal operating mode.

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

○ Connect IC pins to V_{SS0} pins directly.



4.3 Input/Output Circuits and Recommended Connection of Unused Pins

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

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

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
P00/INTP0/TI00	2	Input	Connect to V_{SS0}.
P01/INTP1/TI01	8-C	Input/output	Connect independently via a resistor to V_{SS0}.
P02/INTP2			
P03/INTP3			
P04/INTP4			
P05/INTP5			
P07/XT1	16	Input	Connect to V_{DD0}
P10/ANI0 to P17/ANI7	11-D	Input/output	Connect independently via a resistor to V_{DD0} or V_{SS0}.
P20/SI1	8-C		
P21/SO1	5-H		
P22/$\overline{\text{SCK1}}$	8-C		
P23/STB/TxD1	5-H		
P24/BUSY/RxD1	8-C		
P25/SI0/SB0/SDA0	10-B		
P26/SO0/SB1/SDA1			
P27/$\overline{\text{SCK0}}$/SCL			
P30/TO0	5-H		
P31/TO1			
P32/TO2			
P33/TI1	8-C		
P34/TI2			
P35/PCL	5-H		
P36/BUZ			
P37			
P40/AD0 to P47/AD7	5-N	Input/output	Connect independently via a resistor to V_{DD0}.
P50/A8 to P57/A15	5-H	Input/output	Connect independently via a resistor to V_{DD0} or V_{SS0}.

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

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
P60 to P63 (Mask ROM version)	13-J	Input/output	Connect independently via a resistor to V_{DD0}.
P60 to P63 (Flash memory version)	13-K	Input/output	Connect independently via a resistor to V_{DD0} or V_{SS0}.
P64/$\overline{RD}$	5-H		
P65/$\overline{WR}$			
P66/$\overline{WAIT}$			
P67/ASTB			
P70/SI2/RxD0	8-C		
P71/SO2/TxD0	5-H		
P72/$\overline{SCK2}$/ASCK	8-C		
P120/RTP0 to P127/RTP7	5-H		
P130/ANO0, P131/ANO1	12-C	Input/output	Connect independently via a resistor to V_{SS0}.
$\overline{RESET}$	2	Input	—
XT2	16	—	Leave open
AV_{REF0}	—		Connect to V_{SS0}.
AV_{REF1}			Connect to V_{DD0}.
AV_{SS}			Connect to V_{SS0}.
IC (Mask ROM version)			Connect directly to V_{SS0}.
V_{PP} (Flash memory version)			

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

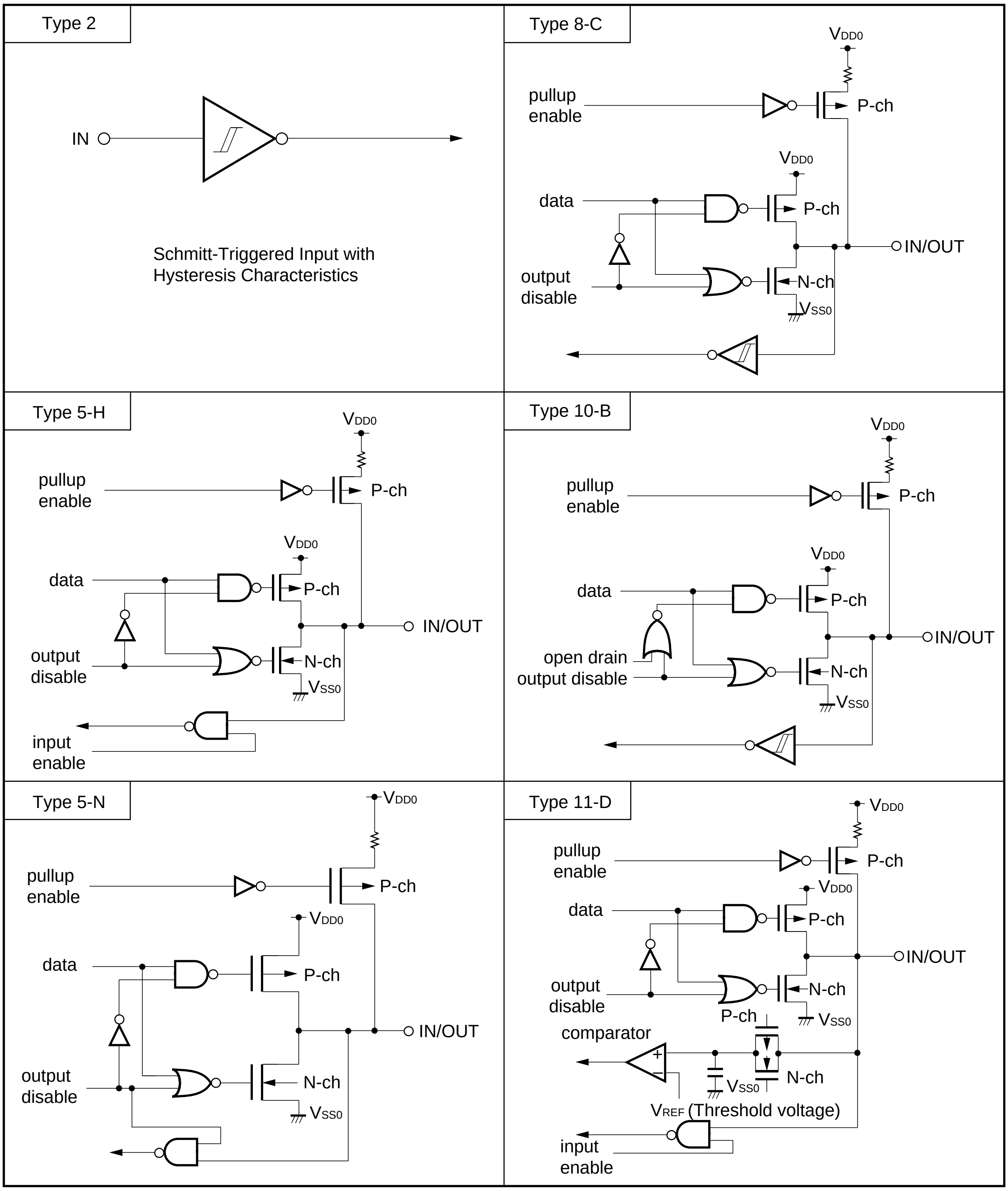
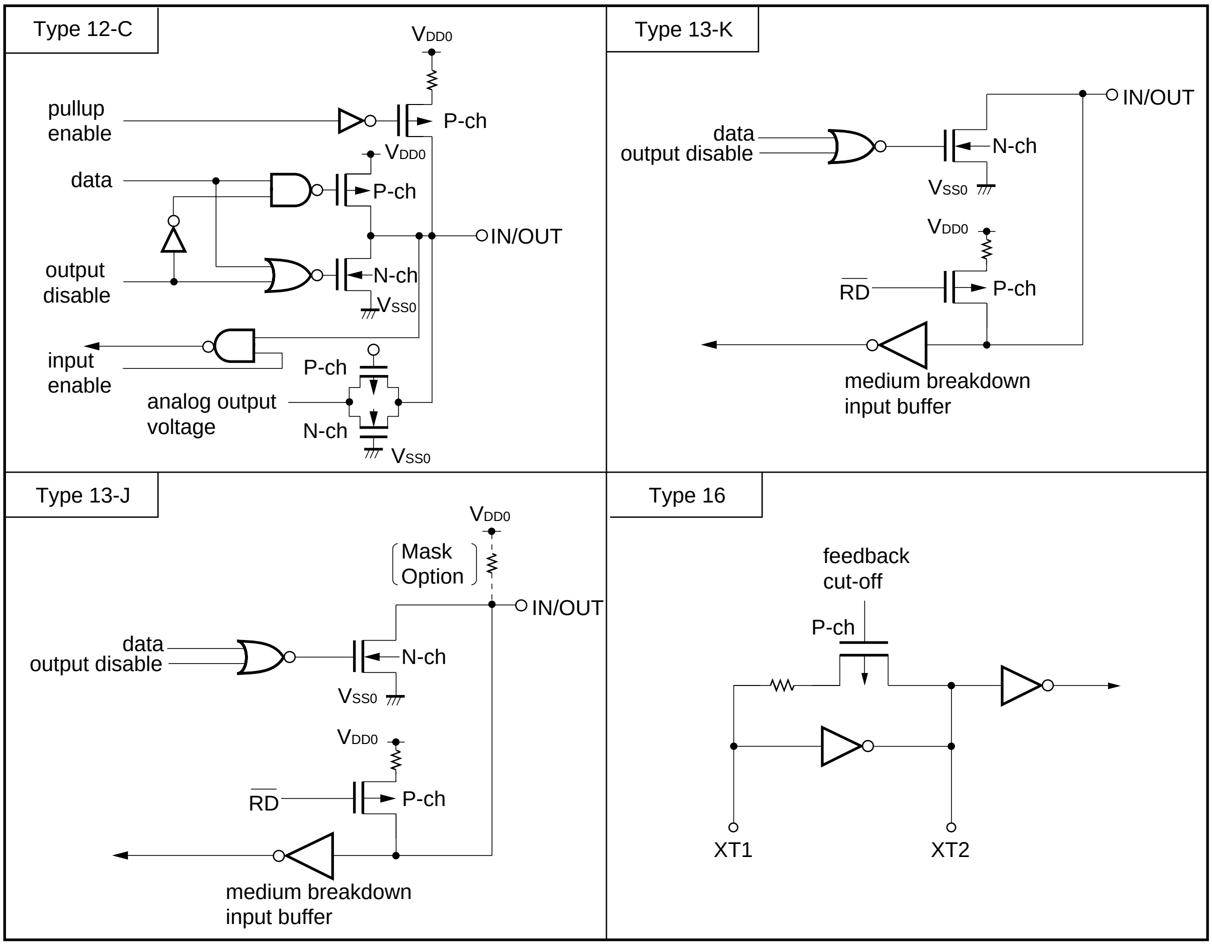


Figure 4-1. Pin Input/Output Circuit List (2/2)



Type 12-C

Type 13-K

Type 13-J

Type 16

[MEMO]

CHAPTER 5 CPU ARCHITECTURE

5.1 Memory Spaces

Figures 5-1 to 5-6 show memory maps.

Figure 5-1. Memory Map (μPD780053, 780053Y)

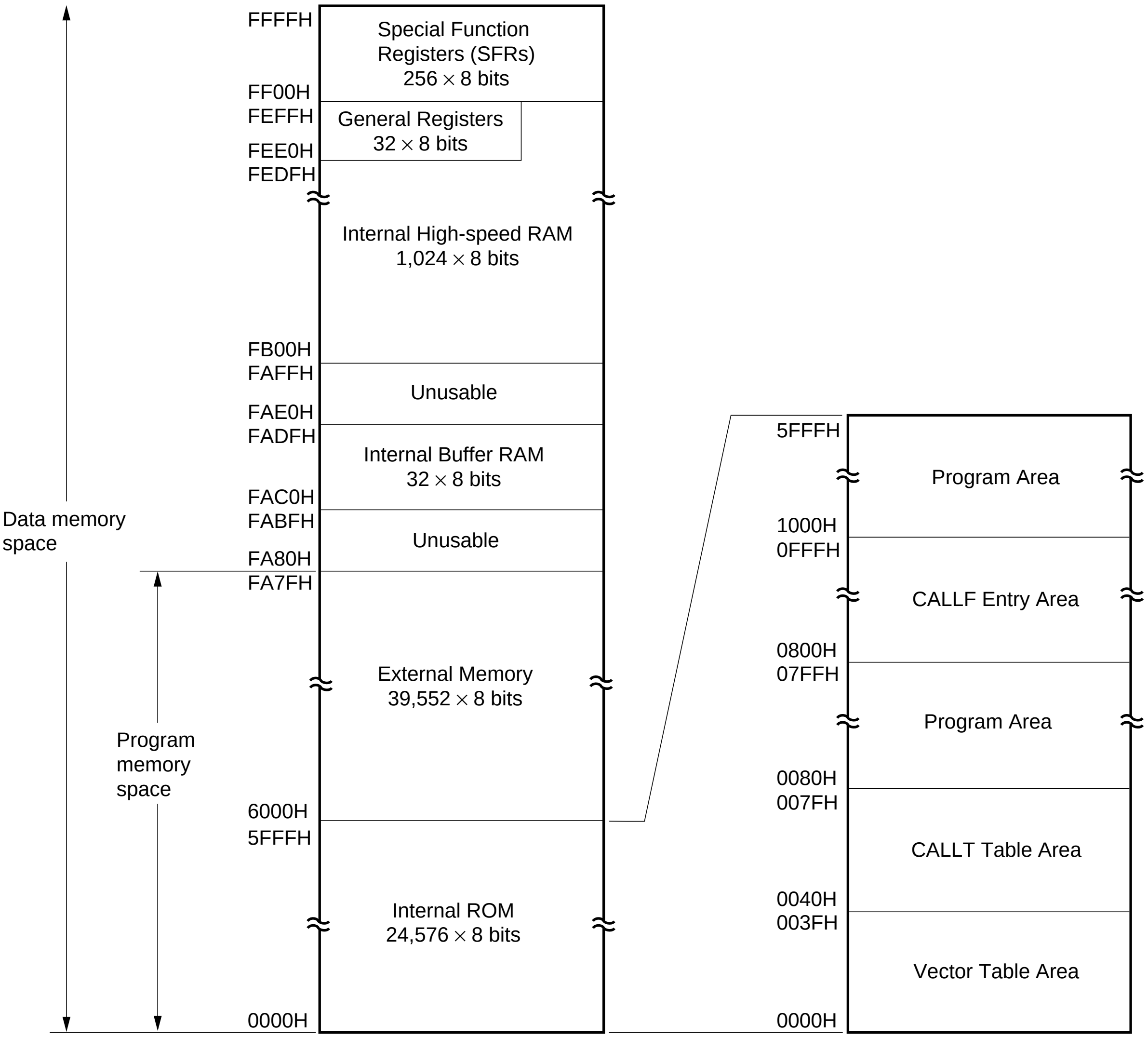


Figure 5-2. Memory Map (μPD780054, 780054Y)

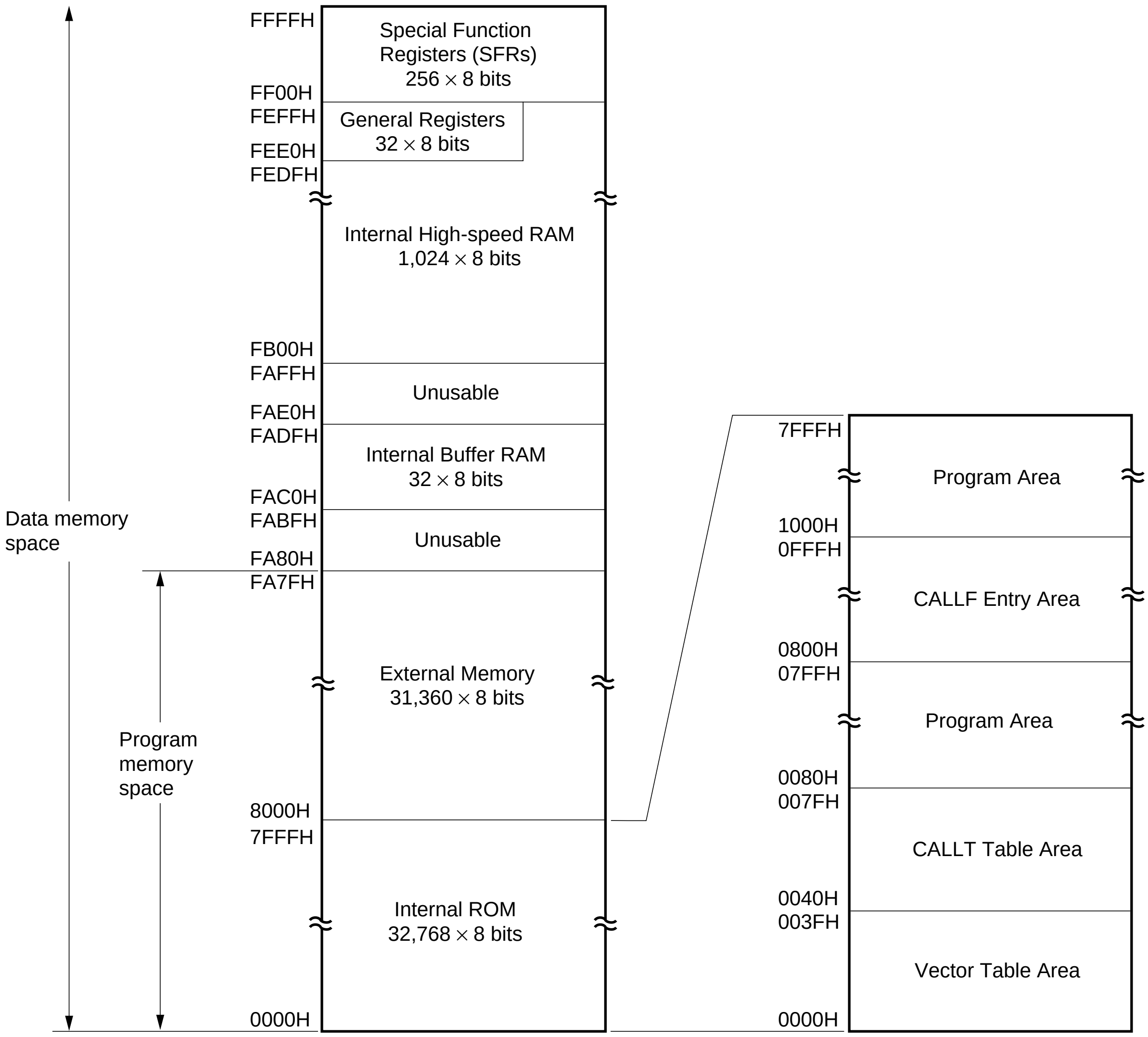


Figure 5-3. Memory Map (μPD780055, 780055Y)

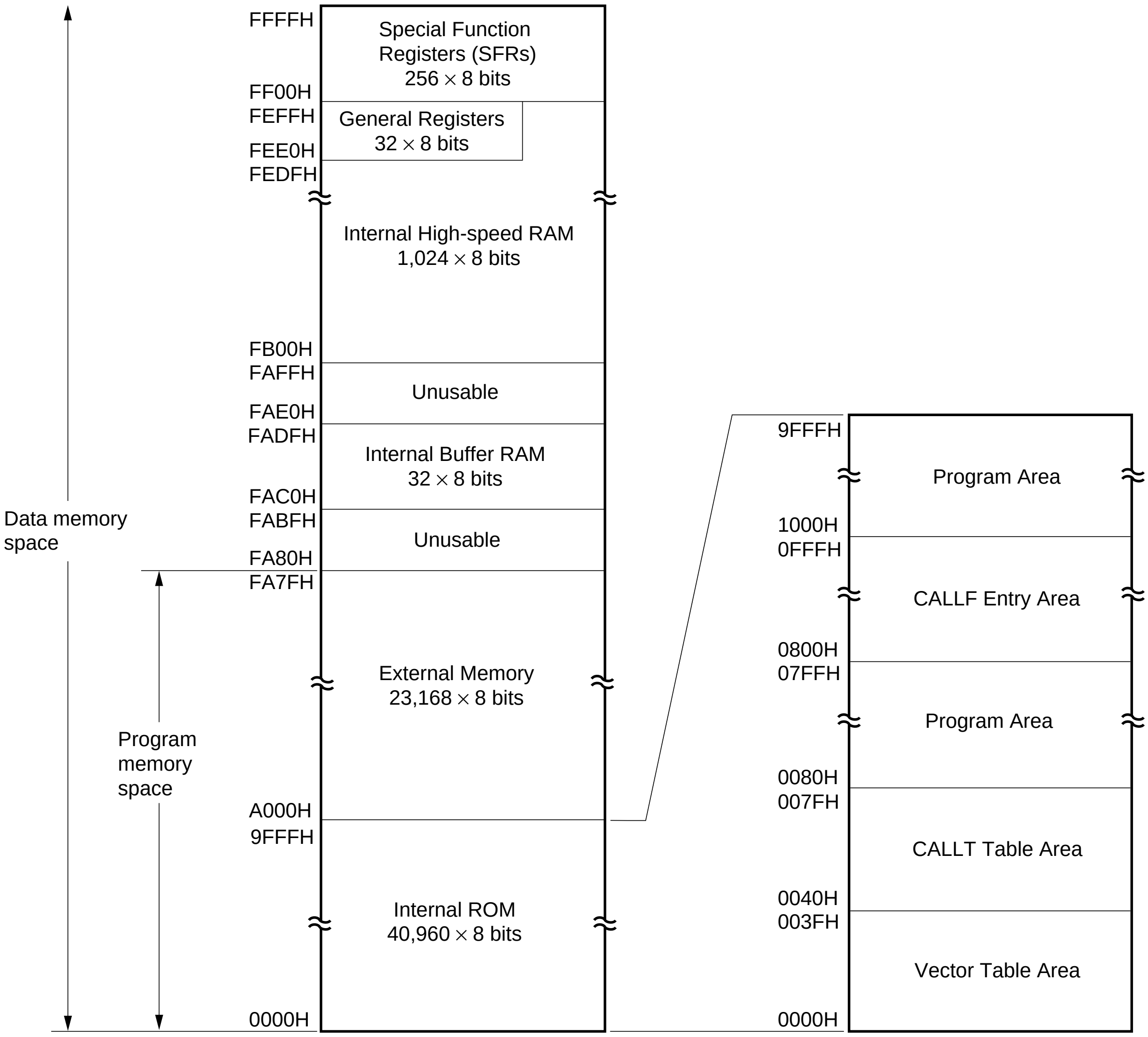


Figure 5-4. Memory Map (μPD780056, 780056Y)

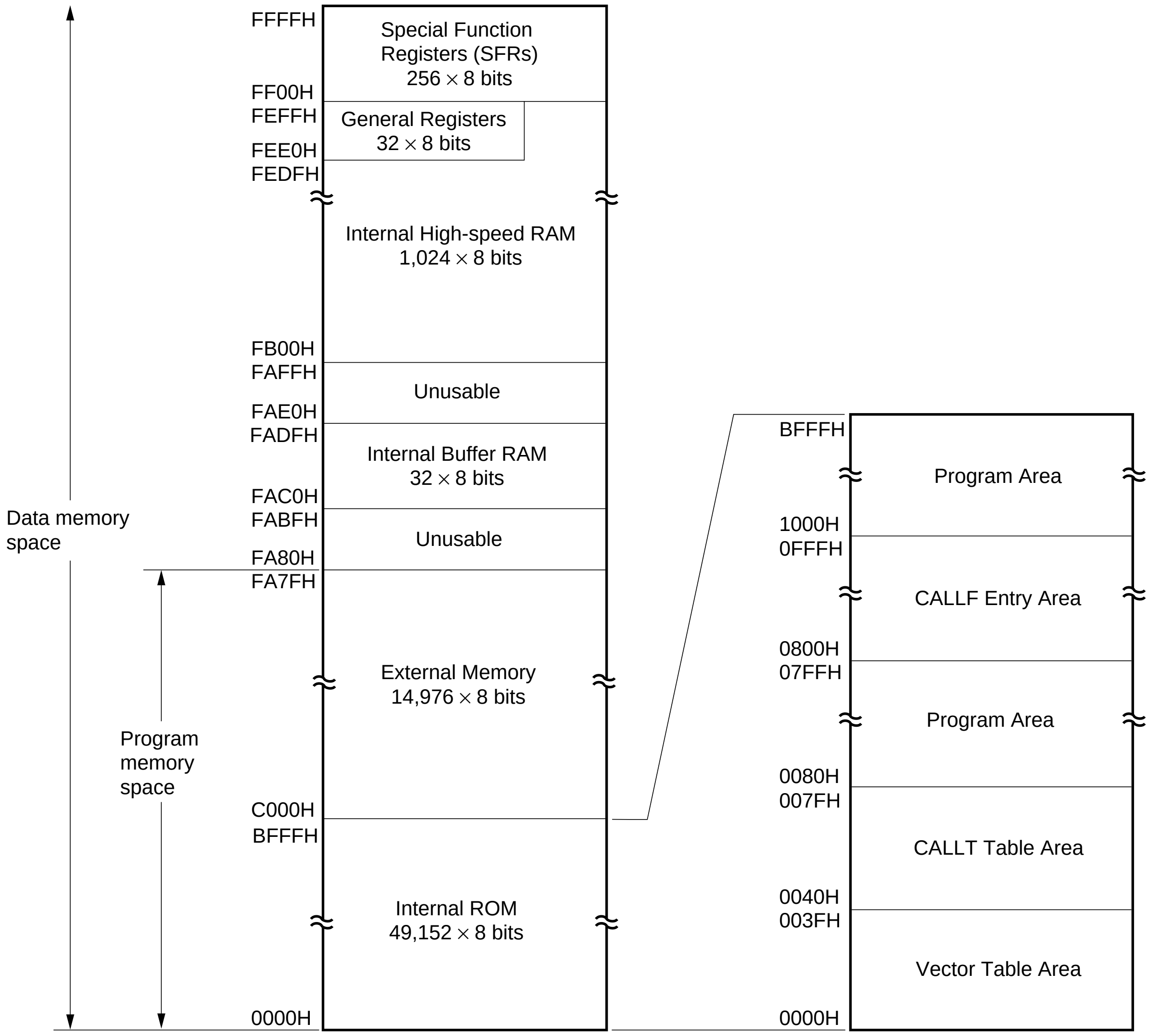
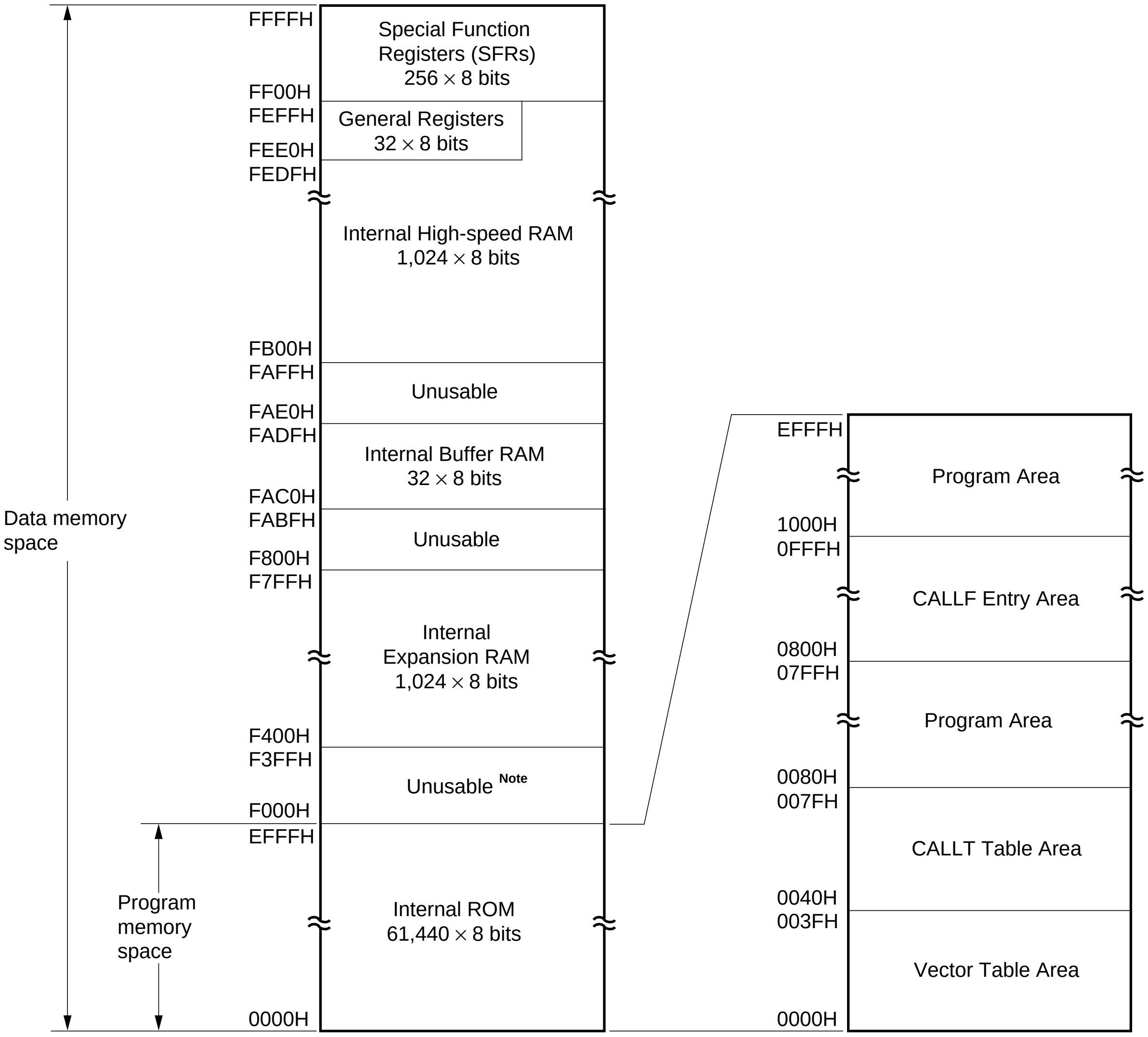
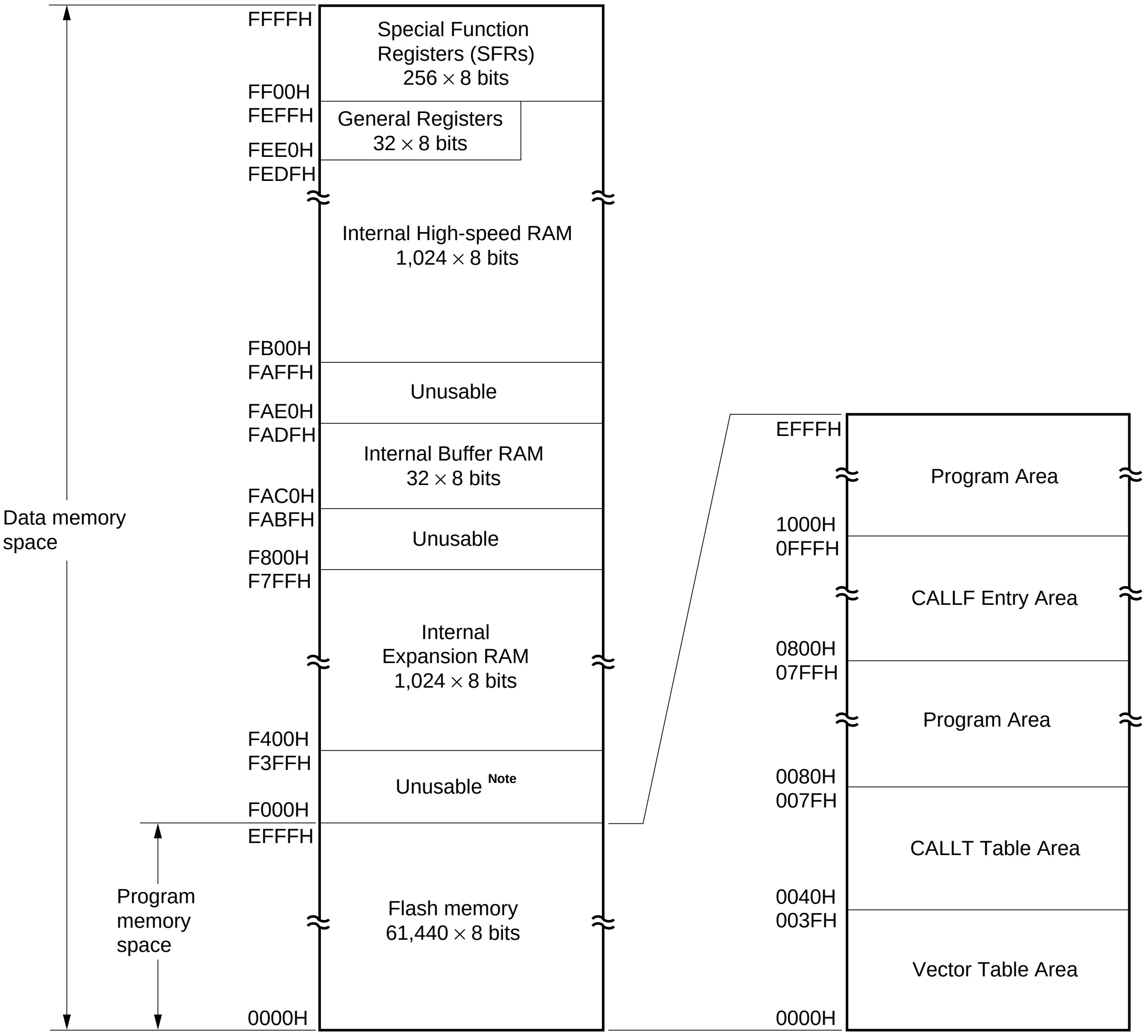


Figure 5-5. Memory Map (μPD780058, 780058Y)



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

Figure 5-6. Memory Map (μPD78F0058, 78F0058Y)



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

5.1.1 Internal program memory space

The μPD780058 and 780058Y Subseries have various sizes of internal ROM or flash memory as shown below.

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

Part Number	Internal ROM	
	Type	Capacity
μPD780053, 780053Y	Mask ROM	24,576 × 8 bits
μPD780054, 780054Y		32,768 × 8 bits
μPD780055, 780055Y		40,960 × 8 bits
μPD780056, 780056Y		49,152 × 8 bits
μPD780058, 780058Y		61,440 × 8 bits
μPD78F0058, 78F0058Y	Flash memory	61,440 × 8 bits

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

(1) Vector table area

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

Table 5-1. Vector Table

Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input
0004H	INTWDT
0006H	INTP0
0008H	INTP1
000AH	INTP2
000CH	INTP3
000EH	INTP4
0010H	INTP5
0014H	INTCSI0
0016H	INTCSI1
0018H	INTSER
001AH	INTSR/INTCSI2
001CH	INTST
001EH	INTTM3
0020H	INTTM00
0022H	INTTM01
0024H	INTTM1
0026H	INTTM2
0028H	INTAD
003EH	BRK

(2) CALLT instruction table area

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

(3) CALLF instruction entry area

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

5.1.2 Internal data memory space

The μPD780058 and 780058Y Subseries incorporate the following RAMs.

(1) Internal high-speed RAM

High-speed memory of the following configuration is incorporated:
1,024 × 8 bits (FB00H to FEFFH)
In this area, four banks of general registers, each bank consisting of eight 8-bit registers, are allocated in the 32-byte area FEE0H to FEFFH.
The internal high-speed RAM can also be used as a stack memory area.

(2) Internal buffer RAM

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

(3) Internal expansion RAM (μPD780058, 780058Y, 78F0058, 78F0058Y only)

Internal expansion RAM is allocated to the 1,024-byte area from F400H to F7FFH.

5.1.3 Special Function Register (SFR) area

An on-chip peripheral hardware special-function register (SFR) is allocated in the area FF00H to FFFFH. (Refer to **Table 5-2 Special-Function Register List** in **5.2.3 Special Function Register (SFR)**).

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

5.1.4 External memory space

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

5.1.5 Data memory addressing

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

The address of the instruction to be executed next is addressed by the program counter PC (for details, refer to **5.3 Instruction Address Addressing**).

To address the memory that is manipulated when an instruction is executed, the μPD780058, 780058Y Subseries is provided with many addressing modes with a high operability. Especially at addresses corresponding to data memory area, particular addressing modes are possible to meet the functions of the special function registers (SFRs) and general registers. This area is between FB00H and FFFFH. The data memory space is the entire 64-Kbyte space (0000H to FFFFH). Figures 5-7 to 5-12 show the data memory addressing modes. For details of each addressing, refer to **5.4 Operand Address Addressing.**

Figure 5-7. Data Memory Addressing (μPD780053, 780053Y)

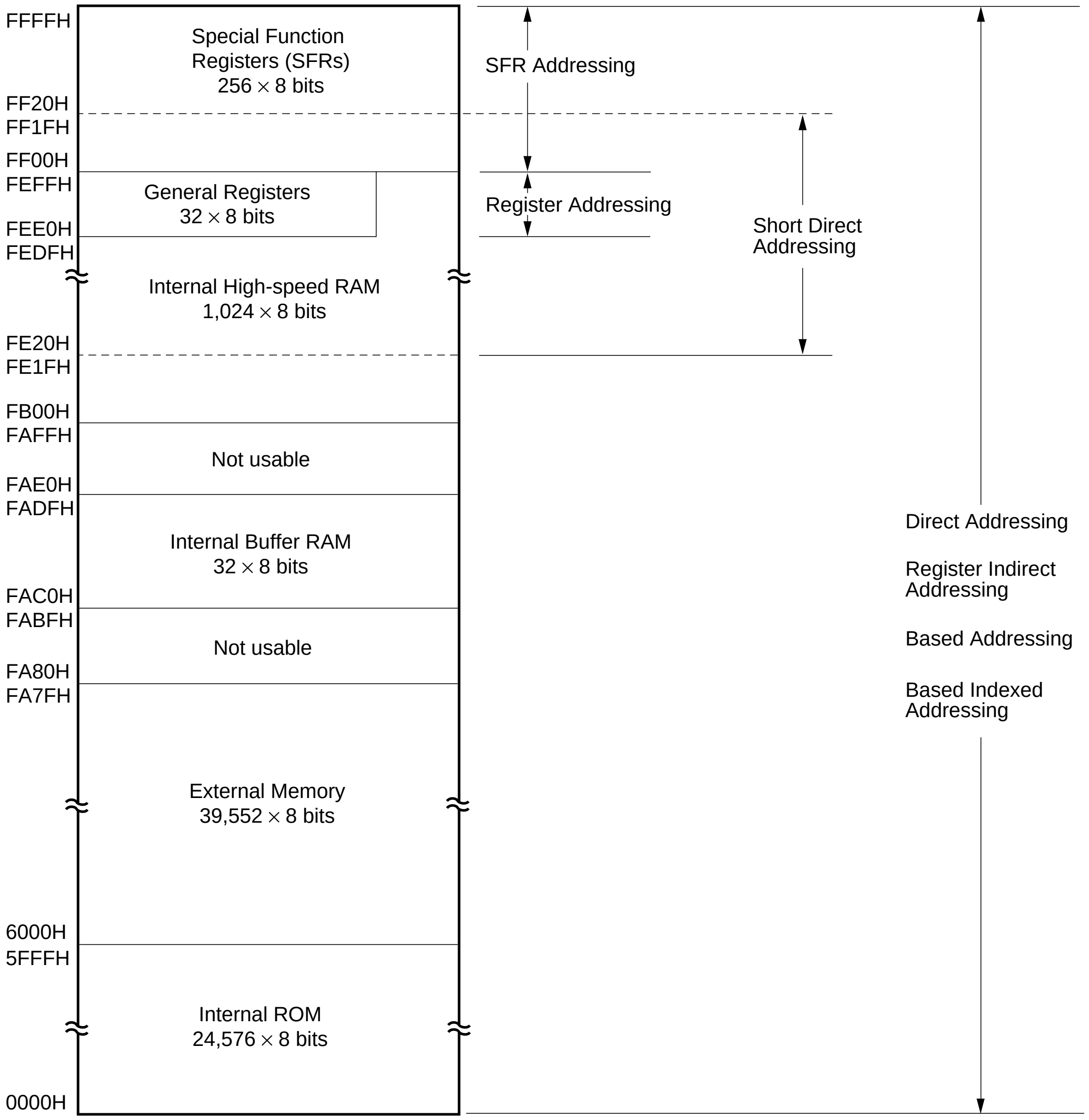


Figure 5-8. Data Memory Addressing (μPD780054, 780054Y)

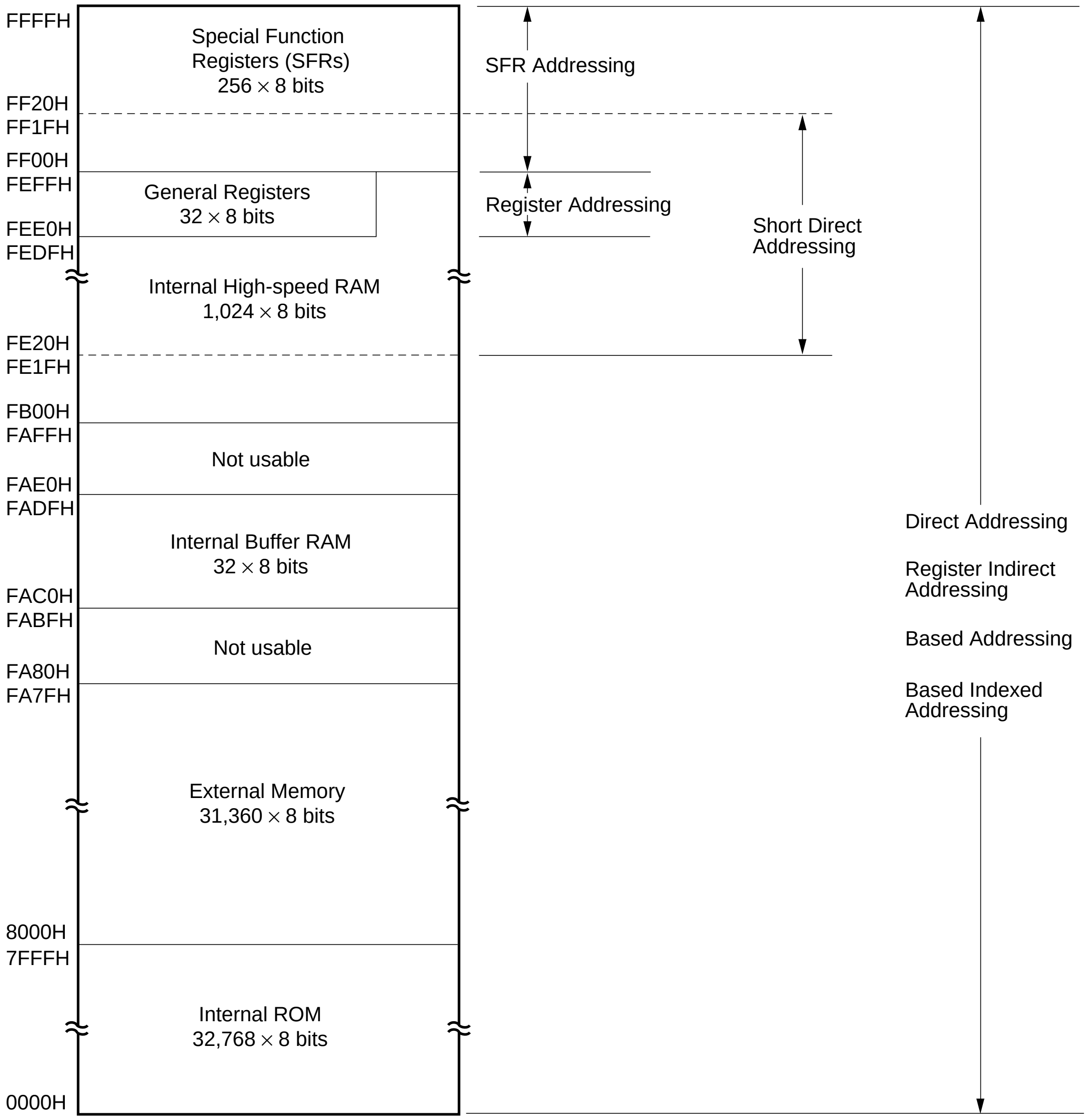


Figure 5-9. Data Memory Addressing (μPD780055, 780055Y)

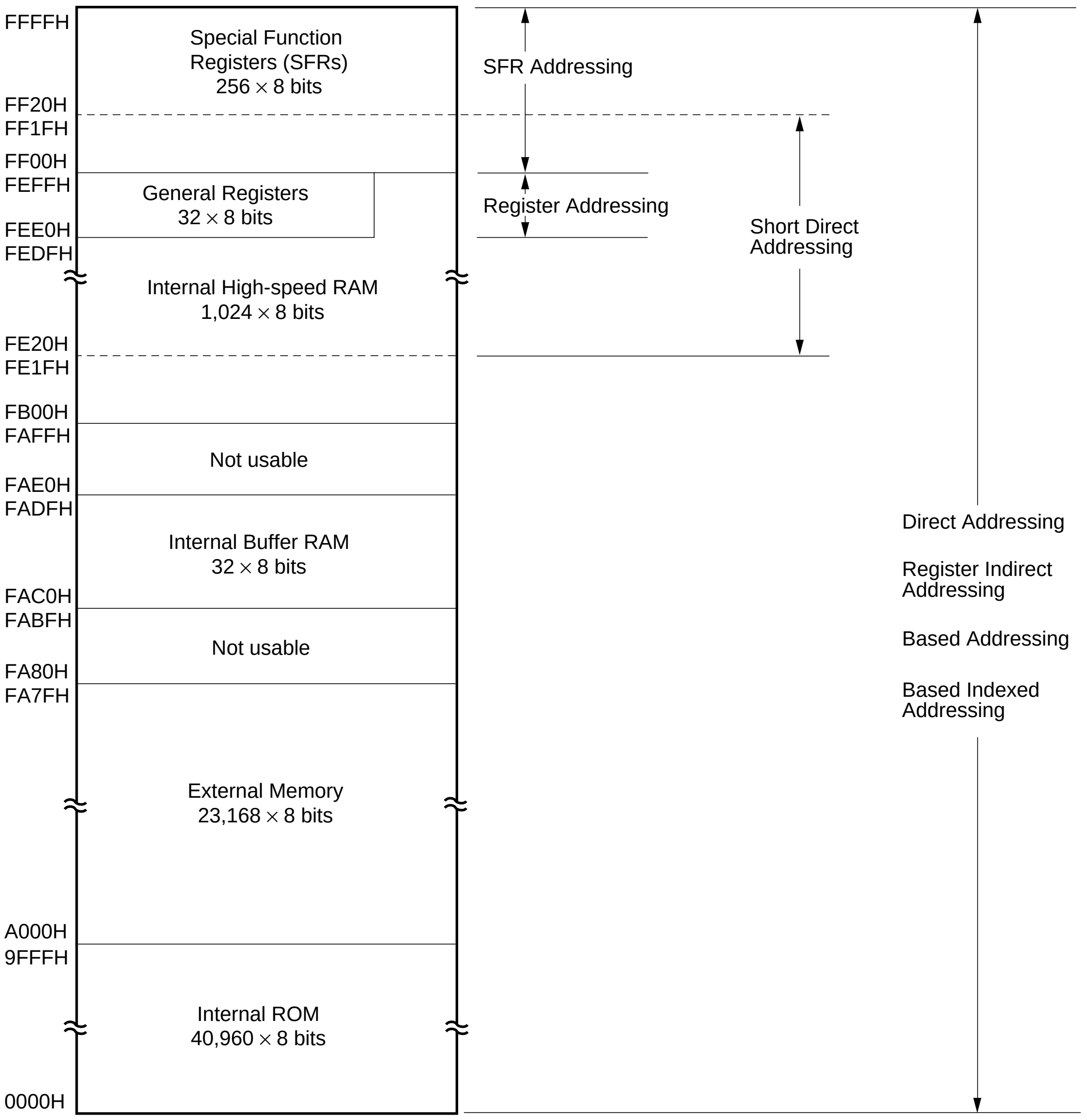


Figure 5-10. Data Memory Addressing (μPD780056, 780056Y)

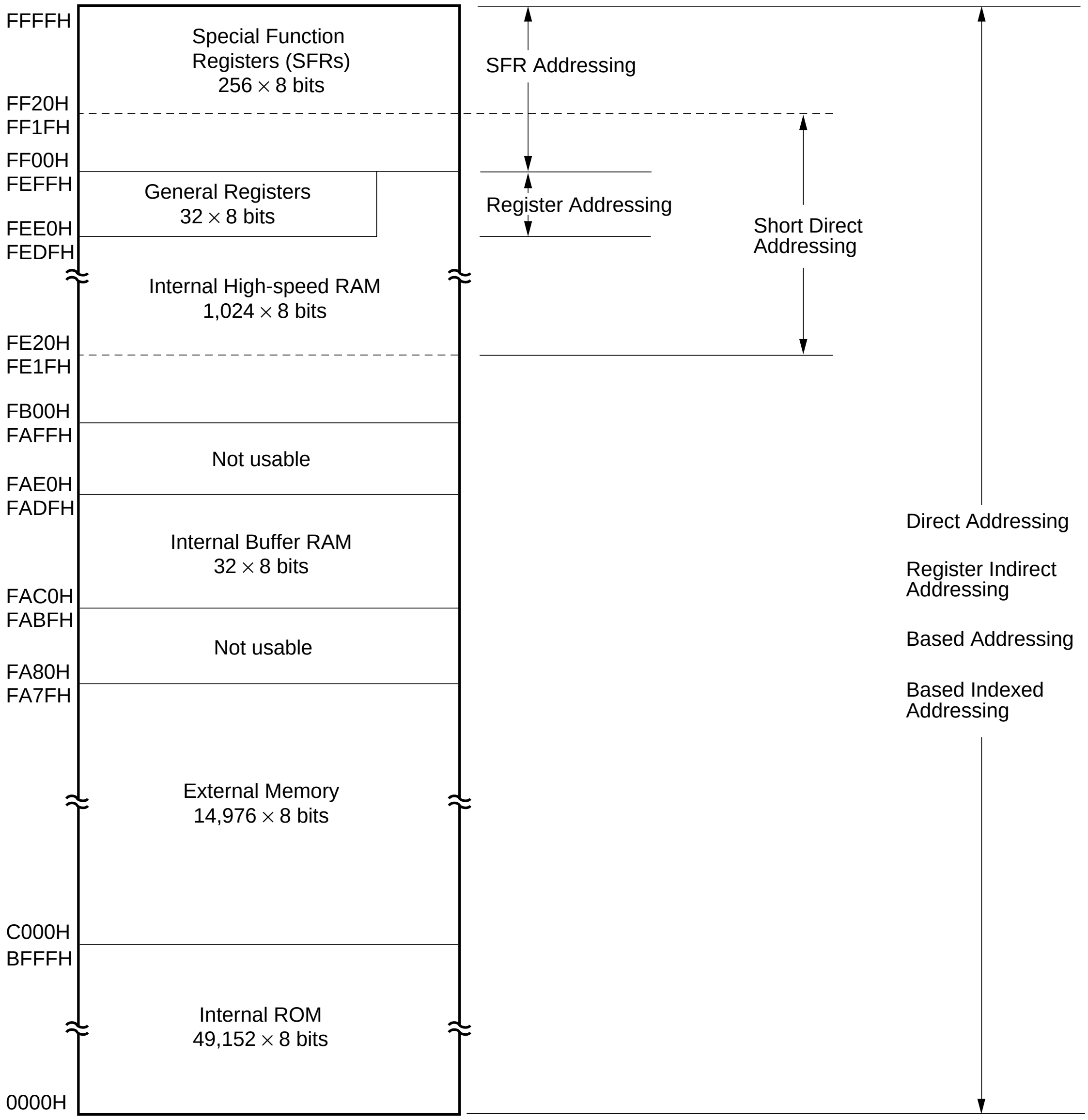
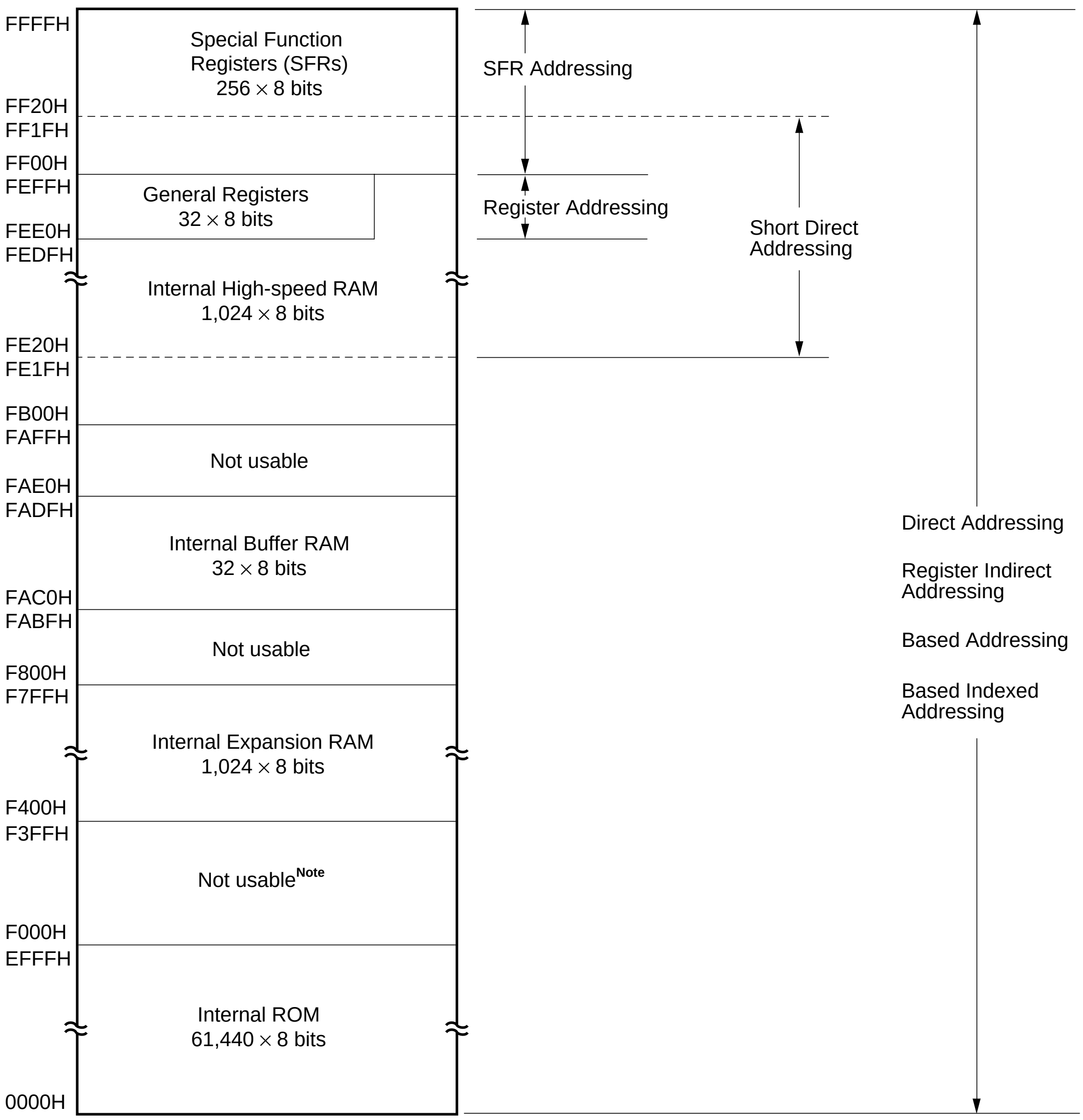
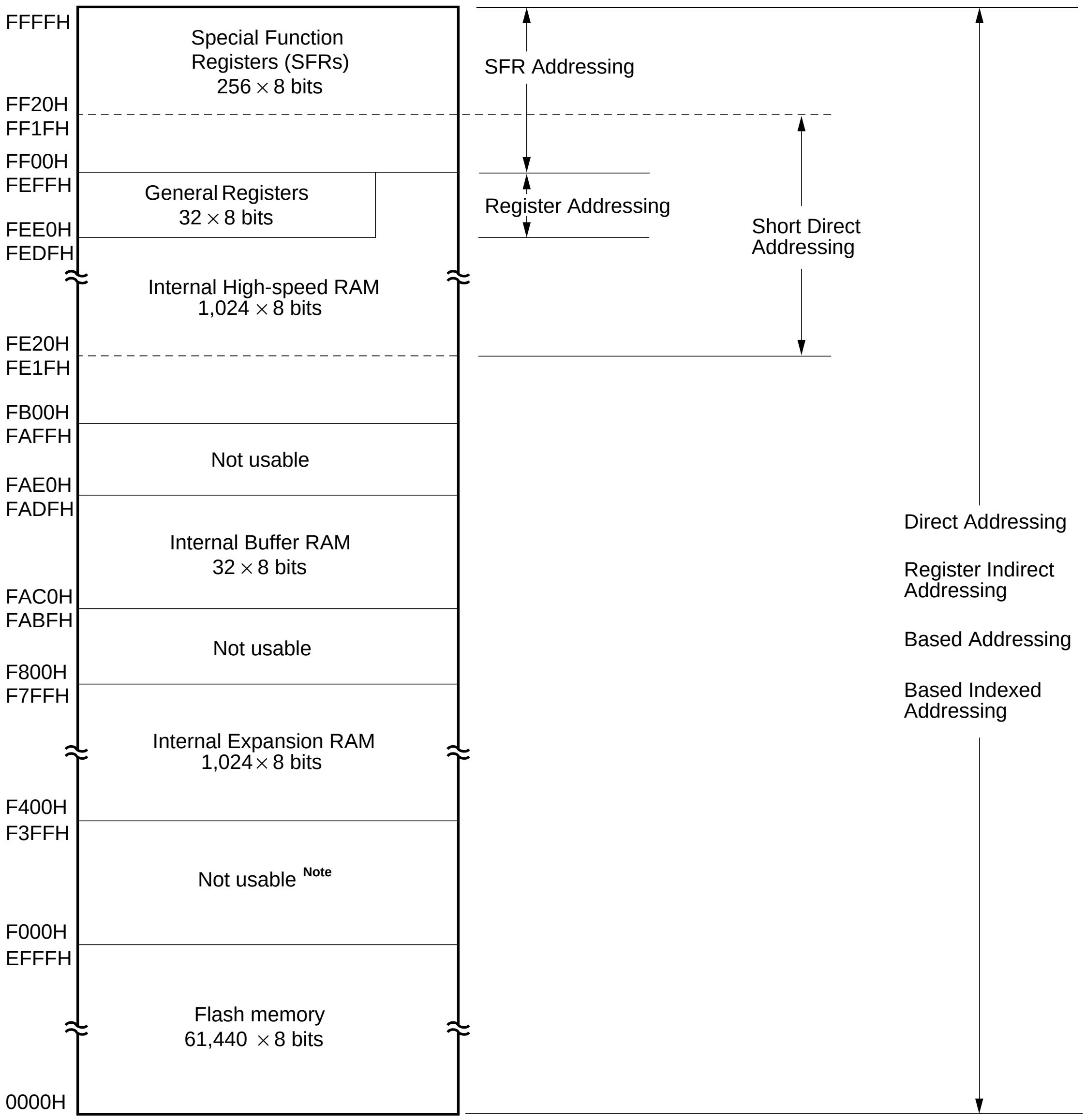


Figure 5-11. Data Memory Addressing (μPD780058, 780058Y)



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

Figure 5-12. Data Memory Addressing (μPD78F0058, 78F0058Y)



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

5.2 Processor Registers

The μPD780058 and 780058Y Subseries incorporate the following processor registers.

5.2.1 Control registers

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

(1) Program counter (PC)

The program counter is a 16-bit register which holds the address information of the next program to be executed.

In normal operation, the PC is automatically incremented according to the number of bytes of the instruction to be fetched. When a branch instruction is executed, immediate data and register contents are set.

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

Figure 5-13. 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 PSW to 02H.

Figure 5-14. 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 IE = 0, all interrupt requests except the non-maskable interrupt are disabled (DI status).

When IE = 1, interrupts are enabled (EI status). At this time, acknowledgment of interrupts is controlled with an inservice priority flag (ISP), an interrupt mask flag for various interrupt sources, and a priority specify flag.

The interrupt enable flag is reset to 0 when the DI instruction is executed or when an interrupt request is acknowledged, and set to 1 when the EI instruction is executed.

(b) Zero flag (Z)

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

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

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

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

(d) Auxiliary carry flag (AC)

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

(e) In-service priority flag (ISP)

This flag manages the priority of acknowledgeable maskable vectored interrupts. When ISP = 0, the vectored interrupt whose priority is specified by the priority specify flag registers (PR0L, PR0H, and PR1L) (refer to **21.3 (3) Priority specify flag registers (PR0L, PR0H, and PR1L)**) to be low is disabled. Whether the interrupt is actually acknowledged is controlled by the status of the interrupt enable flag (IE).

(f) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Figure 5-15. 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 5-16 and 5-17.

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

Figure 5-16. Data to Be Saved to Stack Memory

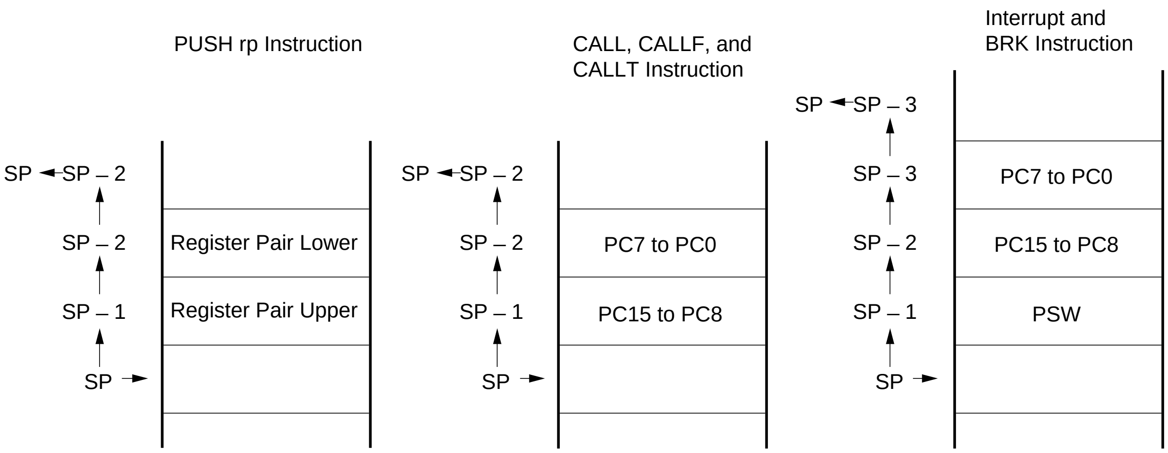
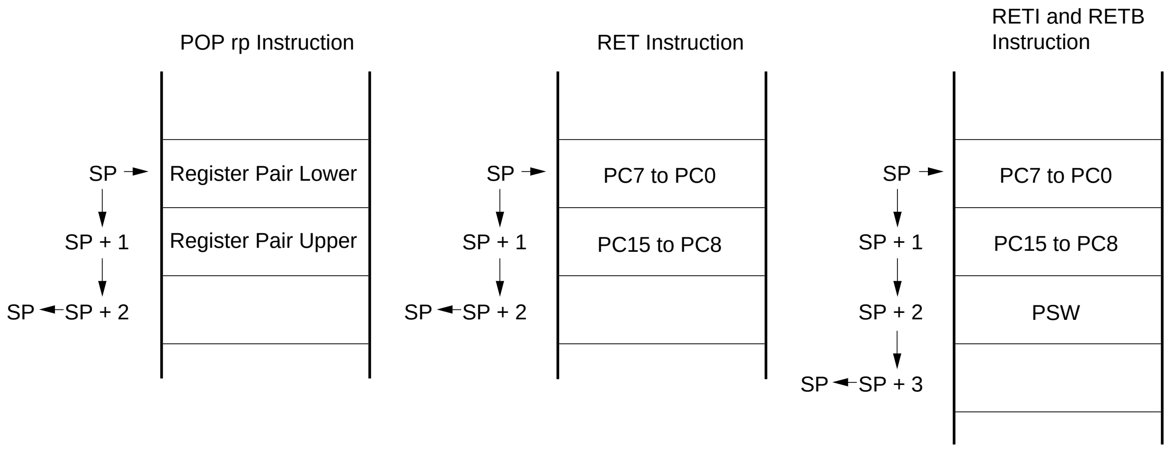


Figure 5-17. Data to Be Reset from Stack Memory



5.2.2 General registers

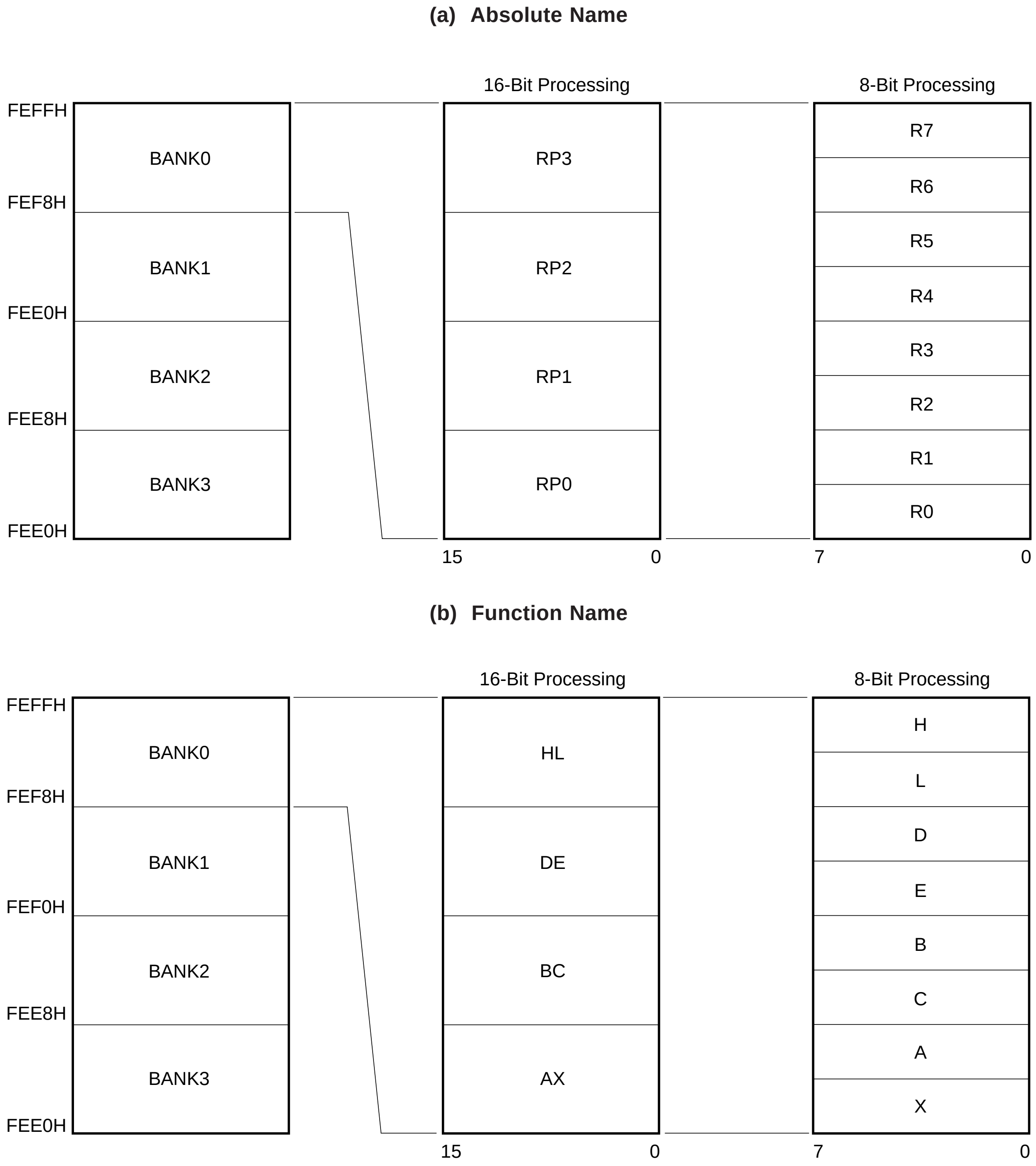
A general register is mapped at particular addresses (FEE0H to FEFFH) of the data memory. It consists of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

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

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

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

Figure 5-18. General Register Configuration



(a) Absolute Name

(b) Function Name

5.2.3 Special Function Register (SFR)

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

It is allocated in the FF00H to FFFFH area.

The special-function 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 5-2 gives a list of special-function registers. The meaning of items in the table is as follows.

- Symbol
  Symbols indicating the addresses of special function register. These symbols are reserved words for the RA78K/0 and defined by header file sfrbit.h for the CC78K/0, and can be used as the operands of instructions when
  ★ the RA78K/0, ID78K0, ID78K0-NS, and SM78K0 are used.
- R/W
  Indicates whether the corresponding special-function register can be read or written.
  R/W : Read/write enable
  R : Read only
  W : Write only
- Manipulatable bit units
  √ indicates bit units (1, 8 or 16 bits) in which the register can be manipulated. — indicates that the register cannot be manipulated in the indicated bit units.
- After reset
  Indicates each register status upon $\overline{\text{RESET}}$ input.

Table 5-2. Special-Function Register List (1/3)

Address	Special-Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 bit	8 bits	16 bits	
FF00H	Port0	P0	R/W	√	√	—	00H
FF01H	Port1	P1		√	√	—	
FF02H	Port2	P2		√	√	—	
FF03H	Port3	P3		√	√	—	
FF04H	Port4	P4		√	√	—	Undefined
FF05H	Port5	P5		√	√	—	
FF06H	Port6	P6		√	√	—	
FF07H	Port7	P7		√	√	—	00H
FF0CH	Port12	P12		√	√	—	
FF0DH	Port13	P13		√	√	—	
FF10H FF11H	Capture/compare register 00	CR00		—	—	√	Undefined
FF12H FF13H	Capture/compare register 01	CR01		—	—	√	
FF14H FF15H	16-bit timer register	TM0	R	—	—	√	0000H
FF16H	Compare register 10	CR10	R/W	—	√	—	Undefined
FF17H	Compare register 20	CR20		—	√	—	
FF18H	8-bit timer register 1	TMS TM1	R	—	√	√	00H
FF19H	8-bit timer register 2	TM2		—	√		
FF1AH	Serial I/O shift register 0	SIO0	R/W	—	√	—	Undefined
FF1BH	Serial I/O shift register 1	SIO1		—	√	—	
FF1FH	A/D conversion result register	ADCR	R	—	√	—	
FF20H	Port mode register 0	PM0	R/W	√	√	—	FFH
FF21H	Port mode register 1	PM1		√	√	—	
FF22H	Port mode register 2	PM2		√	√	—	
FF23H	Port mode register 3	PM3		√	√	—	
FF25H	Port mode register 5	PM5		√	√	—	
FF26H	Port mode register 6	PM6		√	√	—	
FF27H	Port mode register 7	PM7		√	√	—	
FF2CH	Port mode register 12	PM12		√	√	—	
FF2DH	Port mode register 13	PM13		√	√	—	
FF30H	Real-time output buffer register L	RTBL		—	√	—	00H
FF31H	Real-time output buffer register H	RTBH		—	√	—	
FF34H	Real-time output port mode register	RTPM		√	√	—	
FF36H	Real-time output port control register	RTPC		√	√	—	

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

Address	Special-Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 bit	8 bits	16 bits	
FF38H FF39H	Correction address register 0 Note	CORAD0	R/W	—	—	√	0000H
FF3AH FF3BH	Correction address register 1 Note	CORAD1		—	—	√	
FF40H	Timer clock select register 0	TCL0		√	√	—	00H
FF41H	Timer clock select register 1	TCL1		—	√	—	
FF42H	Timer clock select register 2	TCL2		—	√	—	
FF43H	Timer clock select register 3	TCL3		—	√	—	88H
FF47H	Sampling clock select register	SCS		—	√	—	00H
FF48H	16-bit timer mode control register	TMC0		√	√	—	
FF49H	8-bit timer mode control register 1	TMC1		√	√	—	
FF4AH	Watch timer mode control register	TMC2		√	√	—	
FF4CH	Capture/compare control register 0	CRC0		√	√	—	04H
FF4EH	16-bit timer output control register	TOC0		√	√	—	00H
FF4FH	8-bit timer output control register	TOC1		√	√	—	
FF60H	Serial operating mode register 0	CSIM0		√	√	—	
FF61H	Serial bus interface control register	SBIC		√	√	—	
FF62H	Slave address register	SVA		—	√	—	Undefined
FF63H	Interrupt timing specify register	SINT		√	√	—	00H
FF68H	Serial operating mode register 1	CSIM1		√	√	—	
FF69H	Automatic data transmit/receive control register	ADTC		√	√	—	
FF6AH	Automatic data transmit/receive address pointer	ADTP		—	√	—	
FF6BH	Automatic data transmit/receive interval specify register	ADTI		√	√	—	
FF70H	Asynchronous serial interface mode register	ASIM		√	√	—	
FF71H	Asynchronous serial interface status register	ASIS	R	√	√	—	
FF72H	Serial operating mode register 2	CSIM2	RW	√	√	—	
FF73H	Baud rate generator control register	BRGC		—	√	—	
FF74H	Transmit shift register	TXS SIO2	W	—	√	—	FFH
	Receive buffer register	RXB	R				
FF75H	Serial interface pin select register	SIPS	R/W	√	√	—	00H
FF80H	A/D converter mode register	ADM		√	√	—	01H
FF84H	A/D converter input select register	ADIS		—	√	—	00H
FF8AH	Correction control register Note	CORCN		√	√	—	
FF90H	D/A conversion value set register 0	DACS0		—	√	—	
FF91H	D/A conversion value set register 1	DACS1		—	√	—	
FF98H	D/A converter mode register	DAM		√	√	—	

Note This register is provided only in the μPD780058, 78F0058, 780058Y, and 78F0058Y.

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

Address	Special-Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 bit	8 bits	16 bits	
FFD0H to FFDFH	External access area Note 1			R/W	√	√	—	Undefined
FFE0H	Interrupt request flag register 0L	IF0	IF0L		√	√	√	00H
FFE1H	Interrupt request flag register 0H		IF0H		√	√		
FFE2H	Interrupt request flag register 1L	IF1L			√	√	—	
FFE4H	Interrupt mask flag register 0L	MK0	MK0L		√	√	√	FFH
FFE5H	Interrupt mask flag register 0H		MK0H		√	√		
FFE6H	Interrupt mask flag register 1L	MK1L			√	√	—	
FFE8H	Priority order specify flag register 0L	PR0	PR0L		√	√	√	
FFE9H	Priority order specify flag register 0H		PR0H		√	√		
FFEAH	Priority order specify flag register 1L	PR1L			√	√	—	
FFECH	External interrupt mode register 0	INTM0			—	√	—	00H
FFEDH	External interrupt mode register 1	INTM1			—	√	—	
FFF0H	Memory size switching register	IMS			—	√	—	Note 2
FFF2H	Oscillation mode selection register	OSMS		W	—	√	—	00H
FFF3H	Pull-up resistor option register H	PUOH		R/W	√	√	—	
FFF4H	Internal expansion RAM size switching register Note 3	IXS		W	—	√	—	0AH
FFF6H	Key return mode register	KRM		R/W	√	√	—	02H
FFF7H	Pull-up resistor option register L	PUOL			√	√	—	00H
FFF8H	Memory expansion mode register	MM			√	√	—	10H
FFF9H	Watchdog timer mode register	WDTM			√	√	—	00H
FFFAH	Oscillation stabilization time select register	OSTS			—	√	—	04H
FFFBH	Processor clock control register	PCC			√	√	—	

Notes **1.** The external access area cannot be accessed in SFR addressing. Access the area with direct addressing.

2. The value after reset depends on products.
μPD780053, 780053Y: C6H, μPD780054, 780054Y: C8H, μPD780055, 780055Y: CAH,
μPD780056, 780056Y: CCH, μPD780058, 780058Y: CFH, μPD78F0058, 78F0058Y: CFH

3. This register is provided only in the μPD780058, 780058Y, 78F0058, and 78F0058Y.

5.3 Instruction Address Addressing

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

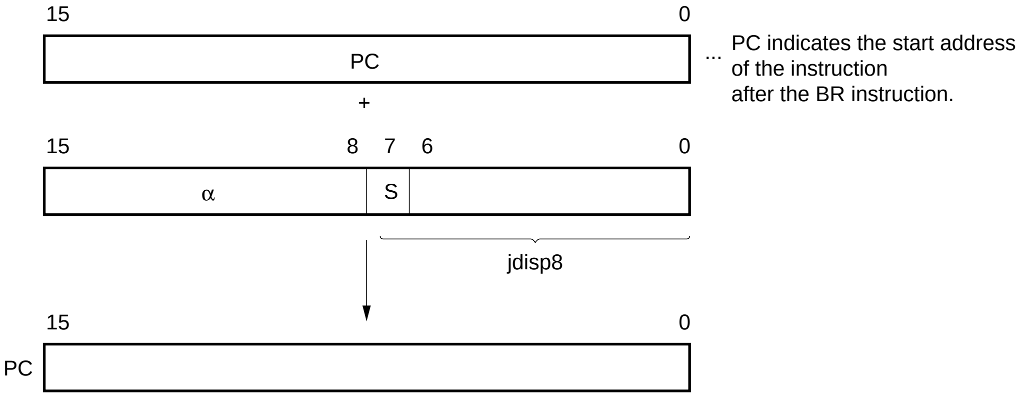
5.3.1 Relative addressing

[Function]

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

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

[Illustration]


15
0
PC
...
PC indicates the start address of the instruction after the BR instruction.
+
15
8
7
6
0
α
S
jdisp8
15
0
PC

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

5.3.2 Immediate addressing

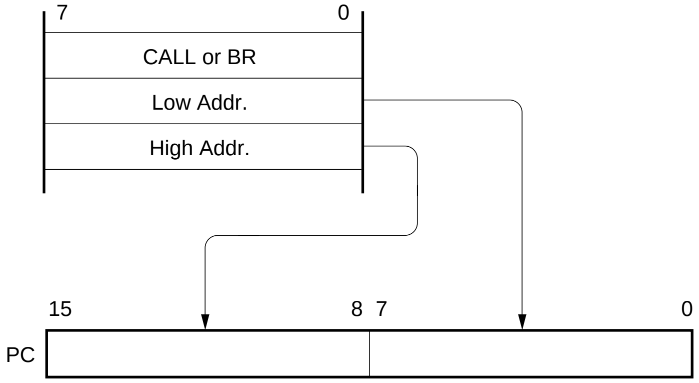
[Function]

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

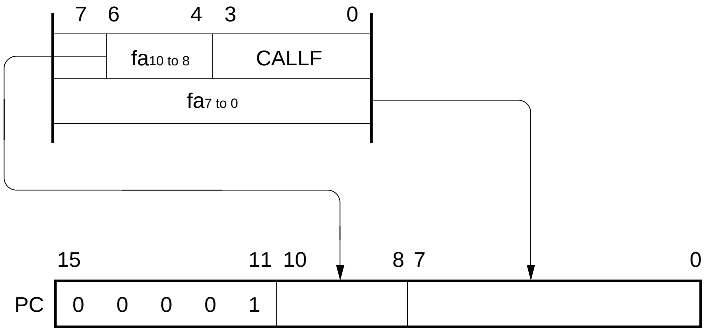
This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed. The CALL !addr16 and BR !addr16 instruction can branch in the entire memory space. The CALLF !addr11 instruction branches to an area of addresses 0800H through 0FFFH.

[Illustration]

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



In the case of CALLF !addr11 instruction



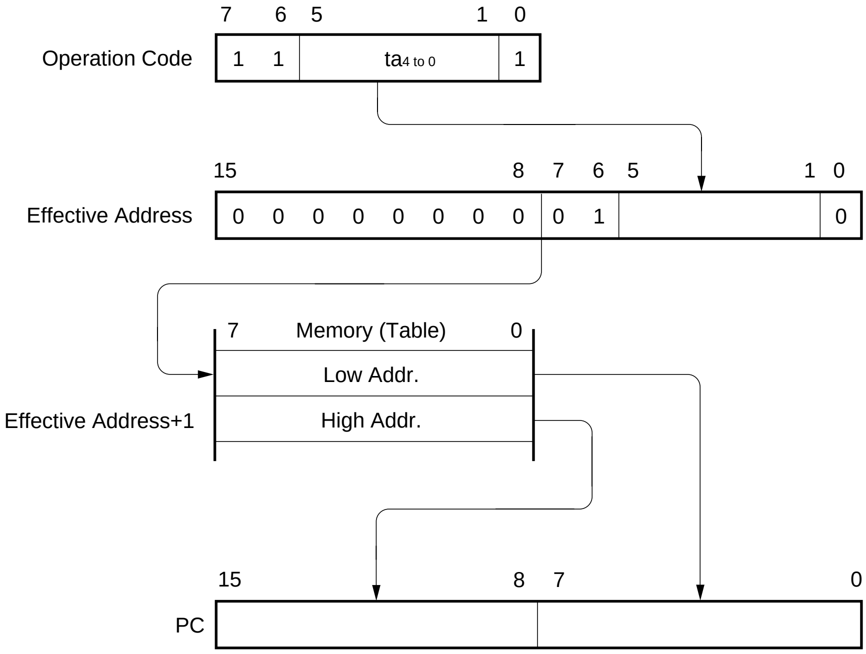
5.3.3 Table indirect addressing

[Function]

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

Before the CALLT [addr5] instruction is executed, table indirect addressing is performed. This instruction references an address stored in the memory table at addresses 40H through 7FH, and can branch in the entire memory space.

[Illustration]



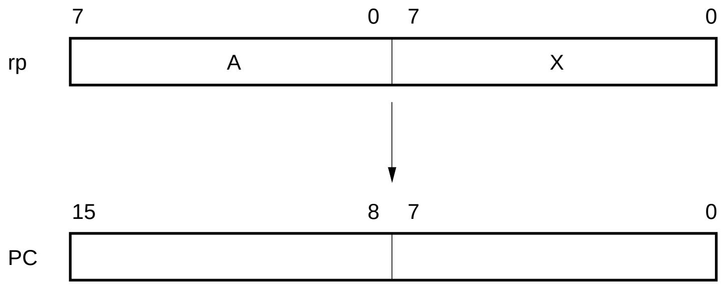
5.3.4 Register addressing

[Function]

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

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

[Illustration]



5.4 Operand Address Addressing

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

5.4.1 Implied addressing

[Function]

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

In the μPD780058 and 780058Y 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.

5.4.2 Register addressing

[Function]

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

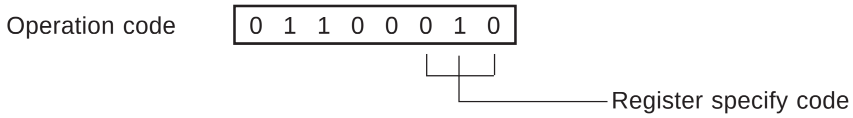
[Operand format]

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

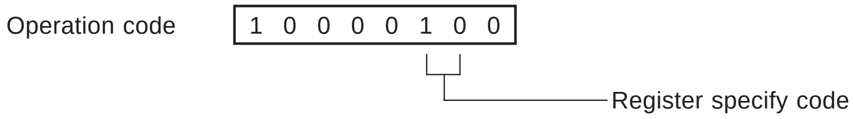
'r' and 'rp' can be described with function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) as well as absolute names (R0 to R7 and RP0 to RP3).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



5.4.3 Direct addressing

[Function]

This addressing directly addresses the memory indicated by the immediate data in an instruction word.

[Operand format]

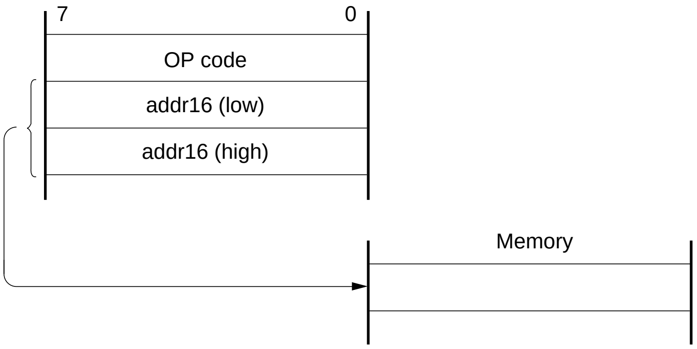
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

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

[Illustration]



5.4.4 Short direct addressing

[Function]

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

The SFR area (FF00H through FF1FH) to which short direct addressing is applied is a part of the entire SFR area. To this area, ports frequently accessed by the program, and the compare registers and capture registers of timer/event counters are mapped. These SFRs can be manipulated with a short byte length and a few clocks. When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is set to 0. When it is at 00H to 1FH, bit 8 is set to 1. Refer to **[Illustration]** on next page.

[Operand format]

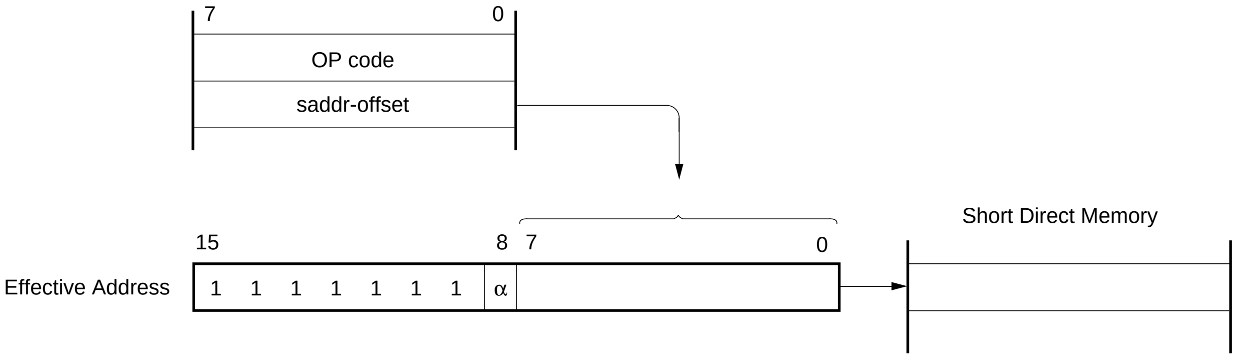
Identifier	Description
saddr	Label or immediate data of FE20H to FF1FH
saddrp	Label or immediate data of FE20H to FF1FH (even address only)

[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

5.4.5 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]

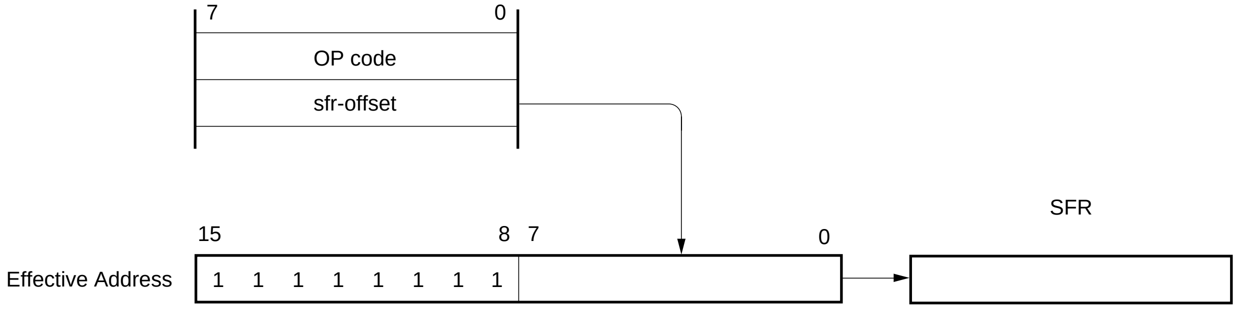
Identifier	Description
sfr	Special-function register name
sfrp	16-bit manipulatable special-function register name (even address only)

[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]



5.4.6 Register indirect addressing

[Function]

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

[Operand format]

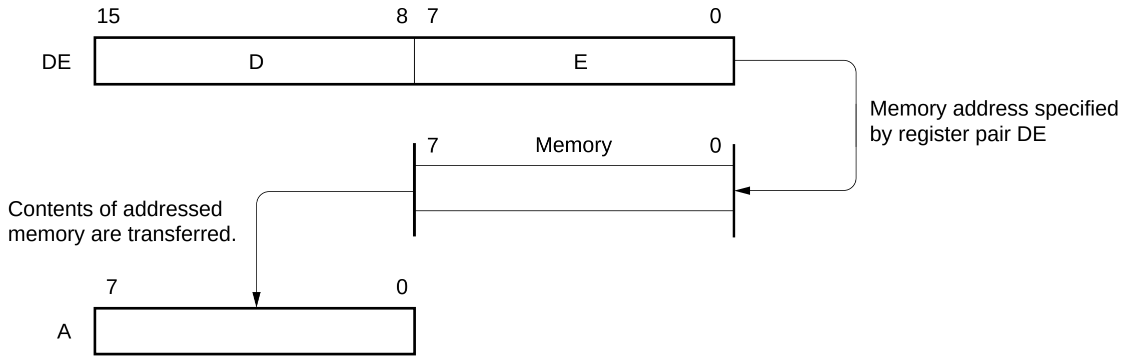
Identifier	Description
—	[DE], [HL]

[Description example]

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

Operation code 1 0 0 0 0 1 0 1

[Illustration]


15
8
7
0
DE
D
E
Memory address specified by register pair DE
7
Memory
0
Contents of addressed memory are transferred.
7
0
A

5.4.7 Based addressing

[Function]

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

[Operand format]

Identifier	Description
—	[HL + byte]

[Description example]

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

Operation code 1 0 1 0 1 1 1 0

0 0 0 1 0 0 0 0

5.4.8 Based indexed addressing

[Function]

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

[Operand format]

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

[Description example]

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

Operation code 1 0 1 0 1 0 1 1

5.4.9 Stack addressing

[Function]

The stack area is indirectly addressed with the stack pointer (SP) contents.
This addressing method is automatically employed when the PUSH, POP, subroutine call and return instructions are executed or the register is saved/reset upon generation of an interrupt request.
Stack addressing enables to address the internal high-speed RAM area only.

[Description example]

In the case of PUSH DE

Operation code 1 0 1 1 0 1 0 1

CHAPTER 6 PORT FUNCTIONS

6.1 Port Functions

The μPD780058 and 780058Y Subseries units incorporate two input ports and sixty-six input/output ports. Figure 6-1 shows the port type. Every port is capable of 1-bit and 8-bit manipulations and can carry out considerably varied control operations. Besides port functions, the ports can also serve as on-chip hardware input/output pins.

Figure 6-1. Port Types

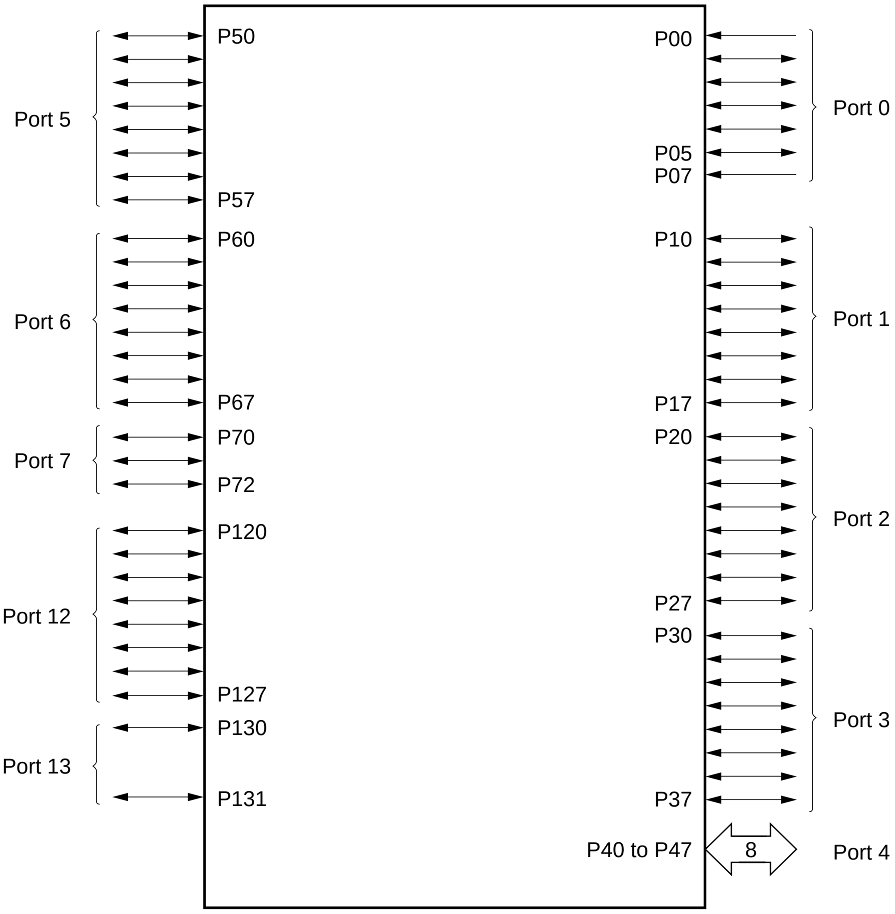


Table 6-1. Port Functions (μPD780058 Subseries) (1/2)

Pin Name	Function		Alternate Function
P00	Port 0 7-bit input/output port	Input only	INTP0/TI00
P01		Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.	INTP1/TI01
P02			INTP2
P03			INTP3
P04			INTP4
P05			INTP5
P07		Input only	XT1
P10 to P17	Port 1 8-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.		ANI0 to ANI7
P20	Port 2 8-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.		SI1
P21			SO1
P22			$\overline{\text{SCK1}}$
P23			STB/TxD1
P24			BUSY/RxD1
P25			SI0/SB0
P26			SO0/SB1
P27			$\overline{\text{SCK0}}$
P30	Port 3 8-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.		TO0
P31			TO1
P32			TO2
P33			TI1
P34			TI2
P35			PCL
P36			BUZ
P37			—
P40 to P47	Port 4 8-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software. Test input flag (KRIF) is set to 1 by falling edge detection.		AD0 to AD7
P50 to P57	Port 5 8-bit input/output port LED can be driven directly. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.		A8 to A15

Table 6-1. Port Functions (μPD780058 Subseries) (2/2)

Pin Name	Function		Alternate Function
P60	Port 6 8-bit input/output port Input/output mode can be specified bit-wise.	N-ch open-drain input/output port. On-chip pull-up resistor can be specified by mask option. (Mask ROM version only). LEDs can be driven directly.	—
P61			
P62			
P63			
P64		If used as an input port, an on-chip pull-up resistor can be used by software.	$\overline{\text{RD}}$
P65			$\overline{\text{WR}}$
P66			$\overline{\text{WAIT}}$
P67			ASTB
P70	Port 7 3-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.		SI2/RxD0
P71			SO2/TxD0
P72			$\overline{\text{SCK2}}$/ASCK
P120 to P127	Port 12 8-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, on-chip pull-up resistor can be used by software.		RTP0 to RTP7
P130 and P131	Port 13 2-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, on-chip pull-up resistor can be used by software.		ANO0, ANO1

Table 6-2. Port Functions (μPD780058Y Subseries) (1/2)

Pin Name	Function		Alternate Function
P00	Port 0 7-bit input/output port	Input only	INTP0/TI00
P01		Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.	INTP1/TI01
P02			INTP2
P03			INTP3
P04			INTP4
P05			INTP5
P07		Input only	XT1
P10 to P17	Port 1 8-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.		ANI0 to ANI7
P20	Port 2 8-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.		SI1
P21			SO1
P22			$\overline{\text{SCK1}}$
P23			STB/TxD1
P24			BUSY/RxD1
P25			SI0/SB0/SDA0
P26			SO0/SB1/SDA1
P27			$\overline{\text{SCK0}}$/SCL
P30	Port 3 8-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.		TO0
P31			TO1
P32			TO2
P33			TI1
P34			TI2
P35			PCL
P36			BUZ
P37			—
P40 to P47	Port 4 8-bit input/output port Input/output mode can be specified 8-bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software. Test input flag (KRIF) is set to 1 by falling edge detection.		AD0 to AD7
P50 to P57	Port 5 8-bit input/output port LEDs can be driven directly. Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.		A8 to A15

Table 6-2. Port Functions (μPD780058Y Subseries) (2/2)

Pin Name	Function		Alternate Function
P60	Port 6 8-bit input/output port Input/output mode can be specified bit-wise.	N-ch open-drain input/output port. On-chip pull-up resistor can be specified by mask option. (Mask ROM version only). LEDs can be driven directly.	—
P61			
P62			
P63			
P64		If used as an input port, an on-chip pull-up resistor can be used by software.	$\overline{\text{RD}}$
P65			$\overline{\text{WR}}$
P66			$\overline{\text{WAIT}}$
P67			ASTB
P70	Port 7 3-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, an on-chip pull-up resistor can be used by software.		SI2/RxD0
P71			SO2/TxD0
P72			$\overline{\text{SCK2}}$/ASCK
P120 to P127	Port 12 8-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, on-chip pull-up resistor can be used by software.		RTP0 to RTP7
P130 and P131	Port 13 2-bit input/output port Input/output mode can be specified bit-wise. If used as an input port, on-chip pull-up resistor can be used by software.		ANO0, ANO1

6.2 Port Configuration

A port consists of the following hardware:

Table 6-3. Port Configuration

Item	Configuration
Control register	Port mode register (PMm: m = 0 to 3, 5 to 10, 12, 13) Pull-up resistor option register (PUOH, PUOL) Memory expansion mode register (MM) Note Key return mode register (KRM)
Port	Total: 68 (Input: 2, I/O: 66)
Pull-up resistor	• Mask ROM version Total: 66 (software specifiable: 62, mask option: 4) • Flash memory version Total: 62

Note MM specifies port 4 input/output.

6.2.1 Port 0

Port 0 is an 7-bit input/output port with output latch. P01 to P05 pins can specify the input mode/output mode in 1-bit units with the port mode register 0 (PM0). P00 and P07 pins are input-only ports. When P01 to P05 pins are used as input ports, an on-chip pull-up resistor can be used to them in 6-bit units with a pull-up resistor option register L (PUOL).

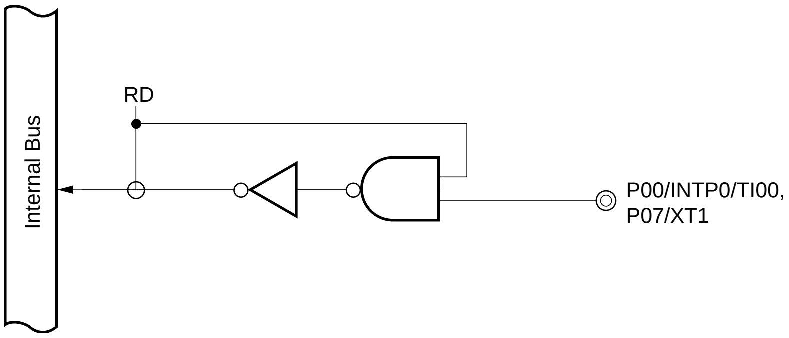
Alternate functions include external interrupt request input, external count clock input to the timer and crystal connection for subsystem clock oscillation.

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

Figures 6-2 and 6-3 show block diagrams of port0.

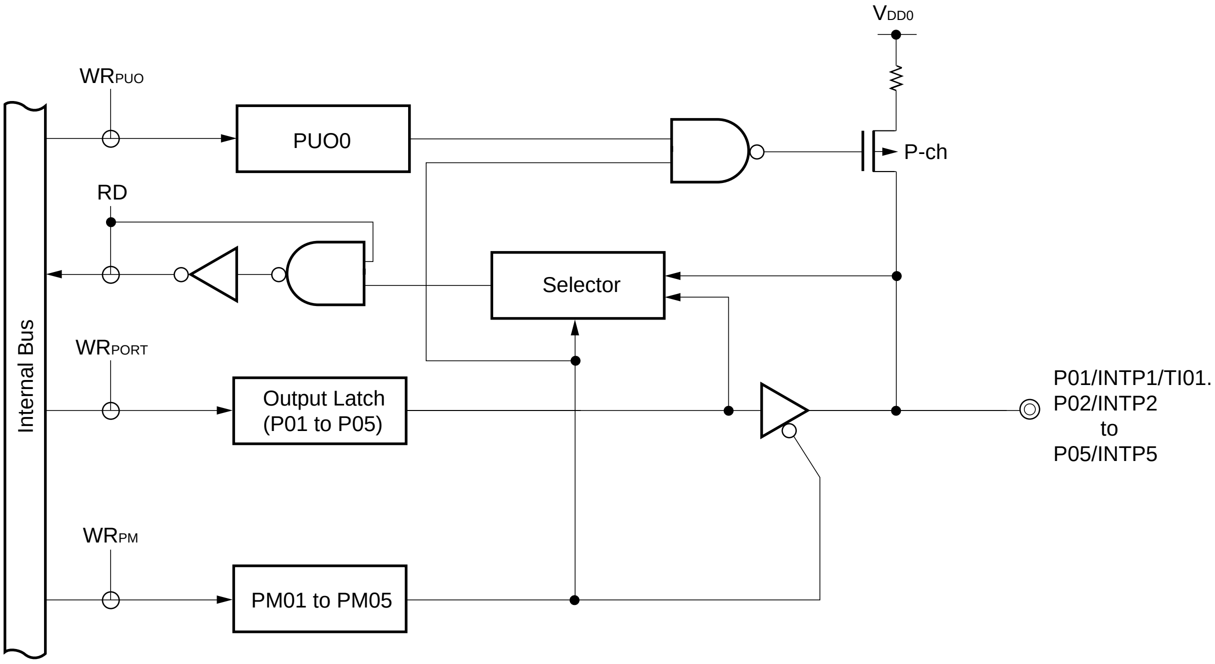
Caution Because port 0 also serves for external interrupt request input, when the port function output mode is specified and the output level is changed, the interrupt request flag is set. Thus, when the output mode is used, set the interrupt mask flag to 1.

Figure 6-2. P00 and P07 Block Diagram



RD : Port 0 read signal

Figure 6-3. P01 to P05 Block Diagram



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

6.2.2 Port 1

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

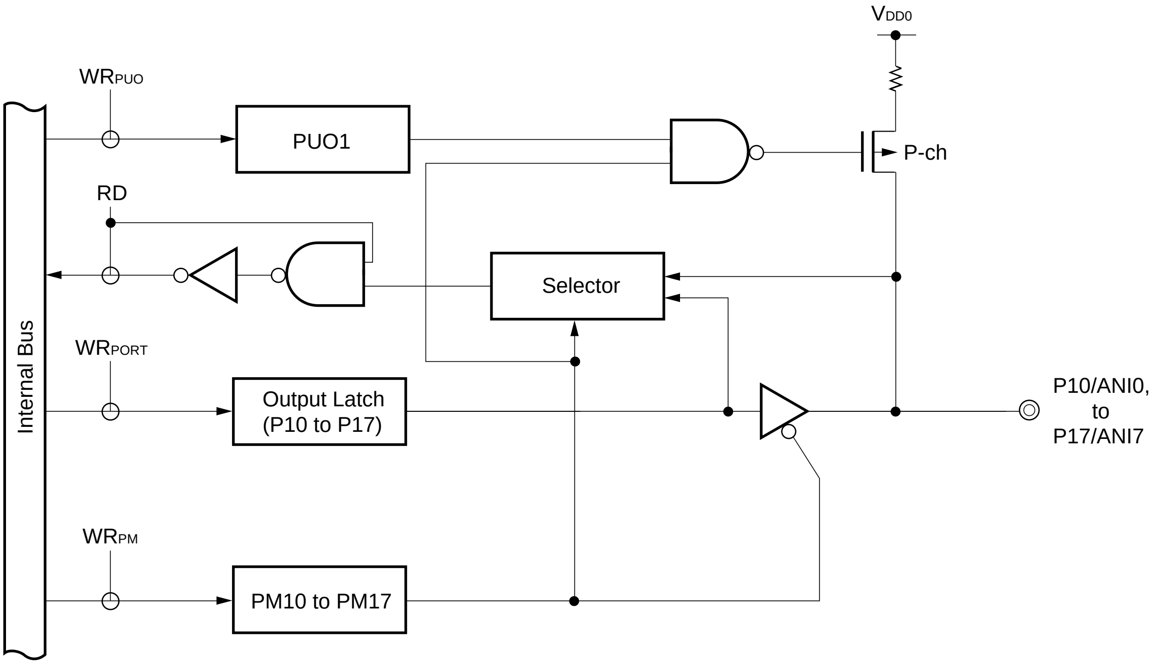
Alternate functions include an A/D converter analog input.

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

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

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

Figure 6-4. P10 to P17 Block Diagram



PUO : Pull-up resistor option register

PM : Port mode register

RD : Port 1 read signal

WR : Port 1 write signal

6.2.3 Port 2 (μPD780058 Subseries)

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

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

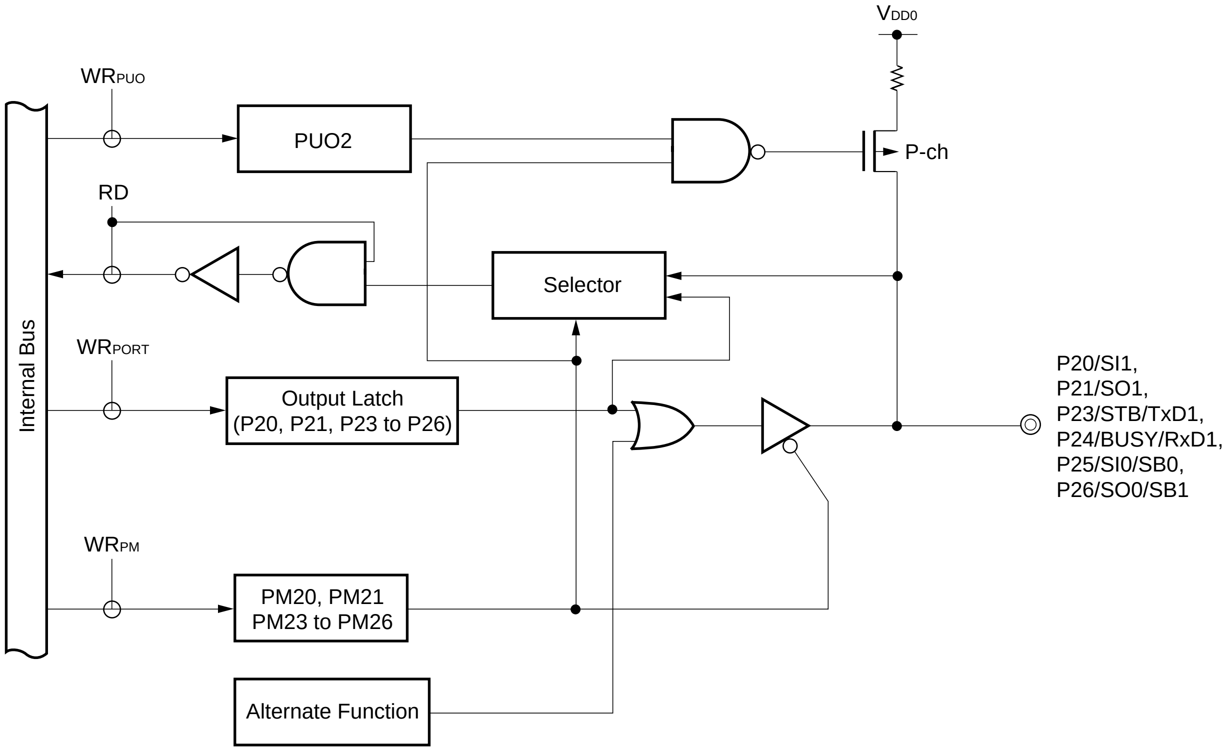
$\overline{\text{RESET}}$ input sets port 2 to input mode.

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

Cautions 1. When used as a serial interface pin, set the input/output and output latch according to its functions. For the setting method, refer to Figure 16-4 Serial Operating Mode Register 0 Format, Figure 18-3 Serial Operating Mode Register 1 Format, and Table 19-2 Serial Interface Channel 2 Operating Mode Settings.

2. When reading the pin state in SBI mode, set the PM2n bit of PM2 to 1 (n = 5, 6) (Refer to the description of (10) Judging busy state of slave in section 16.4.3 SBI mode operation).

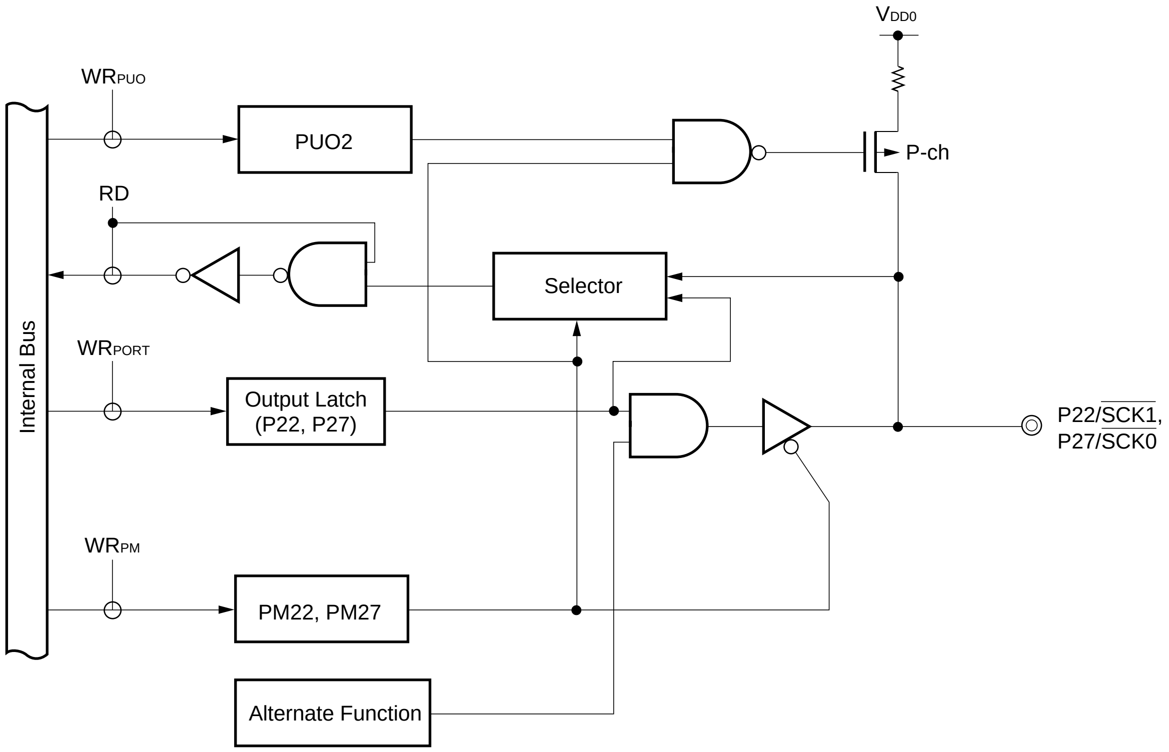
★ **Figure 6-5. P20, P21, and P23 to P26 Block Diagram**



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

★

Figure 6-6. P22 and P27 Block Diagram



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

6.2.4 Port 2 (μPD780058Y Subseries)

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

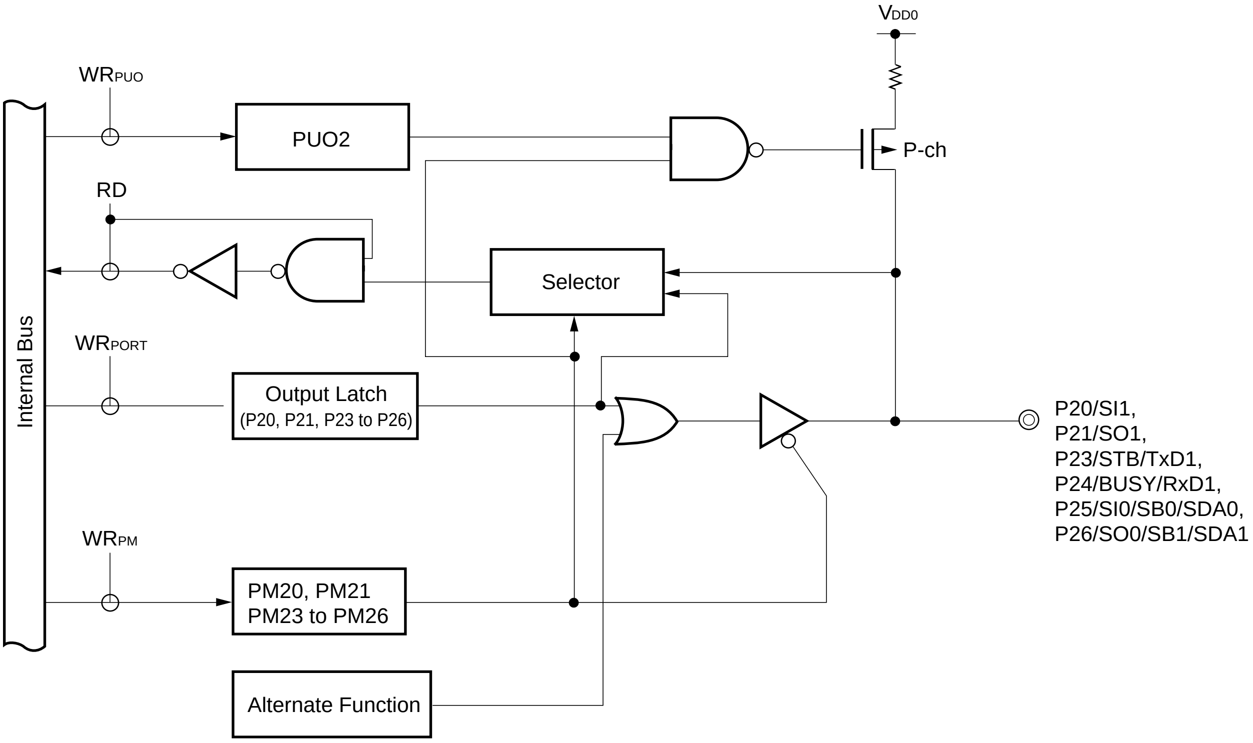
Alternate functions include serial interface data input/output, clock input/output, automatic transmit/receive busy input, and strobe output.

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

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

Caution When used as a serial interface pin, set the input/output and output latch according to its functions. For the setting method, refer to Figure 17-4 Serial Operating Mode Register 0 Format, Figure 18-3 Serial Operating Mode Register 1 Format, and Table 19-2 Serial Interface Channel 2 Operating Mode Settings.

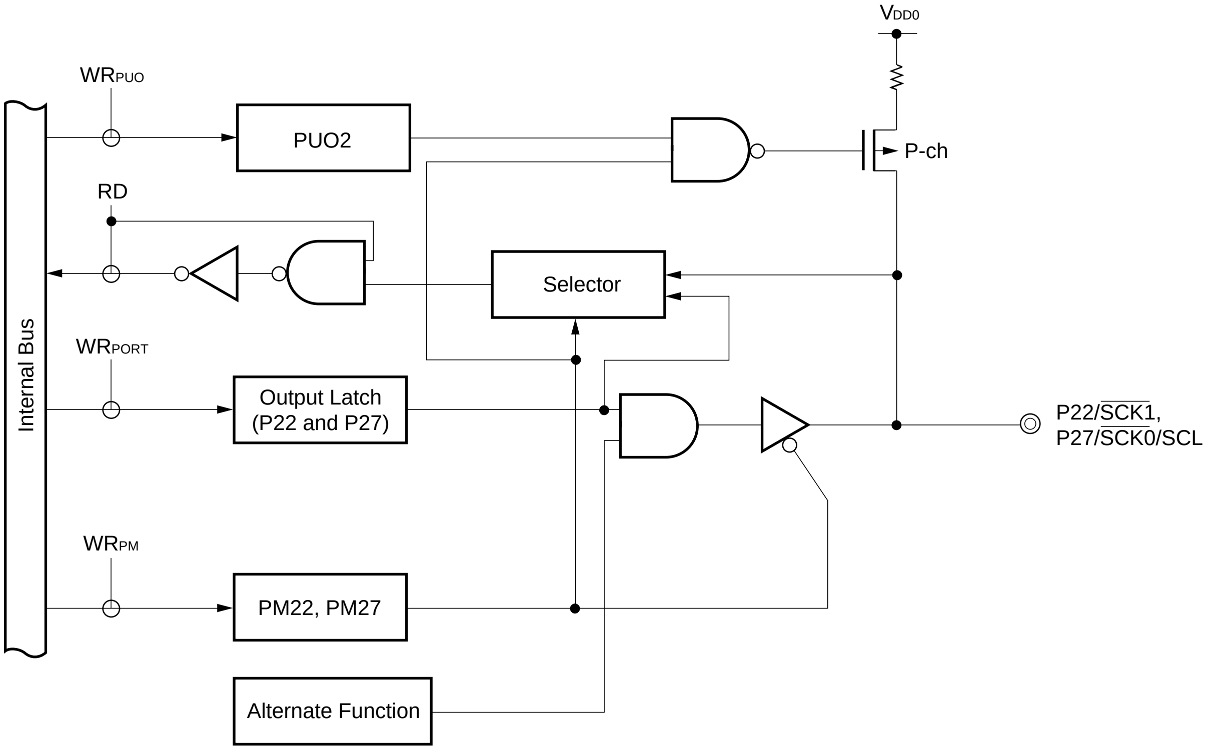
★ **Figure 6-7. P20, P21, and P23 to P26 Block Diagram**



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

★

Figure 6-8. P22 and P27 Block Diagram



PUO : Pull-up resistor option register

PM   : Port mode register

RD   : Port 2 read signal

WR   : Port 2 write signal

6.2.5 Port 3

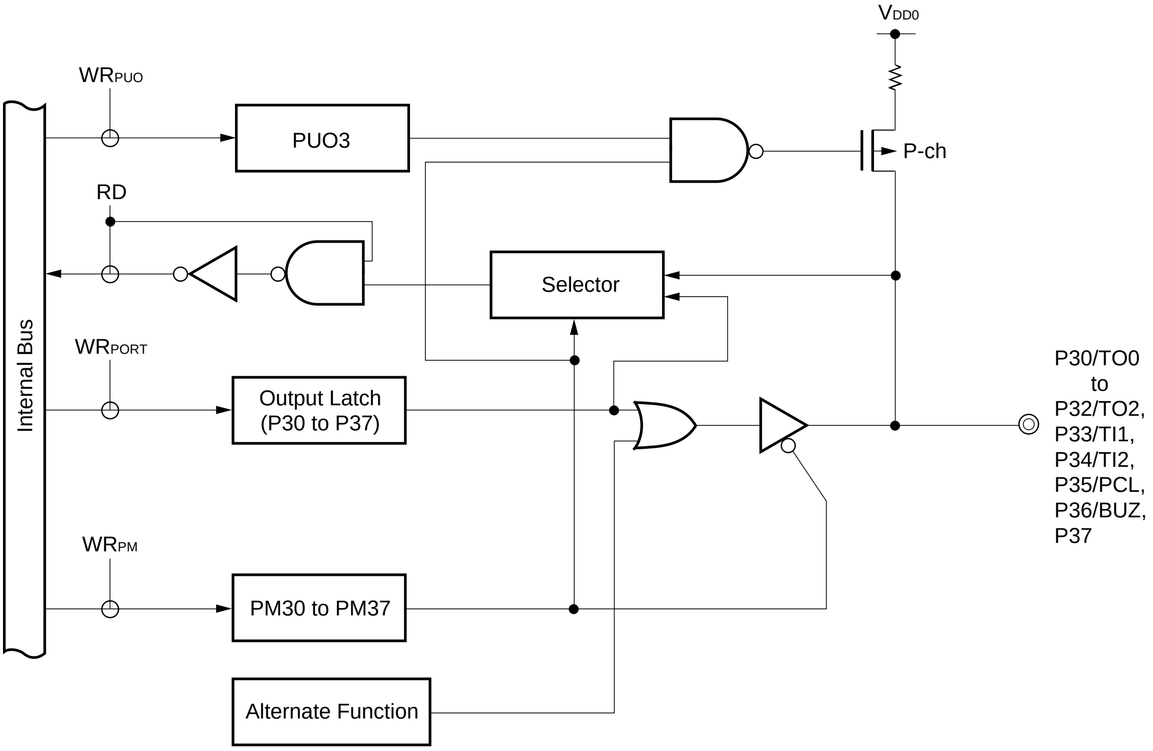
Port 3 is an 8-bit input/output port with output latch. P30 to P37 pins can specify the input mode/output mode in 1-bit units with the port mode register 3 (PM3). When P30 to P37 pins are used as an input port, an on-chip pull-up resistor can be used to them in 8-bit units with a pull-up resistor option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

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

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

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

★ **Figure 6-9. P30 to P37 Block Diagram**



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

6.2.6 Port 4

Port 4 is an 8-bit input/output port with output latch. P40 to P47 pins can specify the input mode/output mode in 8-bit units with the memory expansion mode register (MM). When P40 to P47 pins are used as input ports, an on-chip pull-up resistor can be used to them in 8-bit units with pull-up resistor option register L (PUOL).

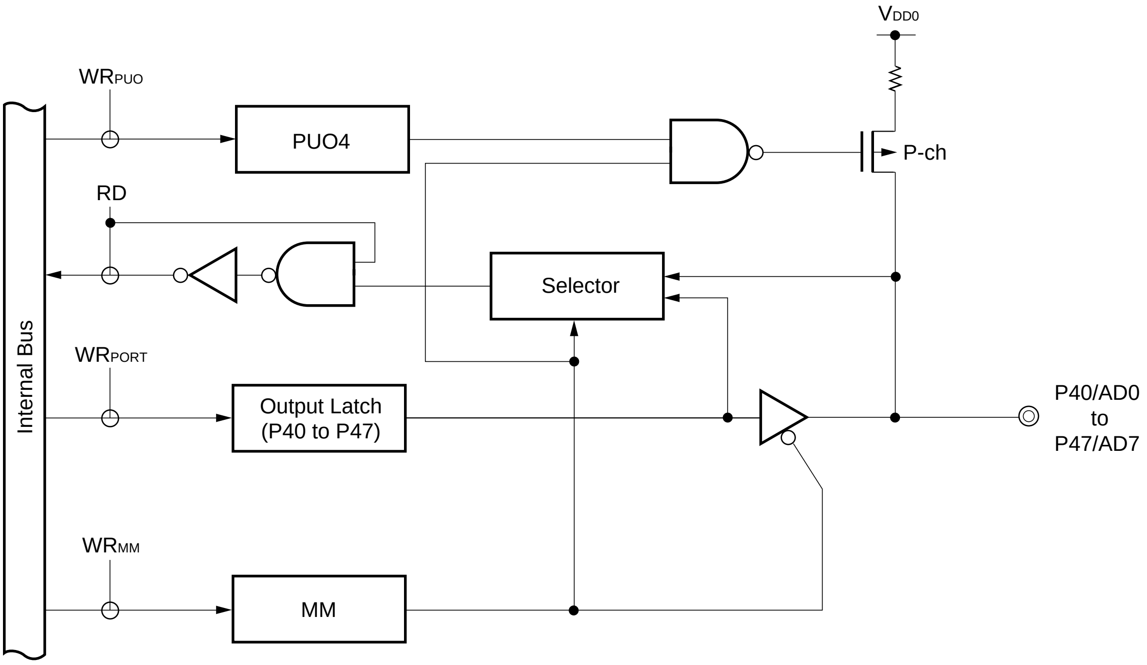
The test input flag (KRIF) can be set to 1 by detecting falling edges.

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

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

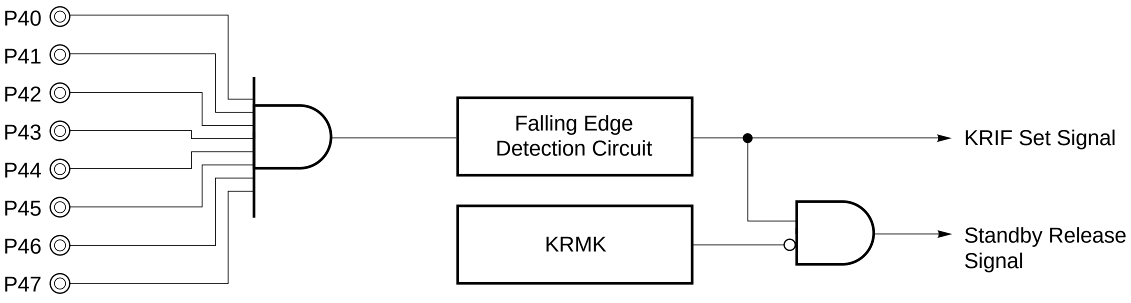
Figures 6-10 and 6-11 show a block diagram of port 4 and block diagram of falling edge detection circuit, respectively.

Figure 6-10. P40 to P47 Block Diagram



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

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



6.2.7 Port 5

Port 5 is an 8-bit input/output port with output latch. P50 to P57 pins can specify the input mode/output mode in 1-bit units with the port mode register 5 (PM5). When P50 to P57 pins are used as an input port, an on-chip pull-up resistor can be used to them in 8-bit units with a pull-up resistor option register L (PUOL).

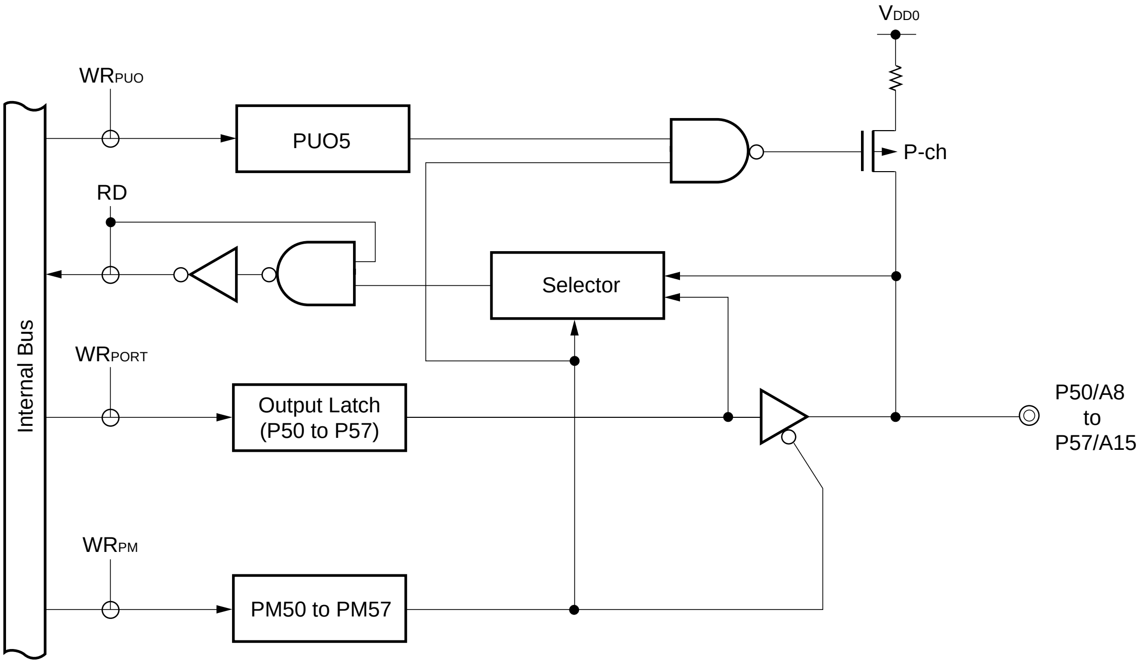
Port 5 can drive LEDs directly.

Alternate functions include address bus function in external memory expansion mode.

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

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

Figure 6-12. P50 to P57 Block Diagram



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

6.2.8 Port 6

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

This port has functions related to pull-up resistors as shown below. These functions depending on whether the higher 4 bits or lower 4 bits of a port are used, and whether the mask ROM model or flash memory model is used.

Table 6-4. Pull-up Resistor of Port 6

	Higher 4 Bits (P64 to P67 pins)	Lower 4 bits (P60 to P63 pins)
Mask ROM version	On-chip pull-up resistor can be connected in 4-bit units by PUO6	Pull-up resistor can be connected in 1-bit units by mask option
Flash memory version		Pull-up resistor is not connected

PUO6: Bit 6 of pull-up resistor option register L (PUOL)

Pins P60 to P63 can drive LEDs directly.

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

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

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

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

2. The value of the low-level input leakage current flowing to the P60 to P63 pins differ depending on the following conditions:

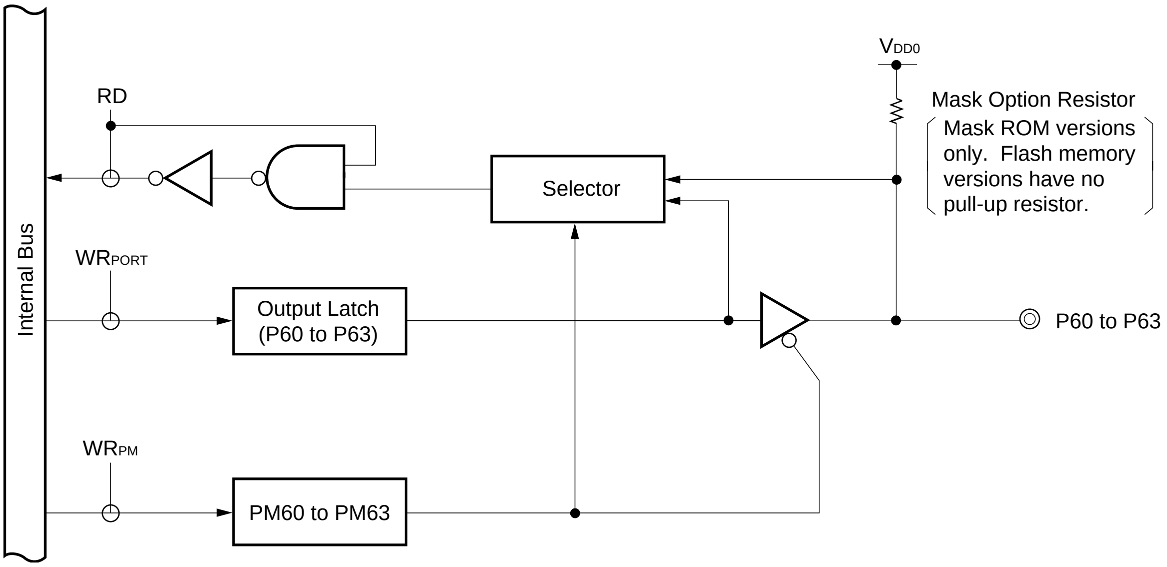
[Mask ROM version]

- **When pull-up resistor is connected: always –3 μA (MAX.)**
- **When pull-up resistor is not connected**
  - **For duration of 1.5 clock (no wait) when instruction to read port 6 (P6) and port mode register 6 (PM6) is executed: –200 μA (MAX.)**
  - **Other than above: –3 μA (MAX.)**

[Flash memory version]

- **For duration of 1.5 clock (no wait) when instruction to read port 6 (P6) and port mode register 6 (PM6) is executed: –200 μA (MAX.)**
- **Other than above: –3 μA (MAX.)**

Figure 6-13. P60 to P63 Block Diagram

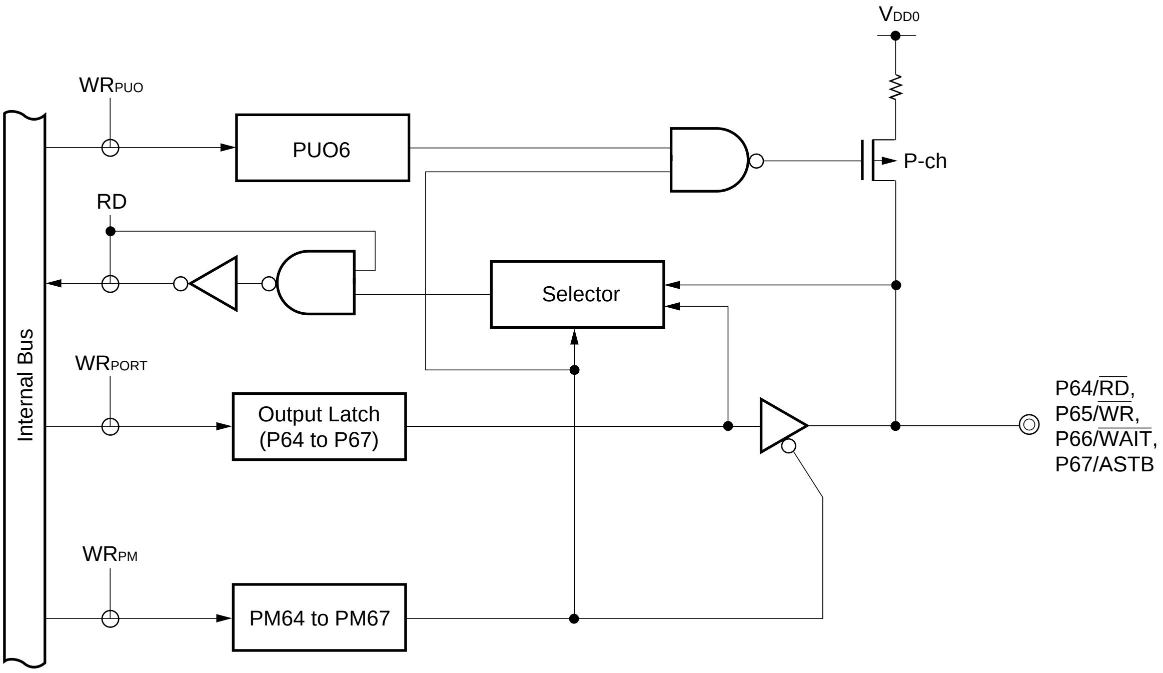

Internal Bus
RD
WRPORT
WRPM
Selector
Output Latch (P60 to P63)
PM60 to PM63
VDD0
Mask Option Resistor
Mask ROM versions only. Flash memory versions have no pull-up resistor.
P60 to P63

PM : Port mode register

RD : Port 6 read signal

WR : Port 6 write signal

Figure 6-14. P64 to P67 Block Diagram



PUO : Pull-up resistor option register

PM : Port mode register

RD : Port 6 read signal

WR : Port 6 write signal

6.2.9 Port 7

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

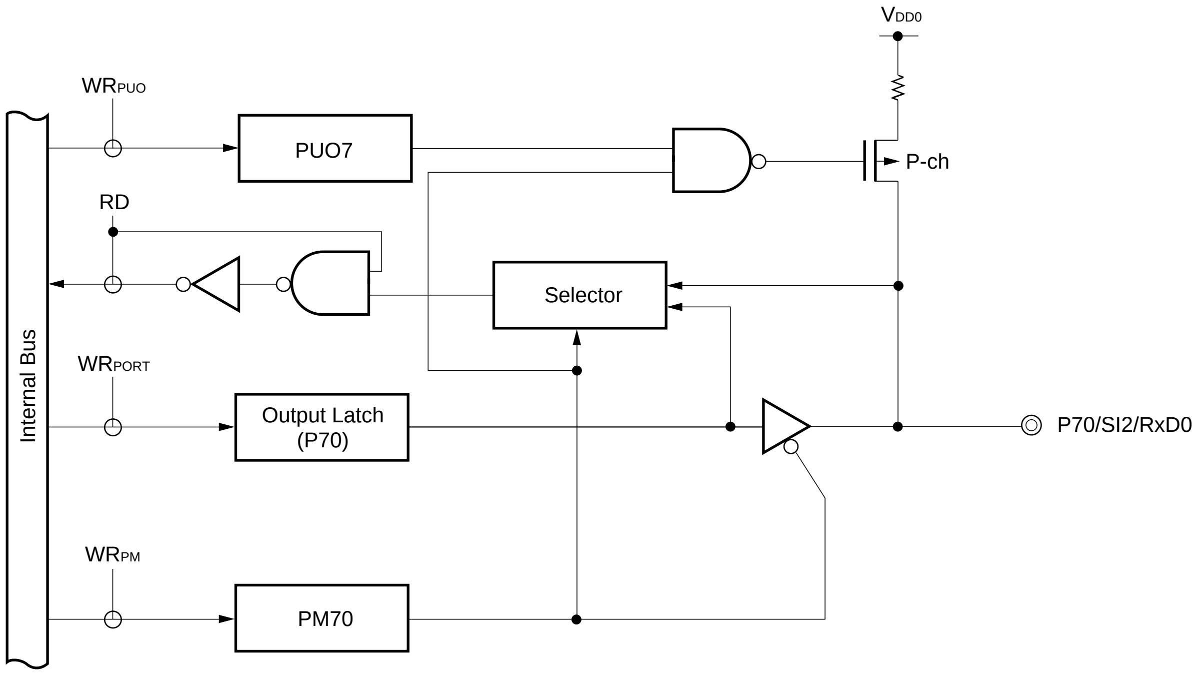
Alternate functions include serial interface channel 2 data input/output and clock input/output.

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

Figures 6-15 and 6-16 show a block diagram of port 7.

Caution When used as a serial interface pin, set the input/output and output latch according to its functions. For the setting method, refer to Table 19-2 Serial Interface Channel 2 Operating Mode Setting.

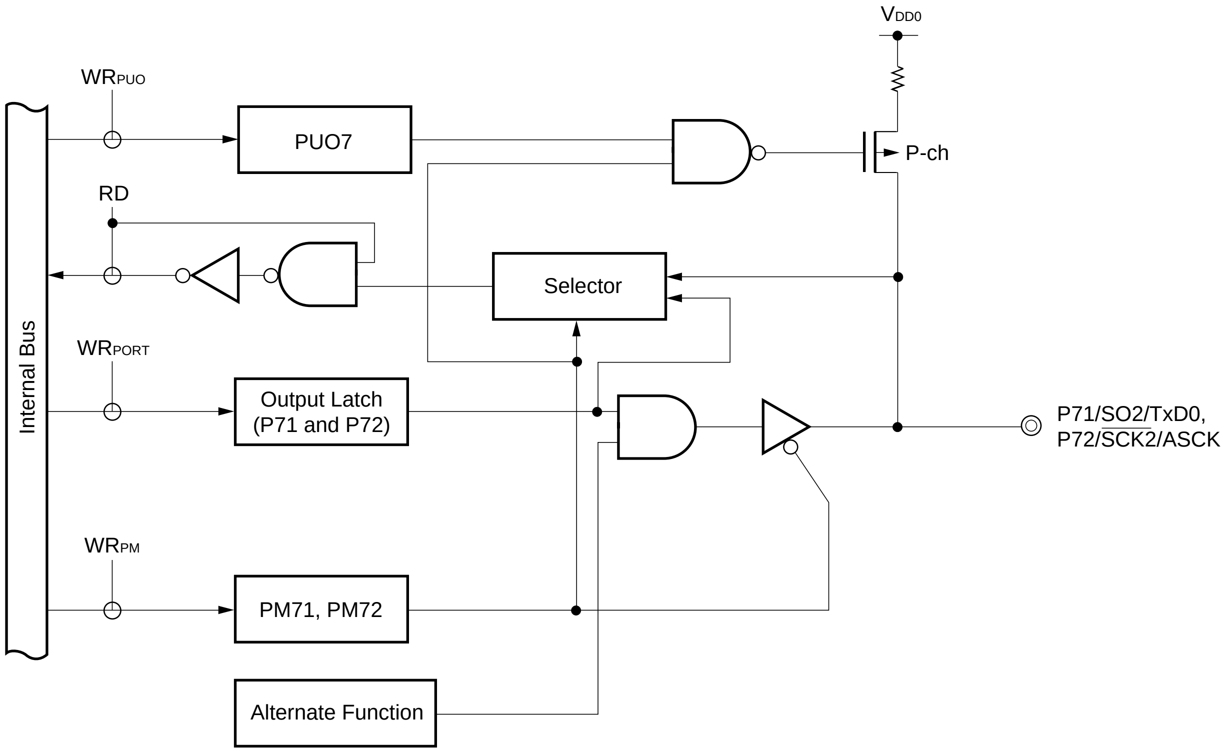
Figure 6-15. P70 Block Diagram



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

★

Figure 6-16. P71 and P72 Block Diagram



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

6.2.10 Port 12

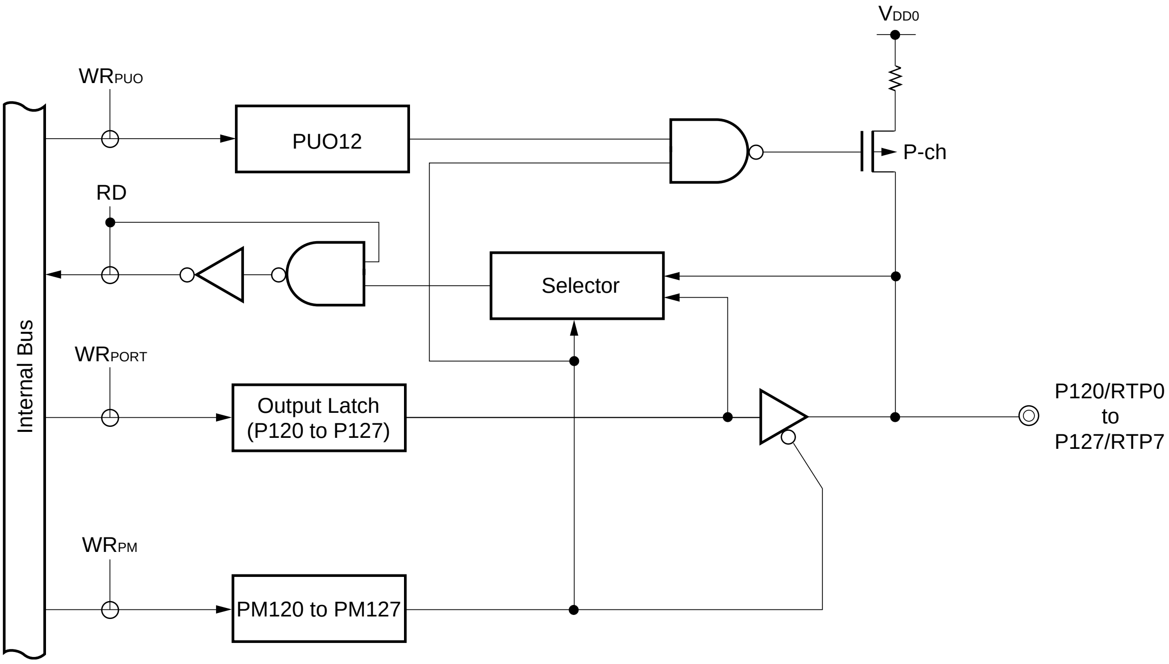
This is an 8-bit input/output port with output latches. Input mode/output mode can be specified bit-wise by means of port mode register 12 (PM12). When pins P120 to P127 are used as an input port pin, an on-chip pull-up resistor can be used as an 8-bit unit by means of pull-up resistor option register H (PUOH).

These pins have an alternate function, serving as real-time outputs.

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

The port 12 block diagram is shown in Figure 6-17.

Figure 6-17. P120 to P127 Block Diagram



PUO : Pull-up resistor option register

PM : Port mode register

RD : Port 12 read signal

WR : Port 12 write signal

6.2.11 Port 13

This is a 2-bit input/output port with output latches. Input mode/output mode can be specified bit-wise by means of port mode register 13 (PM13). When pins P130 and P131 are used as an input port pin, an on-chip pull-up resistor can be used as a 2-bit unit by means of pull-up resistor option register H (PUOH).

These pins have an alternate function, serving as D/A converter analog outputs.

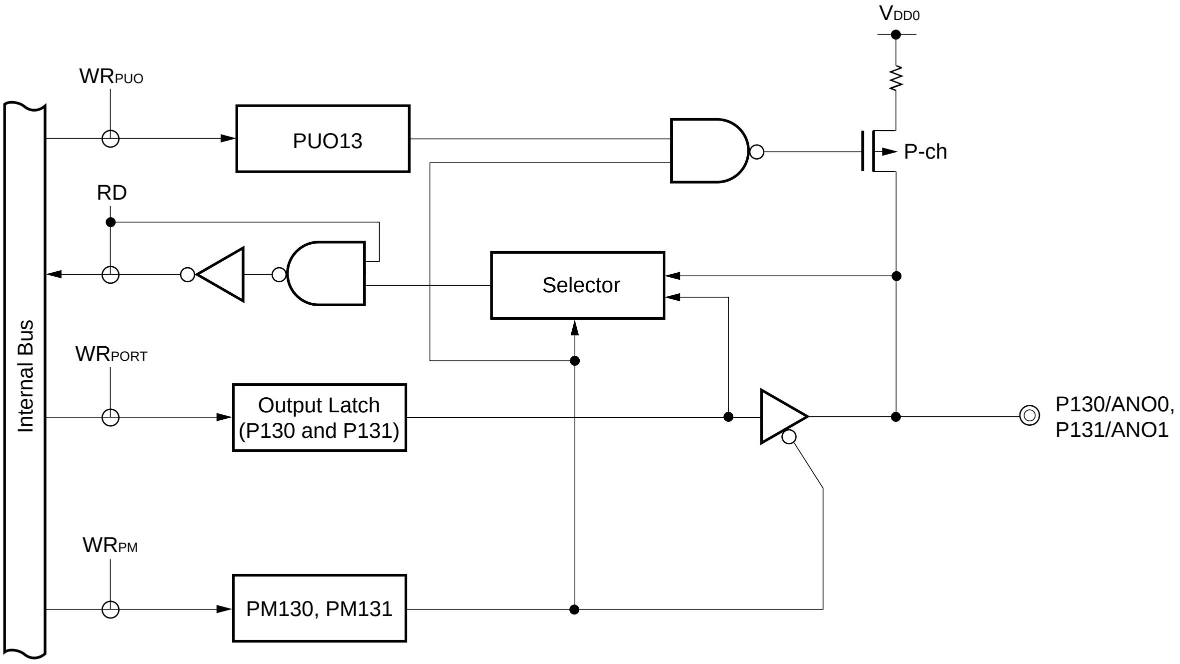
$\overline{\text{RESET}}$ input sets to input mode.

Figure 6-18 shows a block diagram of port 13.

Caution When only either one of the D/A converter channels is used with $AV_{REF1} < V_{DD0}$, the other pins that are not used as analog outputs must be set as follows:

- **Set PM13. bit of the port mode register 13 (PM13) to 1 (input mode) and connect the pin to V_{SS0}.**
- **Set PM13x bit of the port mode register 13 (PM13) to 0 (output mode) and the output latch to 0, to output low level from the pin.**

Figure 6-18. P130 and P131 Block Diagram



PUO : Pull-up resistor option register
PM : Port mode register
RD : Port 13 read signal
WR : Port 13 write signal

6.3 Port Function Control Registers

The following four types of registers control the ports.

- Port mode registers (PM0 to PM3, PM5 to PM7, PM12, PM13)
- Pull-up resistor option register (PUOH, PUOL)
- Memory expansion mode register (MM)
- Key return mode register (KRM)

(1) Port mode registers (PM0 to PM3, PM5 to PM7, PM12, PM13)

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

PM0 to PM3, PM5 to PM7, PM12, and PM13 are independently set with a 1-bit or 8-bit memory manipulation instruction

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

When port pins are used as the alternate function pins, set the port mode register and output latch according to Table 6-5.

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

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

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

Table 6-5. Port Mode Register and Output Latch Settings When Using Alternate Functions

Pin Name	Alternate functions		PM××	P××
	Name	Input/Output		
P00	INTP0	Input	1 (Fixed)	None
	TI00	Input	1 (Fixed)	None
P01	INTP1	Input	1	×
	TI01	Input	1	×
P02 to P05	INTP2 to INTP5	Input	1	×
P07 **Note 1**	XT1	Input	1 (Fixed)	None
P10 to P17 **Note 1**	ANI0 to ANI7	Input	1	×
P30 to P32	TO0 to TO2	Output	0	0
P33, P34	TI1, TI2	Input	1	×
P35	PCL	Output	0	0
P36	BUZ	Output	0	0
P40 to P47	AD0 to AD7	Input/output	× **Note 2**	
P50 to P57	A8 to A15	Output	× **Note 2**	
P64	$\overline{\text{RD}}$	Output	× **Note 2**	
P65	$\overline{\text{WR}}$	Output	× **Note 2**	
P66	$\overline{\text{WAIT}}$	Input	× **Note 2**	
P67	ASTB	Output	× **Note 2**	
P120 to P127	RTP0 to RTP7	Output	0	Desired value
P130, P131 **Note 1**	ANO0, ANO1	Output	1	×

Notes 1. If these ports are read out when these pins are used in the alternate function mode, undefined values are read.

2. When the P40 to P47 pins P50 to P57 pins, and P64 to P67 pins are used for alternate functions, set the function by the memory extension mode register (MM).

Cautions 1. When not using external wait in the external memory extension mode, the P66 pin can be used as an I/O port.

2. When port 2 and port 7 are used for serial interface, the I/O latch or output latch must be set according to its function. For the setting methods, see Figure 16-4 Serial Operation Mode Register 0 Format, Figure 17-4 Serial Operation Mode Register 0 Format, Figure 18-3 Serial Operation Mode Register 1 Format, and Table 19-2 Serial Interface Channel 2 Operating Mode Settings.

Remarks

×	:	don't care
PM××	:	port mode register
P××	:	port output latch

Figure 6-19. Port Mode Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
PM0	1	1	PM05	PM04	PM03	PM02	PM01	1	FF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FF25H	FFH	R/W
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60	FF26H	FFH	R/W
PM7	1	1	1	1	1	PM72	PM71	PM70	FF27H	FFH	R/W
PM12	PM127	PM126	PM125	PM124	PM123	PM122	PM121	PM120	FF2CH	FFH	R/W
PM13	1	1	1	1	1	1	PM131	PM130	FF2DH	FFH	R/W

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

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

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

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

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

Cautions 1. P00 and P07 pins do not incorporate a pull-up resistor.

2. When ports 1, 4, 5, and P64 to P67 pins are used as alternate function pins, an on-chip pull-up resistor cannot be used even if 1 is set in PUOm bit of PUOH, PUOL (m = 1, 4 to 6).

3. Pins P60 to P63 can be connected with pull-up resistor by mask option only for mask ROM version.

Figure 6-20. Pull-up Resistor Option Register Format

Symbol	7	6	⑤	④	3	2	1	0	Address	After Reset	R/W
PUOH	0	0	PUO13	PUO12	0	0	0	0	FFF3H	00H	R/W
	⑦	⑥	⑤	④	③	②	①	⓪			
PUOL	PUO7	PUO6	PUO5	PUO4	PUO3	PUO2	PUO1	PUO0	FFF7H	00H	R/W

PUOm	Pm Internal Pull-up Resistor Selection (m = 0 to 7, 12, 13)
0	Internal pull-up resistor not used
1	Internal pull-up resistor used

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

(3) Memory expansion mode register (MM)

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

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

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

Figure 6-21. Memory Expansion Mode Register Format

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

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

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

Note The full address mode allows external expansion for all areas of the 64-Kbyte address space, except the internal ROM, RAM, SFR, and use-prohibited areas.

Remarks 1. P60 to P63 pins enter the port mode in both the single-chip and memory expansion mode.

2. Besides setting port 4 input/output, MM also sets the wait count and external expansion area.

(4) Key return mode register (KRM)

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

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

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

Figure 6-22. Key Return Mode Register Format

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

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

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

Caution **When falling edge detection of port 4 is used, KRIF should be cleared to 0 (not cleared to 0 automatically).**

6.4 Port Function Operations

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

6.4.1 Writing to input/output port

(1) Output mode

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

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

(2) Input mode

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

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

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

6.4.2 Reading from input/output port

(1) Output mode

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

(2) Input mode

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

6.4.3 Operations on input/output port

(1) Output mode

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

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

(2) Input mode

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

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

6.5 Selection of Mask Option

The following mask option is provided in mask ROM version. The flash memory versions have no mask options.

Table 6-6. Comparison Between Mask ROM Version and Flash Memory Version

Pin Name	Mask ROM Version	Flash Memory Version
Mask option for pins P60 to P63	Bit-wise-selectable on-chip pull-up resistors	No on-chip pull-up resistor

[MEMO]

CHAPTER 7 CLOCK GENERATOR

7.1 Clock Generator Functions

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

(1) Main system clock oscillator

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

(2) Subsystem clock oscillator

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

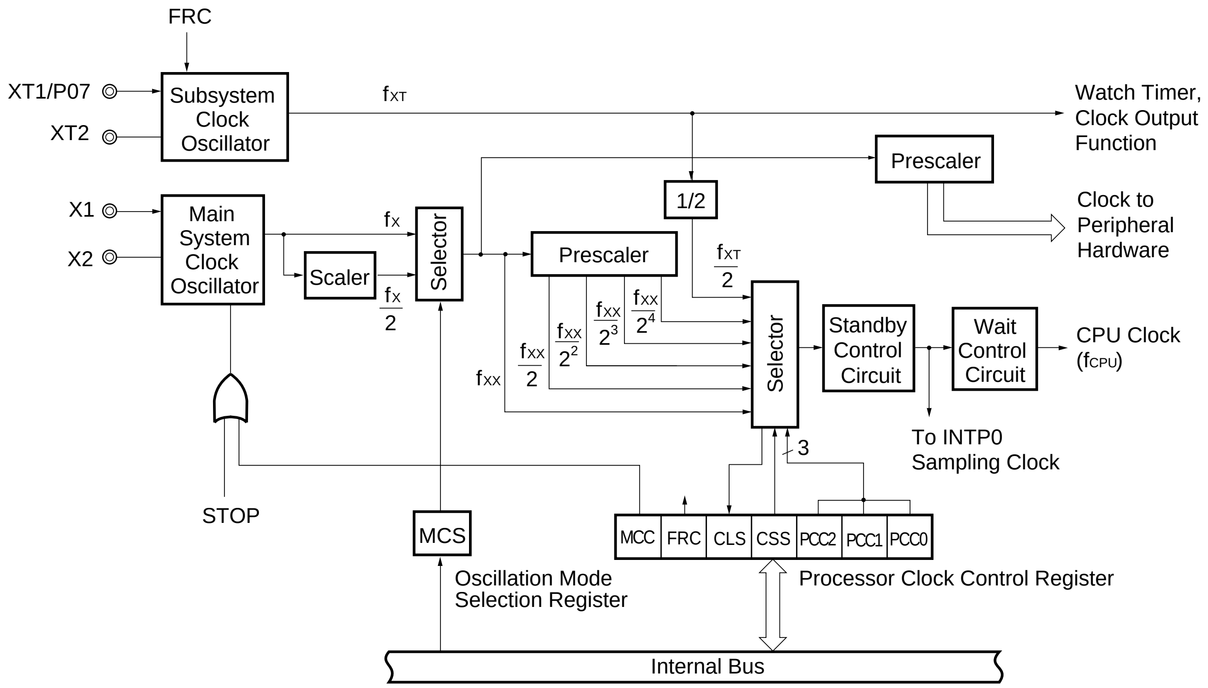
7.2 Clock Generator Configuration

The clock generator consists of the following hardware.

Table 7-1. Clock Generator Configuration

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

Figure 7-1. Clock Generator Block Diagram



7.3 Clock Generator Control Register

The clock generator is controlled by the following two registers:

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

(1) Processor clock control register (PCC)

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.

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

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

Figure 7-2. Subsystem Clock Feedback Resistor

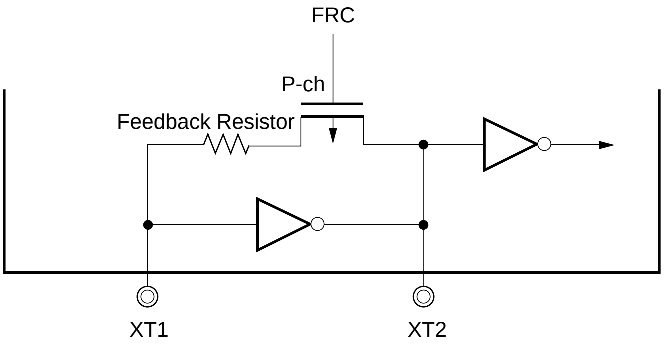


Figure 7-3. Processor Clock Control Register Format

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

R/W

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

R

CLS	CPU Clock Status
0	Main system clock
1	Subsystem clock

R/W

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

R/W

MCC	Main System Clock Oscillation Control Note 2
0	Oscillation possible
1	Oscillation stopped

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

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

Caution Bit 3 must be set to 0.

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

2. f_X : Main system clock oscillation frequency

3. f_{XT} : Subsystem clock oscillation frequency

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

The fastest instruction of the μPD780058, 780058Y Subseries is executed in 2 CPU clocks. Therefore, the relation between the CPU clock (f_{CPU}) and minimum instruction execution time is as shown in Table 7-2.

★ **Table 7-2. Relationships Between CPU Clock and Minimum Instruction Execution Time**

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

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

(2) Oscillation mode selection register (OSMS)

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

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

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

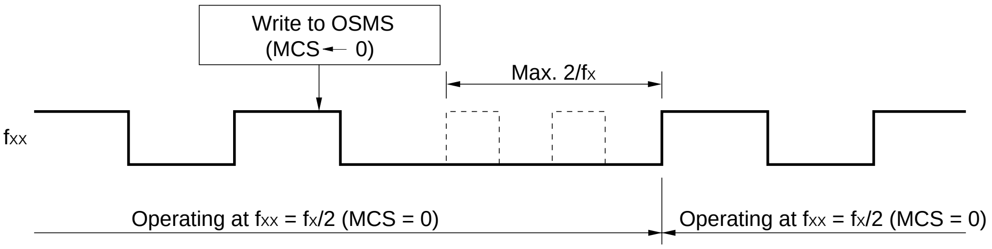
Figure 7-4. Oscillation Mode Selection Register Format

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

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

Cautions 1. Writing to OSMS should be performed only immediately after reset signal release and before peripheral hardware operation starts. As shown in Figure 7-5 below, writing data (including same data as previous) to OSMS cause delay of main system clock cycle up to 2/f_X during the write operation. Therefore, if this register is written during the operation, in peripheral hardware which operates with the main system clock, a temporary error occurs in the count clock cycle of timer, etc. In addition, because the oscillation mode is changed by this register, the clocks for peripheral hardware as well as that for the CPU are switched.

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


Write to OSMS
(MCS ← 0)
Max. 2/fX
fXX
Operating at fXX = fX/2 (MCS = 0)
Operating at fXX = fX/2 (MCS = 0)

2. When writing 1 to MCS, V_{DD} must be 2.7 V or higher before the write execution.

Remarks f_{XX} : Main system clock frequency (f_X or f_X/2)

f_X : Main system clock oscillation frequency

7.4 System Clock Oscillator

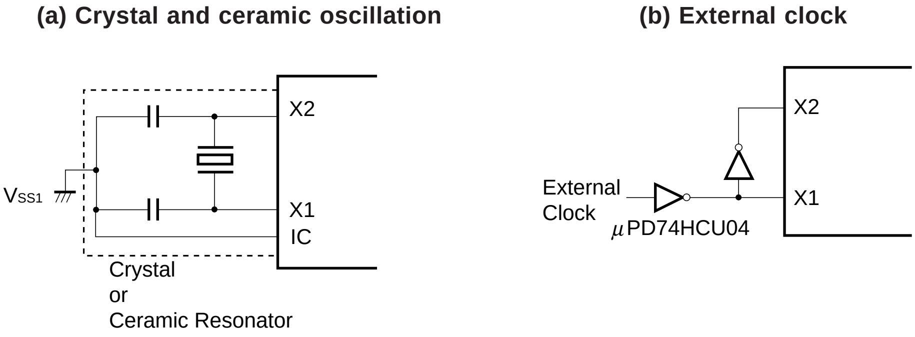
7.4.1 Main system clock oscillator

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

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

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

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



(a) Crystal and ceramic oscillation

(b) External clock

Caution Do not execute the STOP instruction or do not set MCC (bit 7 of processor clock control register (PCC)) to 1 if an external clock is used. Otherwise, the operation of the main system clock will be stopped and the X2 pin will be pulled up to V_{DD1}.

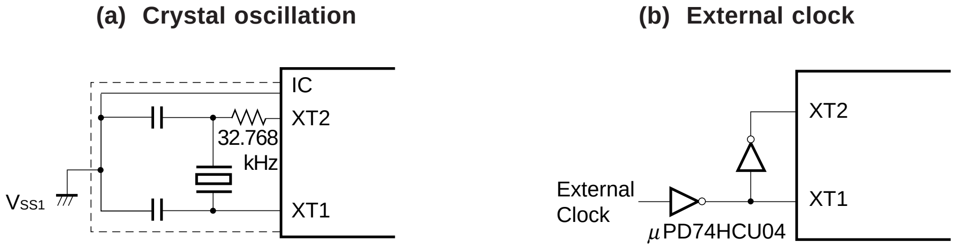
7.4.2 Subsystem clock oscillator

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

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

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

Figure 7-7. External Circuit of Subsystem Clock Oscillator



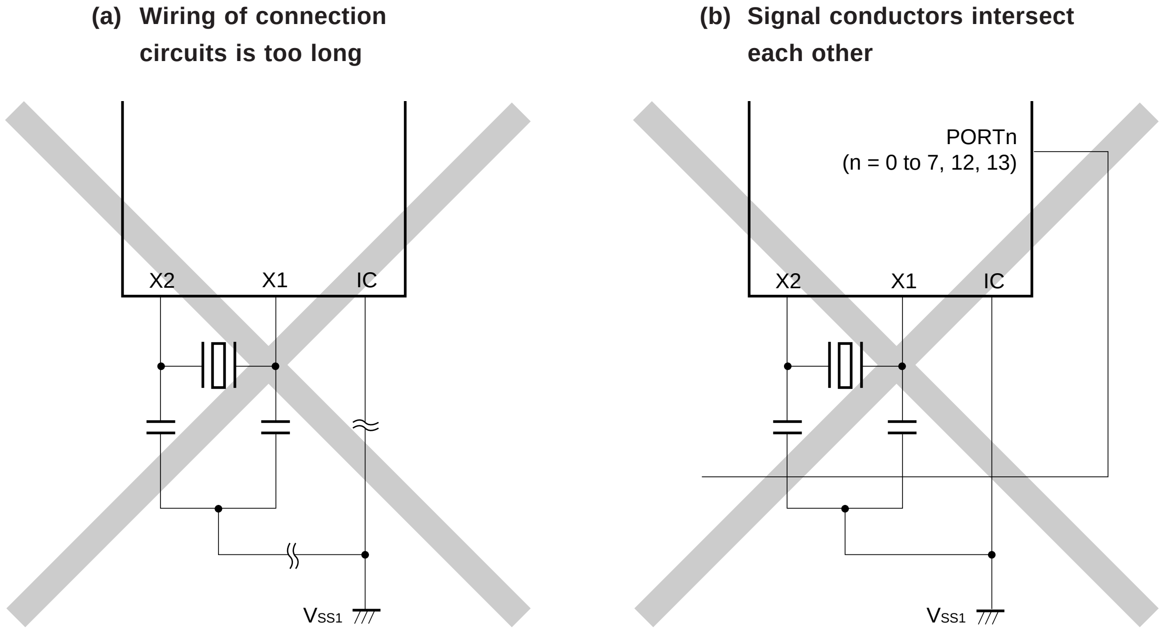
Cautions 1. When using a main system clock oscillator and a subsystem clock oscillator, carry out wiring in the broken line area in Figures 7-6 and 7-7 to prevent any effects from wiring capacities.

- **Minimize the wiring length.**
- **Do not allow wiring to intersect with other signal conductors. Do not allow wiring to come near changing high current.**
- **Set the potential of the grounding position of the oscillator capacitor to that of V_{SS1}. Do not ground to any ground pattern where high current is present.**
- **Do not fetch signals from the oscillator.**

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

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

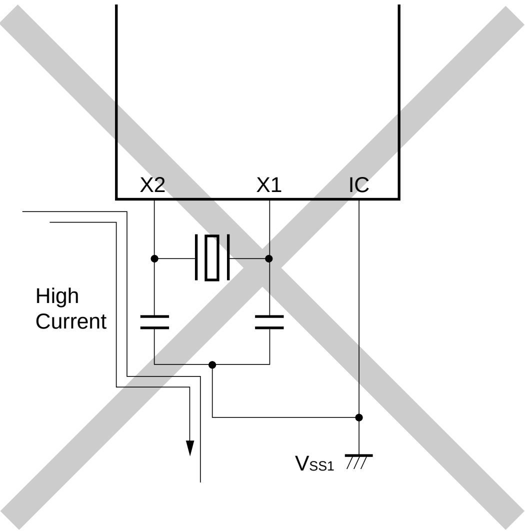
Figure 7-8. Examples of Oscillator with Bad Connection (1/2)


(a) Wiring of connection circuits is too long
X2
X1
IC
V_{SS1}
(b) Signal conductors intersect each other
PORTn
(n = 0 to 7, 12, 13)
X2
X1
IC
V_{SS1}

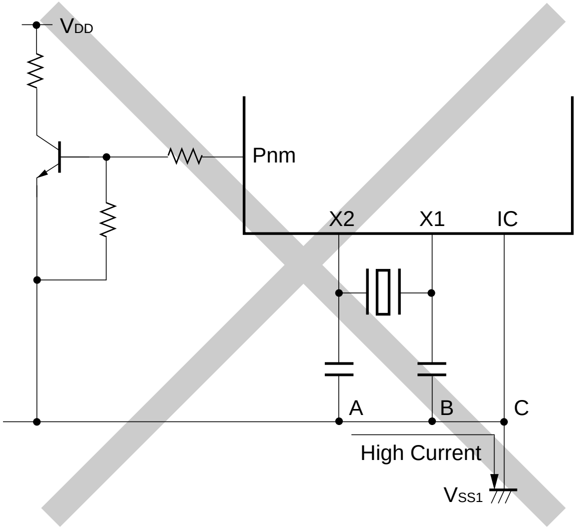
Remark When using a subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Further, insert resistors in series on the side of XT2.

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

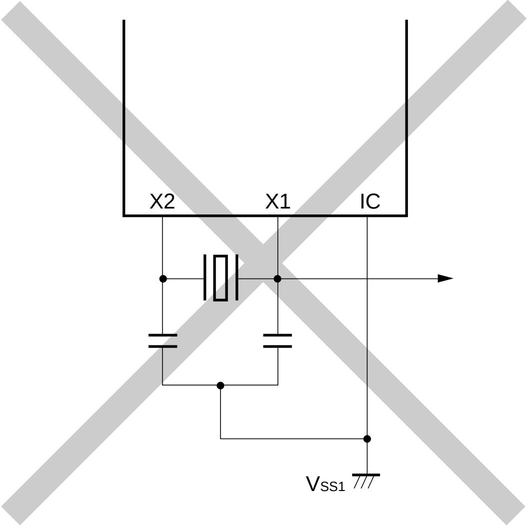
(c) Changing high current is too near a signal conductor



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



(e) Signals are fetched



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. If XT2 and XT1 are wired in parallel, the cross-talk noise of X1 may increase with XT2, resulting in malfunctioning. To prevent that from occurring, it is recommended to wire XT2 and X1 so that they are not in parallel, and to correct the IC pin between XT2 and X1 directly to V_{SS1}.

7.4.3 Divider

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

7.4.4 When not using subsystem clocks

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 : Leave open

In this state, however, some current may leak via the internal feedback resistor of the subsystem clock oscillator when the main system clock stops. To suppress the leakage current, disconnect the above internal feedback resistor by using the bit 6 (FRC) of the processor clock control register (PCC). In this case also, connect the XT1 and XT2 pins as described above.

7.5 Clock Generator Operations

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

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

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

(a) Upon generation of $\overline{\text{RESET}}$ signal, the lowest speed mode of the main system clock (12.8 μs when operated at 5.0 MHz) is selected (PCC = 04H, OSMS = 00H). Main system clock oscillation stops while low level is applied to $\overline{\text{RESET}}$ pin.

(b) With the main system clock selected, one of the six CPU clock types (0.4μs, 0.8μs, 1.6μs, 3.2μs, 6.4μs, 12.8μs @ 5.0 MHz) can be selected by setting the PCC and OSMS.

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

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

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

(f) The main system clock is divided and supplied to the peripheral hardware. The subsystem clock is supplied to 16-bit timer/event counter, the watch timer, and clock output functions only. Thus, 16-bit timer/event counter (when selecting watch timer output for count clock operating with subsystem clock), the watch function, and the clock output function can also be continued in the standby state. However, since all other peripheral hardware operate with the main system clock, the peripheral hardware also stops if the main system clock is stopped. (Except external input clock operation)

7.5.1 Main system clock operations

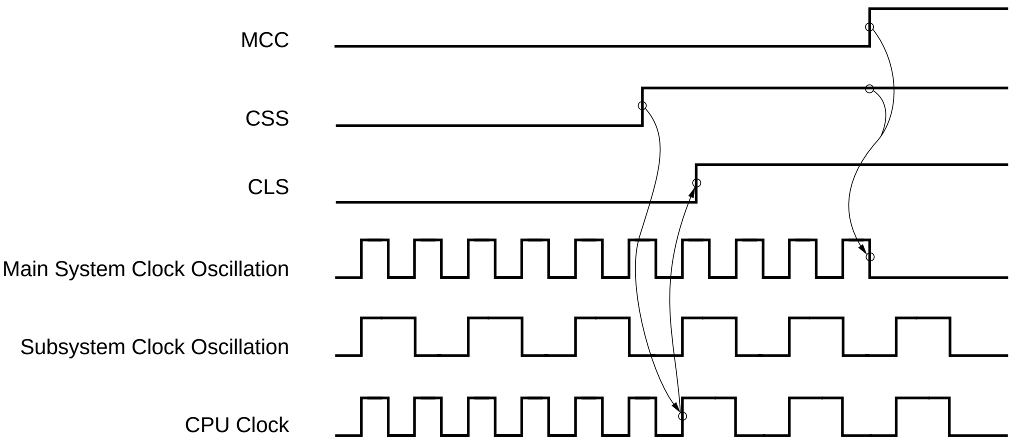
When operated with the main system clock (with bit 5 (CLS) of the processor clock control register (PCC) set to 0), the following operations are carried out by PCC setting.

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

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

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

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



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

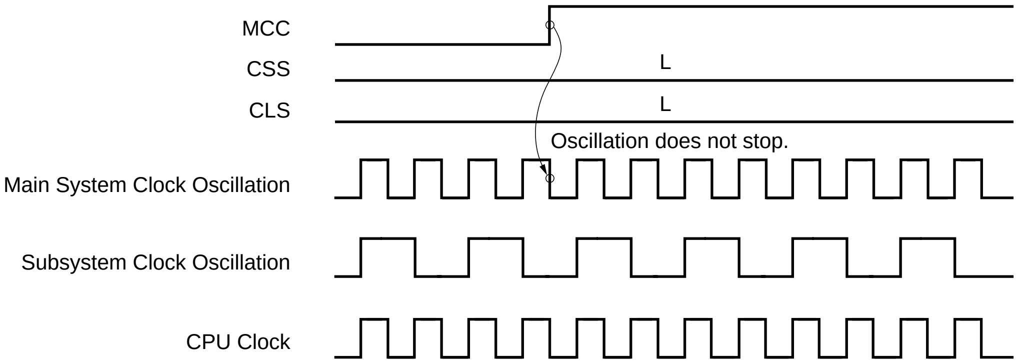
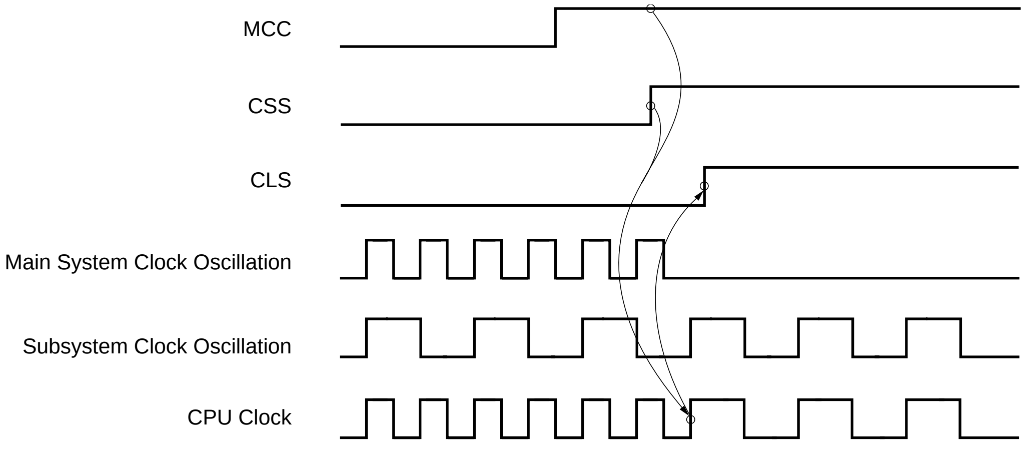

MCC
CSS
L
CLS
L
Oscillation does not stop.
Main System Clock Oscillation
Subsystem Clock Oscillation
CPU Clock

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

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



7.5.2 Subsystem clock operations

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

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

(b) Watchdog timer counting stops.

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

7.6 Changing System Clock and CPU Clock Settings

7.6.1 Time required for switchover between system clock and CPU clock

The system clock and CPU clock can be switched over by means of 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 7-3**).

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

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

Set Values before Switchover				Set Values After Switchover						
									MCS = 1	MCS = 0
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	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 × × ×	1 × × ×
0	0	0	0		16 instructions	16 instructions	16 instructions	16 instructions	$f_X/2f_{XT}$ instruction (77 instructions)	$f_X/4f_{XT}$ instruction (39 instructions)
	0	0	1	8 instructions		8 instructions	8 instructions	8 instructions	$f_X/4f_{XT}$ instruction (39 instructions)	$f_X/8f_{XT}$ instruction (20 instructions)
	0	1	0	4 instructions	4 instructions		4 instructions	4 instructions	$f_X/8f_{XT}$ instruction (20 instructions)	$f_X/16f_{XT}$ instruction (10 instructions)
	0	1	1	2 instructions	2 instructions	2 instructions		2 instructions	$f_X/16f_{XT}$ instruction (10 instructions)	$f_X/32f_{XT}$ instruction (5 instructions)
	1	0	0	1 instruction	1 instruction	1 instruction	1 instruction		$f_X/32f_{XT}$ instruction (5 instructions)	$f_X/64f_{XT}$ instruction (3 instructions)
1	×	×	×	1 instruction	1 instruction	1 instruction	1 instruction	1 instruction		

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

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

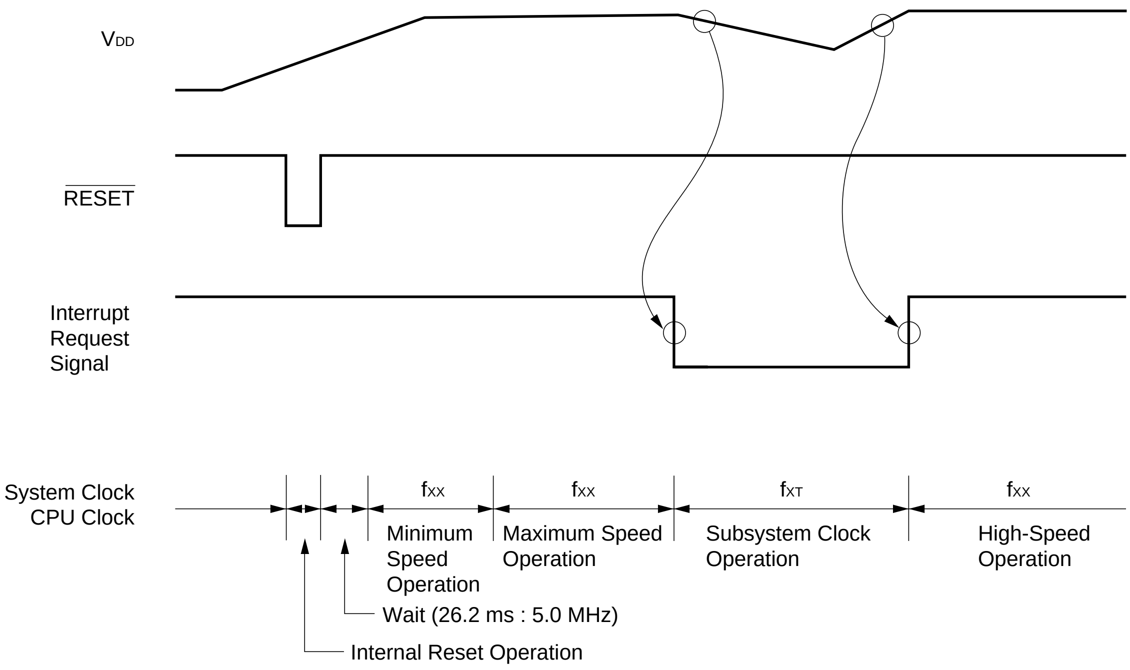
3. Figures in parentheses apply to operation with f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

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

7.6.2 System clock and CPU clock switching procedure

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

Figure 7-10. System Clock and CPU Clock Switching



(1) The CPU is reset by setting the $\overline{\text{RESET}}$ signal to low level after power-on. After that, when reset is released by setting the $\overline{\text{RESET}}$ signal to high level, main system clock starts oscillation. At this time, oscillation stabilization time ($2^{17}/f_X$) is secured automatically.
After that, the CPU starts executing the instruction at the minimum speed of the main system clock (12.8 μs when operated at 5.0 MHz).

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

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

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

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

CHAPTER 8 16-BIT TIMER/EVENT COUNTER

8.1 Outline of Internal Timer of μPD780058 and 780058Y Subseries

This chapter explains the 16-bit timer/event counter. Before that, the timers incorporated into the μPD780058 and 780058Y Subseries and the related functions are outlined below.

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

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

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

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

(3) Watch timer (TM3)

This timer can set a flag every 0.5 sec. and simultaneously generates interrupt requests at the preset time intervals (see **CHAPTER 10 WATCH TIMER**).

(4) Watchdog timer (WDTM)

WDTM can perform the watchdog timer function or generate non-maskable interrupt requests, maskable interrupt requests and $\overline{\text{RESET}}$ at the preset time intervals (see **CHAPTER 11 WATCHDOG TIMER**).

(5) Clock output control circuit

This circuit supplies other devices with the divided main system clock and the subsystem clock (see **CHAPTER 12 CLOCK OUTPUT CONTROL CIRCUIT**).

(6) Buzzer output control circuit

This circuit outputs the buzzer frequency obtained by dividing the main system clock (see **CHAPTER 13 BUZZER OUTPUT CONTROL CIRCUIT**).

Table 8-1. Timer/Event Counter Operation

		16-bit Timer/ Event Counter	8-bit Timer/Event Counters 1 and 2	Watch Timer	Watchdog Timer
Operating Mode	Interval timer	2 channels Note 3	2 channels	1 channel Note 1	1 channel Note 2
	External event counter	√	√	—	—
Function	Timer output	√	√	—	—
	PWM output	√	—	—	—
	Pulse width measurement	√	—	—	—
	Square-wave output	√	√	—	—
	One-shot pulse output	√	—	—	—
	Interrupt request	√	√	√	√
	Test input	—	—	√	—

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

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

3. When capture/compare registers (CR00, CR01) are specified as compare registers.

8.2 16-Bit Timer/Event Counter Functions

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

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

PWM output and pulse width measurement can be used at the same time.

(1) Interval timer

TM0 generates interrupt requests at the preset time interval.

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

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

Remarks 1. f_X: Main system clock oscillation frequency

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

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

(2) PWM output

TM0 can generate 14-bit resolution PWM output.

(3) Pulse width measurement

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

(4) External event counter

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

(5) Square-wave output

TM0 can output a square wave with any selected frequency.

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

Minimum Pulse Time		Maximum Pulse Time		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
2 × TI00 input cycle		2^{16} × TI00 input cycle		TI00 input edge cycle	
—	$2 \times 1/f_X$ (400 ns)	—	$2^{16} \times 1/f_X$ (13.1 ms)	—	$1/f_X$ (200 ns)
$2 \times 1/f_X$ (400 ns)	$2^2 \times 1/f_X$ (800 ns)	$2^{16} \times 1/f_X$ (13.1 ms)	$2^{17} \times 1/f_X$ (26.2 ms)	$1/f_X$ (200 ns)	$2 \times 1/f_X$ (400 ns)
$2^2 \times 1/f_X$ (800 ns)	$2^3 \times 1/f_X$ (1.6 μs)	$2^{17} \times 1/f_X$ (26.2 ms)	$2^{18} \times 1/f_X$ (52.4 ms)	$2 \times 1/f_X$ (400 ns)	$2^2 \times 1/f_X$ (800 ns)
$2^3 \times 1/f_X$ (1.6 μs)	$2^4 \times 1/f_X$ (3.2 μs)	$2^{18} \times 1/f_X$ (52.4 ms)	$2^{19} \times 1/f_X$ (104.9 ms)	$2^2 \times 1/f_X$ (800 ns)	$2^3 \times 1/f_X$ (1.6 μs)
2 × watch timer output cycle		2^{16} × watch timer output cycle		Watch timer output edge cycle	

Remarks 1. f_X: Main system clock oscillation frequency

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

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

(6) One-shot pulse output

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

8.3 16-Bit Timer/Event Counter Configuration

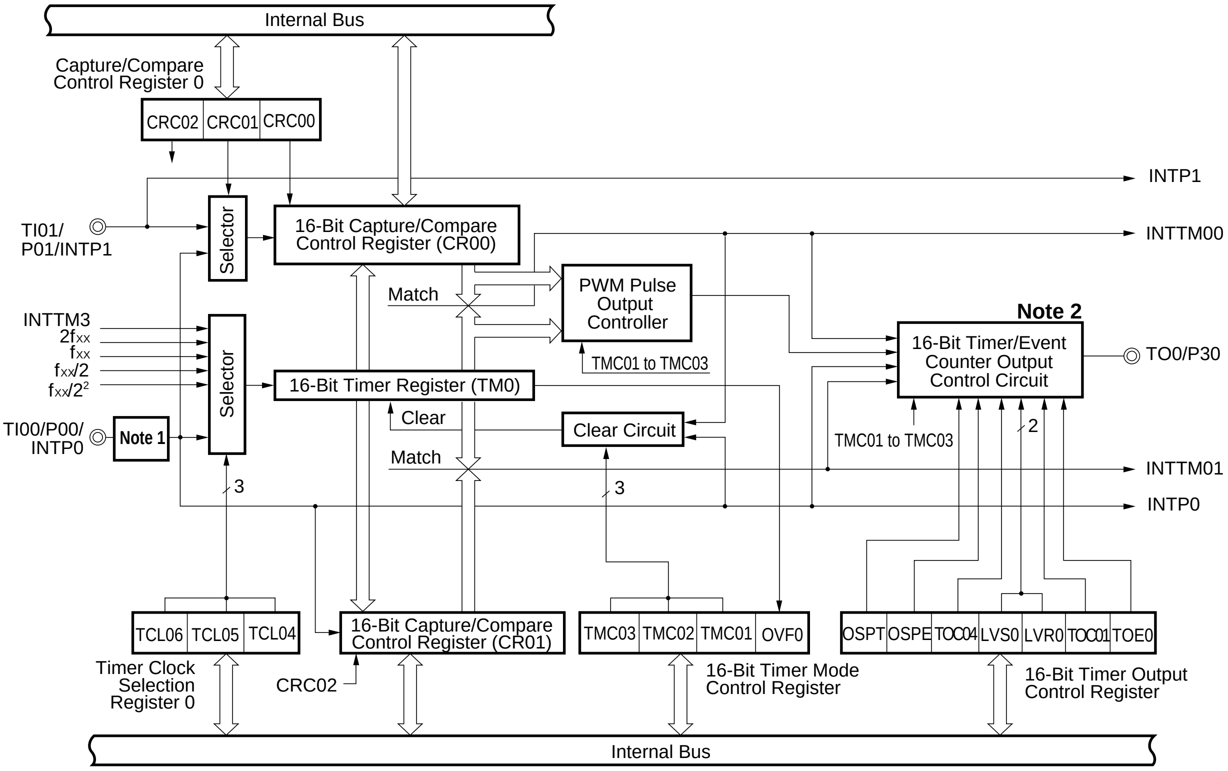
The 16-bit timer/event counter consists of the following hardware.

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

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

Note Refer to **Figure 21-1 Basic Configuration of Interrupt Function**.

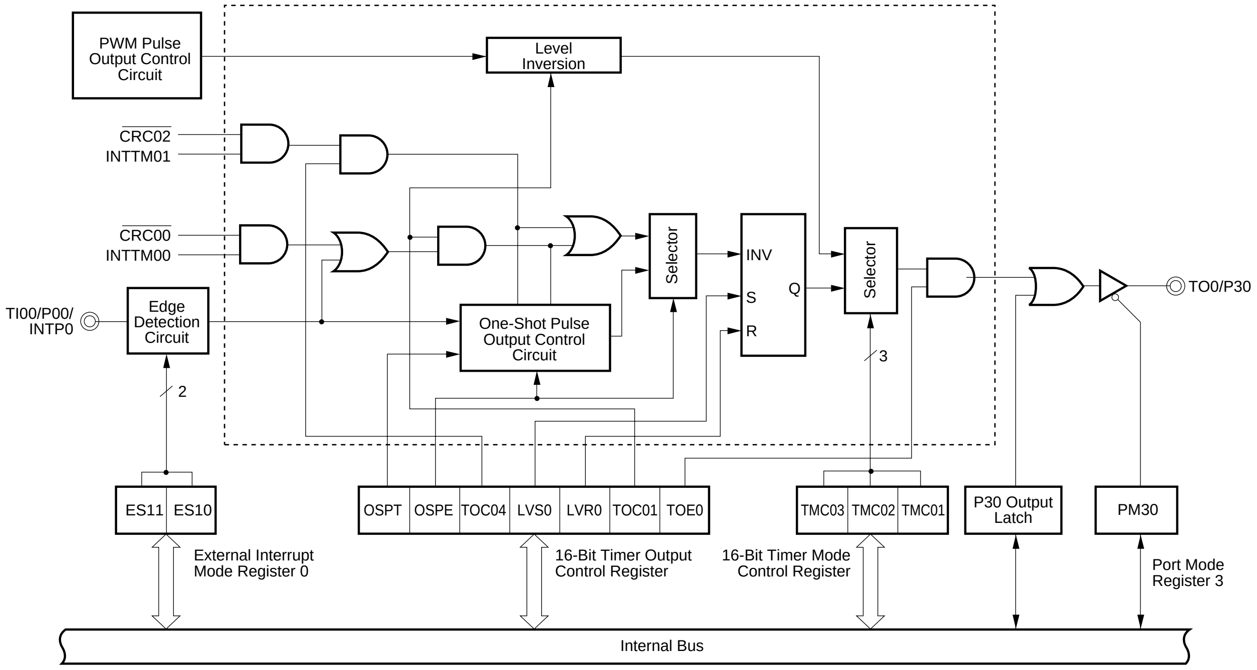
Figure 8-1. 16-bit Timer/Event Counter Block Diagram


Internal Bus
Capture/Compare Control Register 0
CRC02
CRC01
CRC00
INTP1
TI01/ P01/INTP1
Selector
16-Bit Capture/Compare Control Register (CR00)
INTTM00
Match
PWM Pulse Output Controller
Note 2
16-Bit Timer/Event Counter Output Control Circuit
TO0/P30
INTTM3
2fxx
fxx
fxx/2
fxx/2²
Selector
16-Bit Timer Register (TM0)
TMC01 to TMC03
TI00/P00/ INTP0
Note 1
Clear
Clear Circuit
TMC01 to TMC03
2
Match
INTTM01
3
3
INTP0
TCL06
TCL05
TCL04
16-Bit Capture/Compare Control Register (CR01)
TMC03
TMC02
TMC01
OVF0
OSPT
OSPE
TOC04
LVS0
LVR0
TOC01
TOE0
Timer Clock Selection Register 0
CRC02
16-Bit Timer Mode Control Register
16-Bit Timer Output Control Register
Internal Bus

Notes 1. Edge detection circuit

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

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


PWM Pulse Output Control Circuit
Level Inversion
$\overline{\text{CRC02}}$
INTTM01
$\overline{\text{CRC00}}$
INTTM00
TI00/P00/INTP0
Edge Detection Circuit
2
One-Shot Pulse Output Control Circuit
Selector
INV
S
R
Q
Selector
3
TO0/P30
ES11
ES10
OSPT
OSPE
TOC04
LVS0
LVR0
TOC01
TOE0
TMC03
TMC02
TMC01
P30 Output Latch
PM30
External Interrupt Mode Register 0
16-Bit Timer Output Control Register
16-Bit Timer Mode Control Register
Port Mode Register 3
Internal Bus

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

(1) Capture/compare register 00 (CR00)

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

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

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

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

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

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

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

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

Cautions 1. Set the data of PWM (14 bits) to the higher 14 bits of CR00. At this time, clear the lower 2 bits to 00.

2. Set a value other than 0000H to CR00. When the event counter function is used, therefore, one pulse cannot be counted.

3. If the new value of CR00 is less than the value of the 16-bit timer register (TM0), TM0 continues counting, overflows, and then starts counting again from 0. If the new value of CR00 is less than the old value, the timer must be restarted after changing the value of CR00.

(2) Capture/compare register 01 (CR01)

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

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

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

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

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

Caution If the valid edge of the TIO0/P00 pin is input while CR01 is read, CR01 does not perform the capture operation and retains the current data. However, the interrupt request flag (PIF0) is set.

(3) 16-bit timer register (TM0)

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

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

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

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

8.4 16-Bit Timer/Event Counter Control Registers

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

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

(1) Timer clock select register 0 (TCL0)

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

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

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

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

Figure 8-3. Timer Clock Select Register 0 Format

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

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

TCL06	TCL05	TCL04	16-Bit Timer Register Count Clock Selection		
				MCS = 1	MCS = 0
0	0	0	TI00 (Valid edge specifiable)		
0	0	1	$2f_{XX}$	Setting prohibited	f_X (5.0 MHz)
0	1	0	f_{XX}	f_X (5.0 MHz)	$f_X/2$ (2.5 MHz)
0	1	1	$f_{XX}/2$	$f_X/2$ (2.5 MHz)	$f_X/2^2$ (1.25 MHz)
1	0	0	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	1	1	Watch timer output (INTTM 3)		
Other than above			Setting prohibited		

CLOE	PCL Output Control
0	Output disabled
1	Output enabled

Cautions 1. TI00/INTP0 pin valid edge is set by external interrupt mode register 0 (INTM0), and the sampling clock frequency is selected by the sampling clock selection register (SCS).

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

3. To read the count value when TI00 has been specified as the TM0 count clock, the value should be read from TM0, not from 16-bit capture/compare register 01 (CR01).

4. When rewriting TCL0 to other data, stop the timer operation beforehand.

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

2. f_X : Main system clock oscillation frequency

3. f_{XT} : Subsystem clock oscillation frequency

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

5. TM0 : 16-bit timer register

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

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

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

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

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

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

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

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

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

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

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

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

2. Set the valid edge of the TI00/INTP0 pin with an external interrupt mode register 0 (INTM0) and select the sampling clock frequency with a sampling clock select register (SCS).

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

4. 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 register

4. CR00 : Compare register 00

5. CR01 : Compare register 01

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

This register controls the operation of the capture/compare registers 00, 01 (CR00, CR01).

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

$\overline{\text{RESET}}$ input sets CRC0 to 04H.

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

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
CRC0	0	0	0	0	0	CRC02	CRC01	CRC00	FF4CH	04H	R/W

CRC00	CR00 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

CRC02	CR01 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 (TMC0), CR00 should not be specified as a capture register.

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

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

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

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

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

Symbol	7	⑥	⑤	4	③	②	1	⓪	Address	After Reset	R/W
TOC0	0	OSPT	OSPE	TOC04	LVS0	LVR0	TOC01	TOE0	FF4EH	00H	R/W

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

TOC01	In PWM Mode	In Other Modes
	Active level selection	Timer output F/F control by match of CR00 and TM0
0	Active high	Inversion operation disabled
1	Active low	Inversion operation enabled

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

TOC04	Timer Output F/F Control by Match of CR01 and TM0
0	Inversion operation disabled
1	Inversion operation enabled

OSPE	One-Shot Pulse Output Control
0	Continuous pulse output
1	One-shot pulse output

OSPT	Control of One-Shot Pulse Output Trigger by Software
0	One-shot pulse trigger not used
1	One-shot pulse trigger used

Cautions 1. Timer operation must be stopped before setting TOC0 (however, except OSPT).

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

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

(5) Port mode register 3 (PM3)

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

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

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

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

Figure 8-7. Port Mode Register 3 Format

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

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

(6) External interrupt mode register 0 (INTM0)

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

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

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

Figure 8-8. External Interrupt Mode Register 0 Format

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

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

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

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

Caution Before setting the valid edge of the INTP0/TI00/P00 pin, be sure to set bits 1 to 3 (TMC01 to TMC03) of the 16-bit timer mode control register (TMC0) to 0, 0, 0 to stop the timer operation.

(7) Sampling clock select registers (SCS)

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

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

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

Figure 8-9. Sampling Clock Select Register Format

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

SCS1	SCS0	INTP0 Sampling Clock Selection		
			MCS = 1	MCS = 0
0	0	$f_{XX}/2^N$		
0	1	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	0	$f_{XX}/2^5$	$f_X/2^5$ (156.3 kHz)	$f_X/2^6$ (78.1 kHz)
1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)

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

Remarks

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

2. f_{XX} : Main system clock frequency (f_X or $f_X/2$)

3. f_X : Main system clock oscillation frequency

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

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

8.5 16-Bit Timer/Event Counter Operations

8.5.1 Interval timer operations

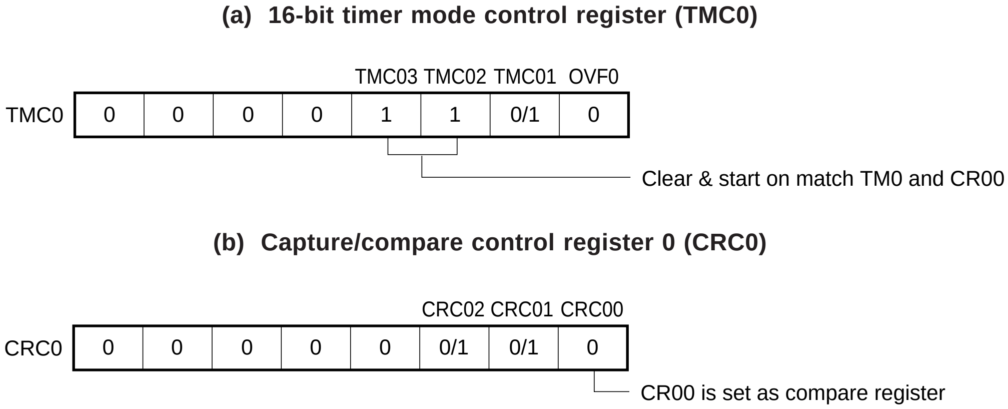
Setting the 16-bit timer mode control register (TMC0) and capture/compare control register 0 (CRC0) as shown in Figure 8-10 allows operation as an interval timer. Interrupt requests are generated repeatedly using the count value set in 16-bit capture/compare register 00 (CR00) beforehand as the interval.

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

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

For the operation when the value of the compare register is changed during the timer/counter operation, refer to **8.6 (3) Operation after compare register change during timer count operation**.

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



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

					TMC03	TMC02	TMC01	OVF0
TMC0	0	0	0	0	1	1	0/1	0

Clear & start on match TM0 and CR00

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

						CRC02	CRC01	CRC00
CRC0	0	0	0	0	0	0/1	0/1	0

CR00 is set as compare register

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

Figure 8-11. Interval Timer Configuration Diagram

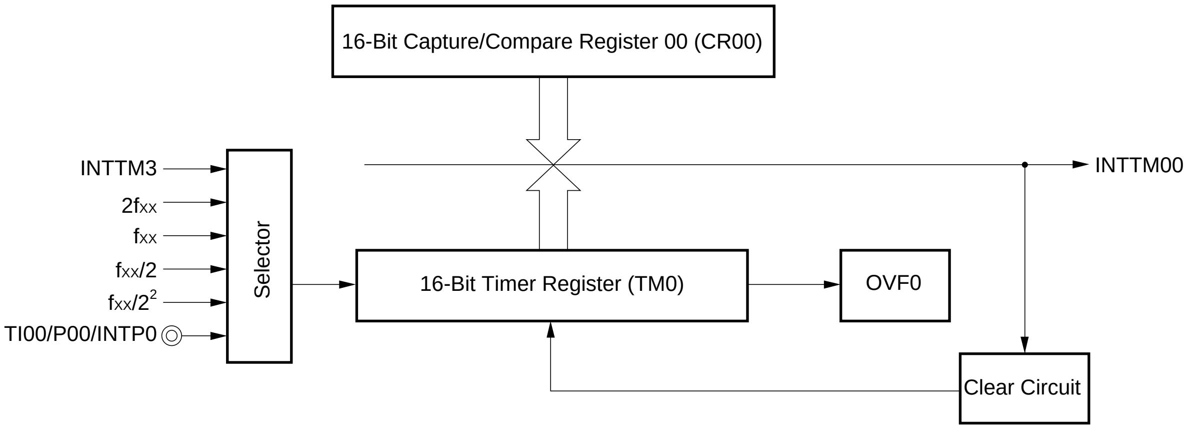
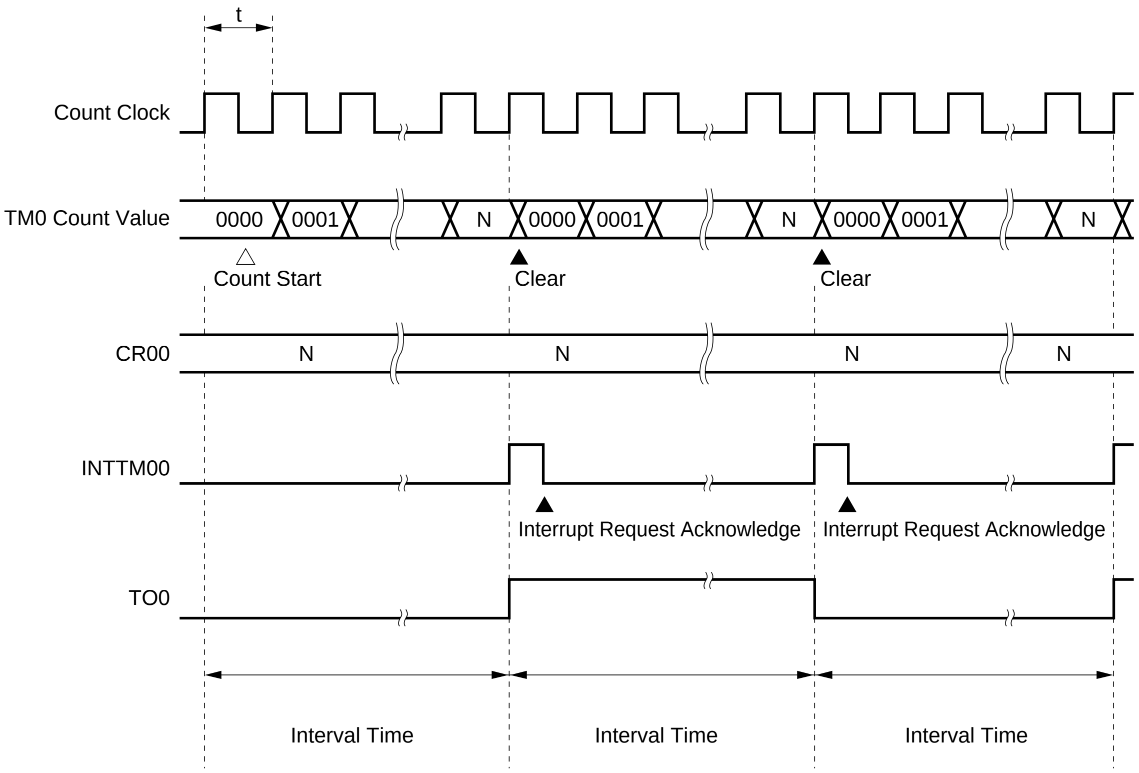


Figure 8-12. Interval Timer Operation Timings



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

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

TCL06	TCL05	TCL04	Minimum Interval Time		Maximum Interval Time		Resolution	
			MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
0	0	0	2 × TI00 input cycle		2^{16} × TI00 input cycle		TI00 input edge cycle	
0	0	1	Setting prohibited	2 × 1/fx (400 ns)	Setting prohibited	2^{16} × 1/fx (13.1 ms)	Setting prohibited	1/fx (200 ns)
0	1	0	2 × 1/fx (400 ns)	2^2 × 1/fx (800 ns)	2^{16} × 1/fx (13.1 ms)	2^{17} × 1/fx (26.2 ms)	1/fx (200 ns)	2 × 1/fx (400 ns)
0	1	1	2^2 × 1/fx (800 ns)	2^3 × 1/fx (1.6 μs)	2^{17} × 1/fx (26.2 ms)	2^{18} × 1/fx (52.4 ms)	2 × 1/fx (400 ns)	2^2 × 1/fx (800 ns)
1	0	0	2^3 × 1/fx (1.6 μs)	2^4 × 1/fx (3.2 μs)	2^{18} × 1/fx (52.4 ms)	2^{19} × 1/fx (104.9 ms)	2^2 × 1/fx (800 ns)	2^3 × 1/fx (1.6 μs)
1	1	1	2 × watch timer output cycle		2^{16} × watch timer output cycle		Watch timer output edge cycle	
Other than above			Setting prohibited					

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

8.5.2 PWM output operations

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

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

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

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

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

Figure 8-13. Control Register Settings for PWM Output Operation

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

					TMC03	TMC02	TMC01	OVF0
TMC0	0	0	0	0	0	0	1	0

TMC03, TMC02, TMC01: PWM mode

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

						CRC02	CRC01	CRC00
CRC0	0	0	0	0	0	0/1	0/1	0

CRC00: CR00 is set as compare register

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

		OSPT	OSPE	TOC04	LVS0	LVR0	TOC01	TOE0
TOC0	0	×	×	×	×	×	0/1	1

TOE0: TO0 Output Enabled

TOC01: Specifies Active Level

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

× : don't care

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

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

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

V_{REF}: External switching circuit reference voltage

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

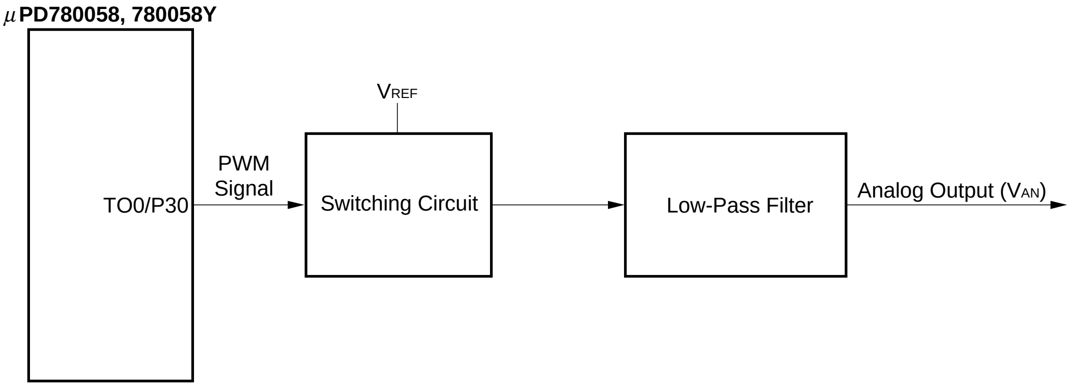
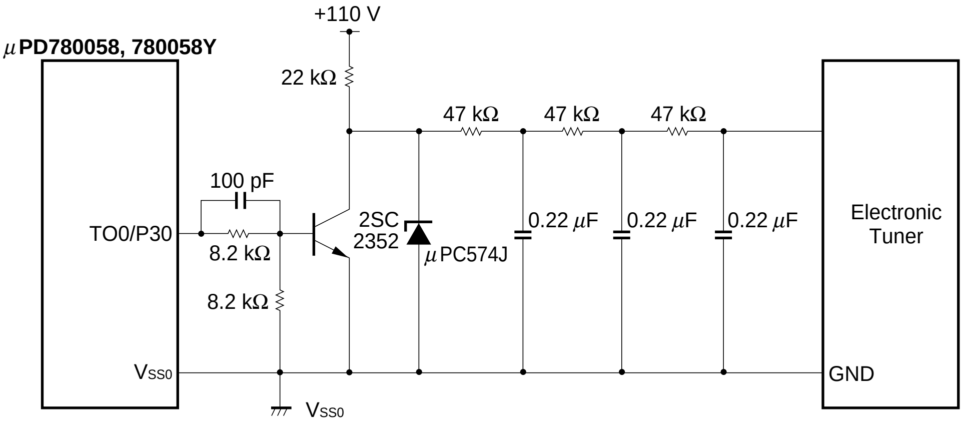


Figure 8-15 shows an example in which PWM output is converted to an analog voltage and used in a voltage synthesizer type TV tuner.

Figure 8-15. TV Tuner Application Circuit Example

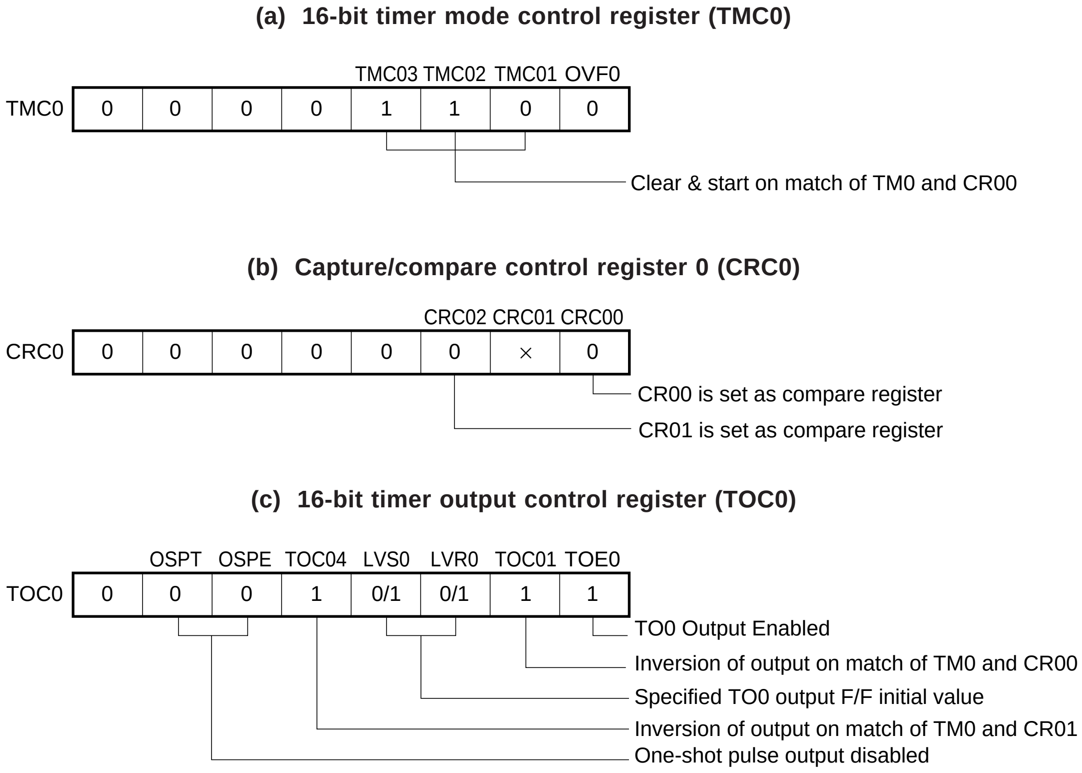


8.5.3 PPG output operations

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

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

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



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

TMC03 TMC02 TMC01 OVF0
TMC0
0 0 0 0 1 1 0 0
Clear & start on match of TM0 and CR00

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

CRC02 CRC01 CRC00
CRC0
0 0 0 0 0 0 × 0
CR00 is set as compare register
CR01 is set as compare register

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

OSPT OSPE TOC04 LVS0 LVR0 TOC01 TOE0
TOC0
0 0 0 1 0/1 0/1 1 1
TO0 Output Enabled
Inversion of output on match of TM0 and CR00
Specified TO0 output F/F initial value
Inversion of output on match of TM0 and CR01
One-shot pulse output disabled

Remark × : don't care

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

8.5.4 Pulse width measurement operations

It is possible to measure the pulse width of the signals input to the TI00/P00 pin and TI01/P01 pin using the 16-bit timer register (TM0).

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

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

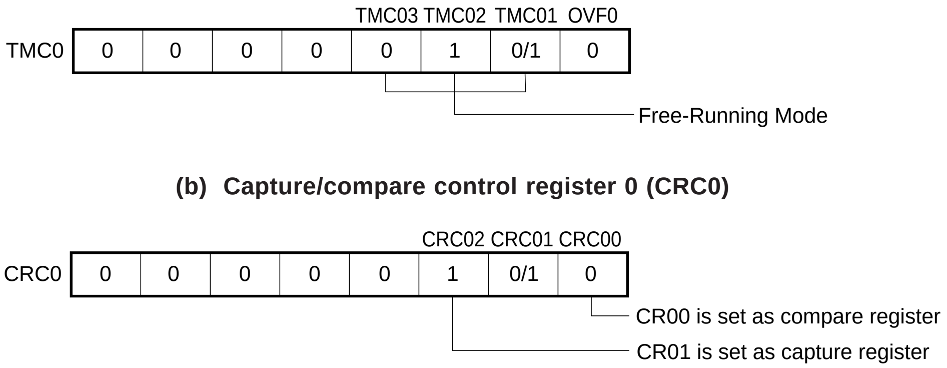
When the 16-bit timer register (TM0) is operated in free-running mode (see register settings in Figure 8-17), and the edge specified by external interrupt mode register 0 (INTM0) is input to the TI00/P00 pin, the value of TM0 is taken into 16-bit capture/compare register 01 (CR01) and an external interrupt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

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

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

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

(a) 16-bit timer mode control register (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 8-18. Configuration Diagram for Pulse Width Measurement by Free-Running Counter

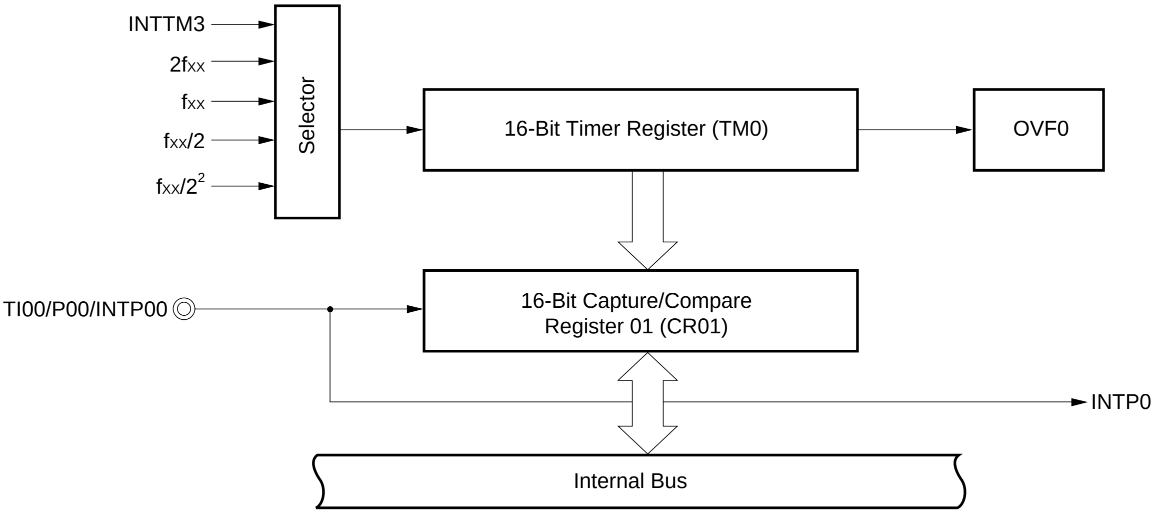
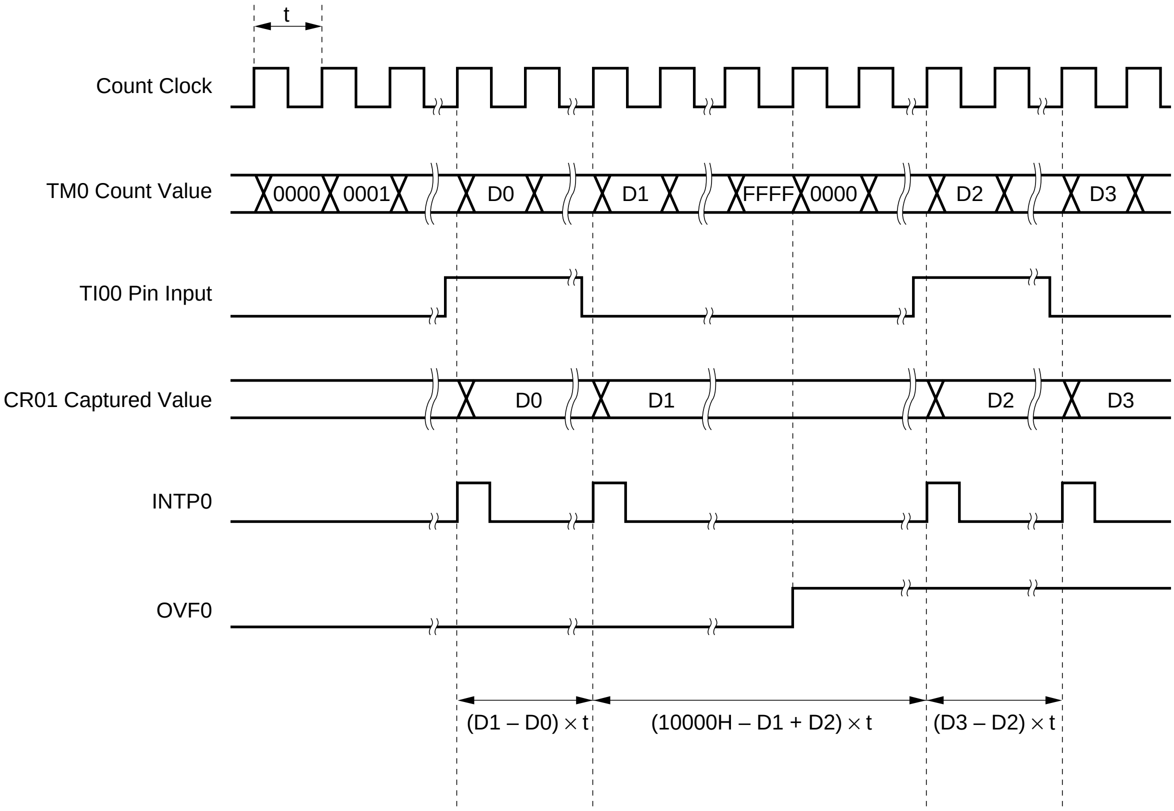

INTTM3
2fXX
fXX
fXX/2
fXX/2²
Selector
16-Bit Timer Register (TM0)
OVF0
TI00/P00/INTP00
16-Bit Capture/Compare Register 01 (CR01)
INTP0
Internal Bus

Figure 8-19. 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 the 16-bit timer register (TM0) is operated in free-running mode (see register settings in Figure 8-20), it is possible to simultaneously measure the pulse widths of the two signals input to the TI00/P00 pin and the TI01/P01 pin.

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

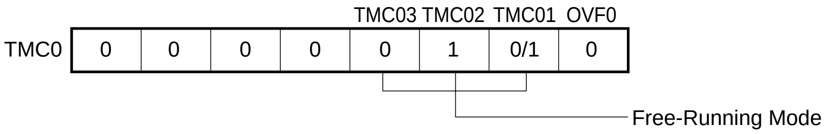
Also, when the edge specified by bits 4 and 5 (ES20 and ES21) of INTM0 is input to the TI01/P01 pin, the value of TM0 is taken into 16-bit capture/compare register 00 (CR00) and an external interrupt request signal (INTP1) is set.

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

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

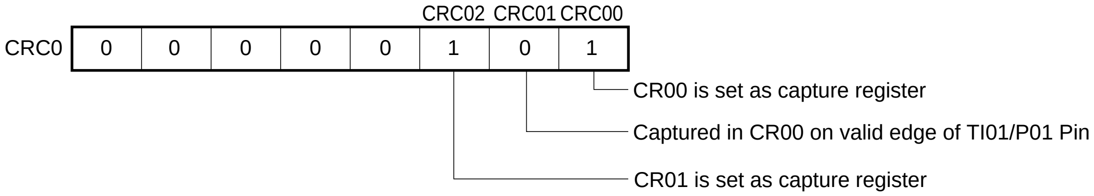
Figure 8-20. Control Register Settings for Two Pulse Width Measurements with Free-Running Counter

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



					TMC03	TMC02	TMC01	OVF0
TMC0	0	0	0	0	0	1	0/1	0

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

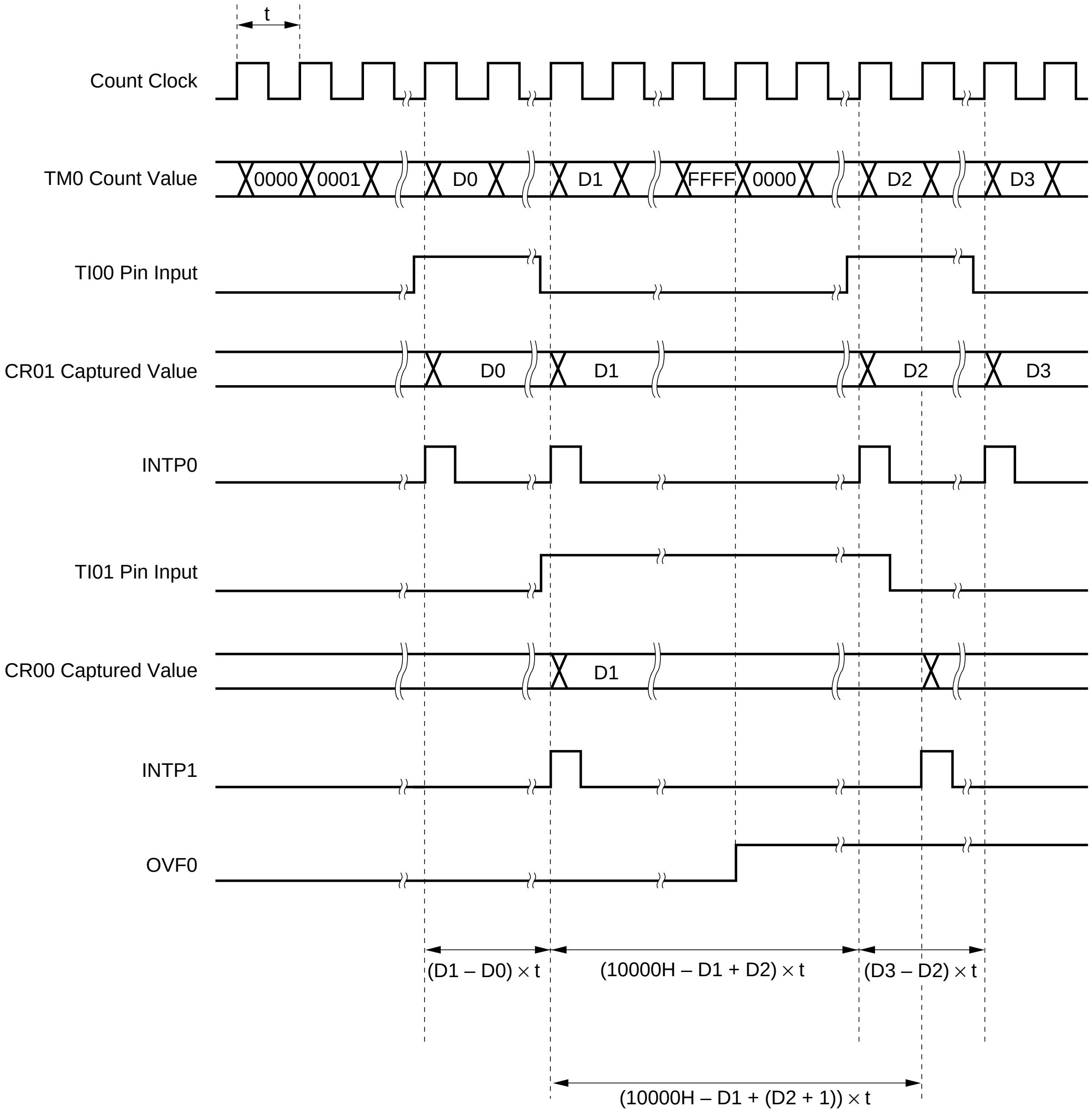


						CRC02	CRC01	CRC00
CRC0	0	0	0	0	0	1	0	1

CR00 is set as capture register
Captured in CR00 on valid edge of TI01/P01 Pin
CR01 is set as capture register

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

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



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

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

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

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

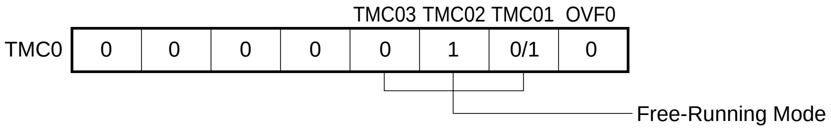
Either of two edge specifications can be selected —rising or falling— as the valid edges for the TI00/P00 pin by means of bits 2 and 3 (ES10 and ES11) of INTM0.

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

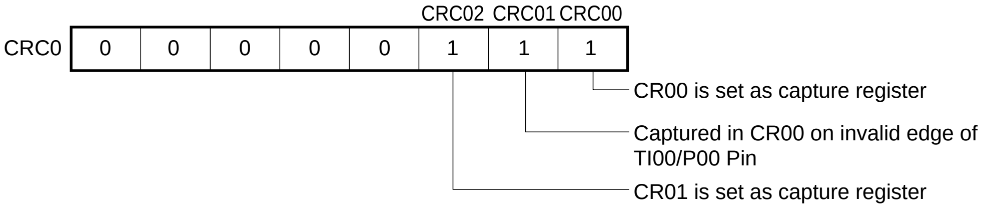
Caution If the valid edge of TI00/P00 is specified to be both rising and falling edge, capture/compare register 00 (CR00) cannot perform the capture operation.

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

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

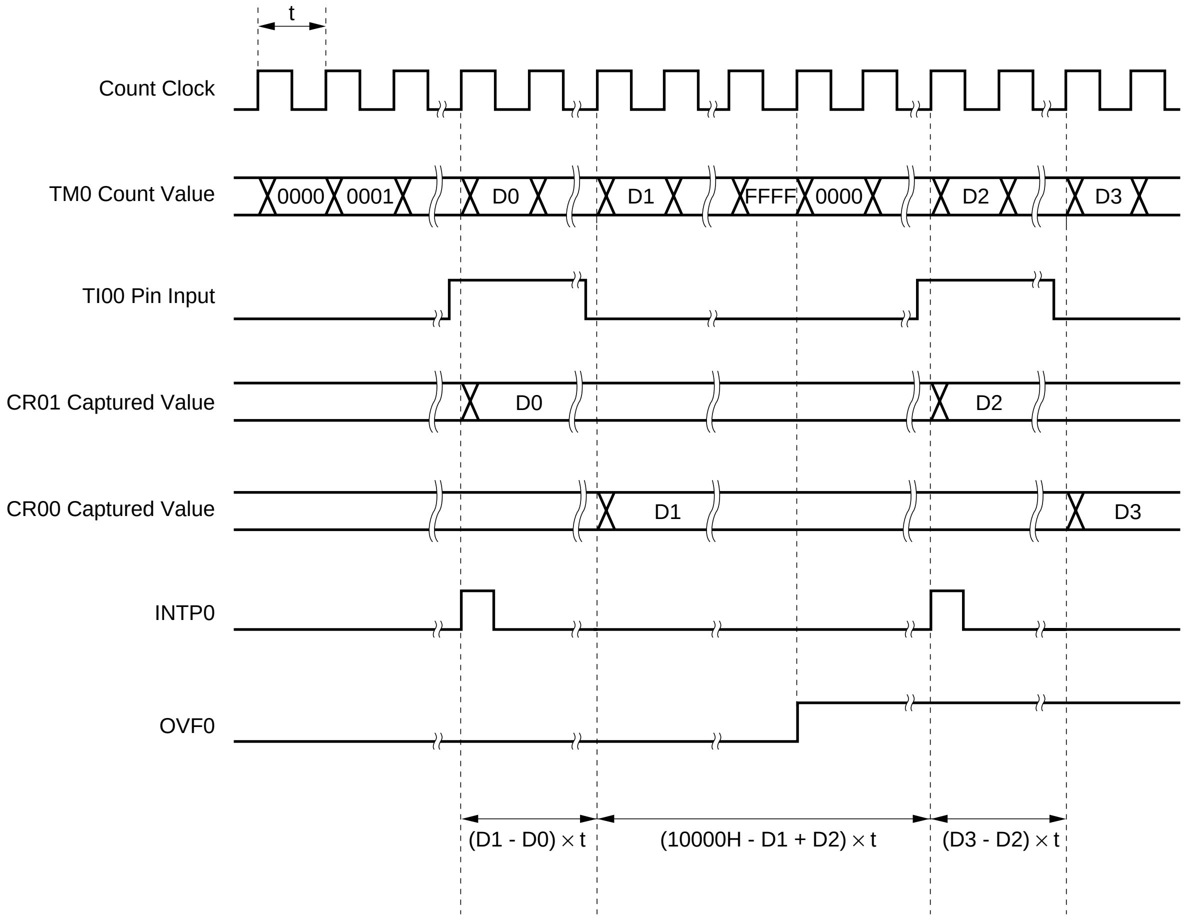


(b) Capture/compare control register 0 (CRC0)


CRC02
CRC01
CRC00
CRC0
0
0
0
0
0
1
1
1
CR00 is set as capture register
Captured in CR00 on invalid edge of TI00/P00 Pin
CR01 is set as capture register

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

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



(4) Pulse width measurement by means of restart

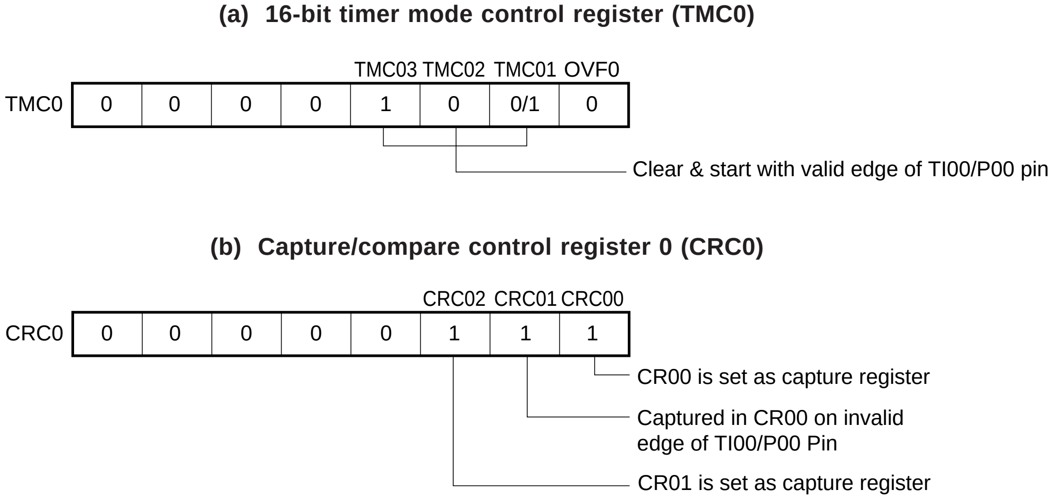
When input of a valid edge to the TI00/P00 pin is detected, the count value of the 16-bit timer register (TM0) is taken into 16-bit capture/compare register 01 (CR01), and then the pulse width of the signal input to the TI00/P00 pin is measured by clearing TM0 and restarting the count (see register settings in Figure 8-24).

The edge specification can be selected from two types, rising and falling edges by external interrupt mode register 0 (INTM0) bits 2 and 3 (ES10 and ES11).

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

Caution If the valid edge of TI00/P00 is specified to be both rising and falling edge, the 16-bit capture/compare register 00 (CR00) cannot perform the capture operation.

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



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

					TMC03	TMC02	TMC01	OVF0
TMC0	0	0	0	0	1	0	0/1	0

Clear & start with valid edge of TI00/P00 pin

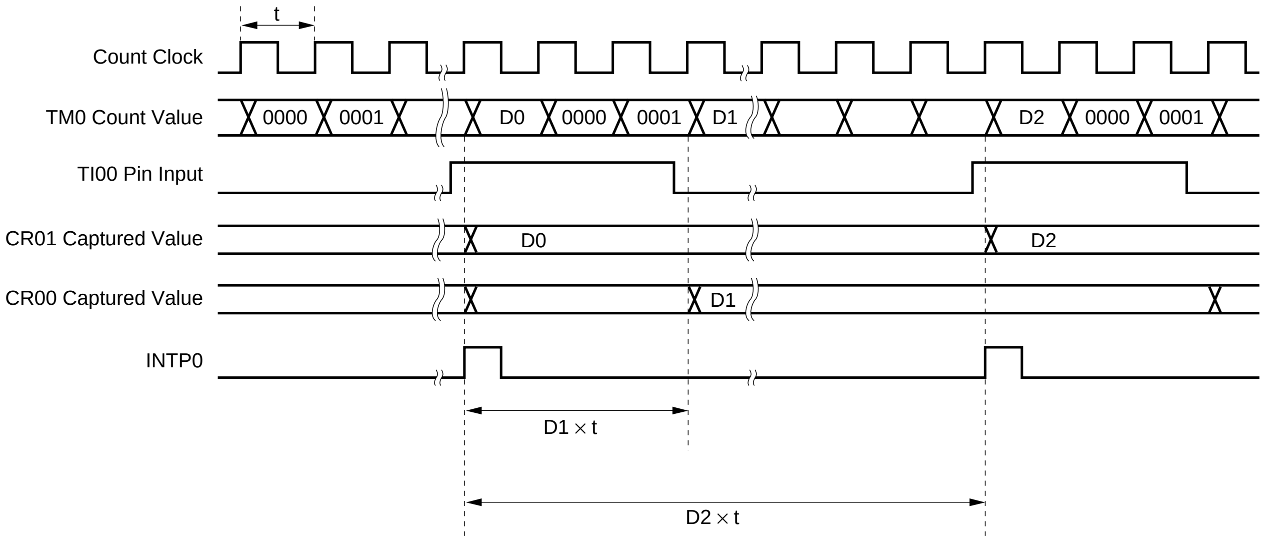
(b) Capture/compare control register 0 (CRC0)

						CRC02	CRC01	CRC00
CRC0	0	0	0	0	0	1	1	1

CR00 is set as capture register
Captured in CR00 on invalid edge of TI00/P00 Pin
CR01 is set as capture register

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

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



8.5.5 External event counter operation

The external event counter counts the number of external clock pulses to be input to the TI00/P00 pin with the 16-bit timer register (TM0).

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

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

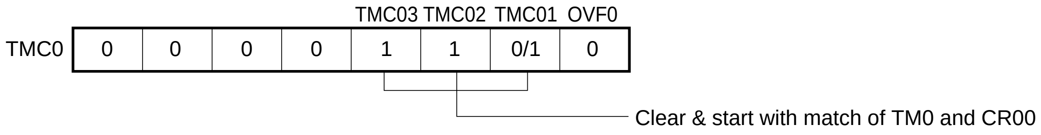
Set a value other than 0000H to CR00 (1-pulse count operation cannot be performed).

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

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

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

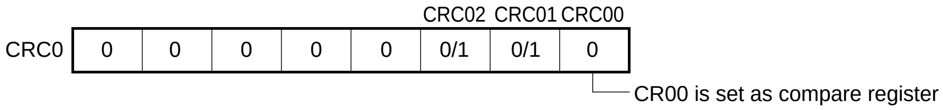
(a) 16-bit timer mode control register (TMC0)



					TMC03	TMC02	TMC01	OVF0
TMC0	0	0	0	0	1	1	0/1	0

Clear & start with match of TM0 and CR00

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



						CRC02	CRC01	CRC00
CRC0	0	0	0	0	0	0/1	0/1	0

CR00 is set as compare register

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

Figure 8-27. External Event Counter Configuration Diagram

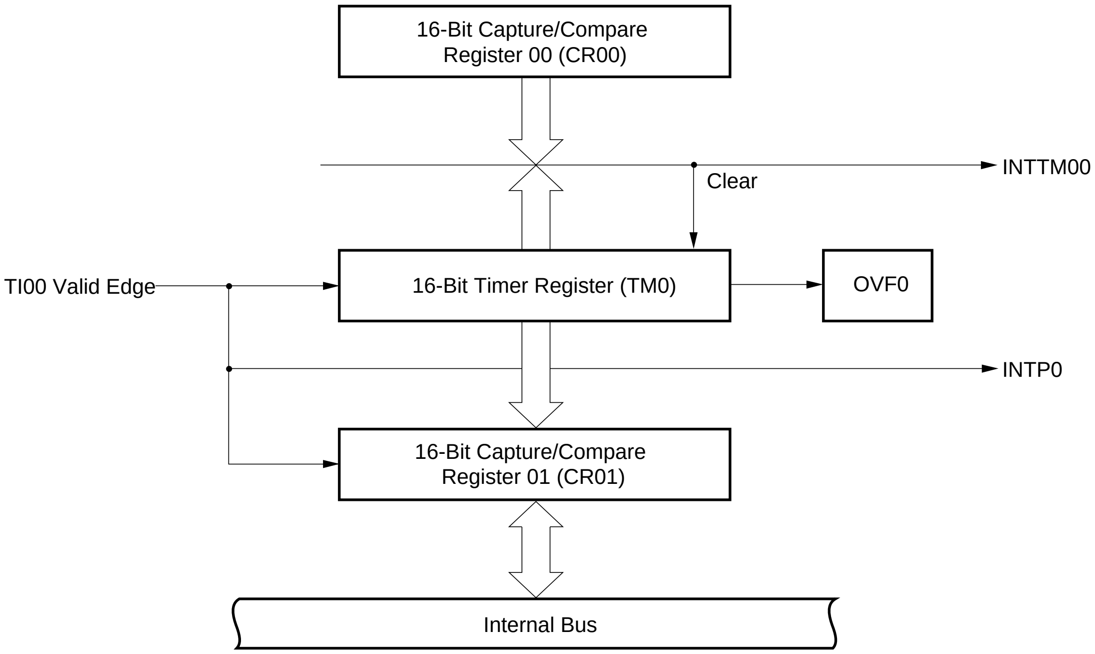
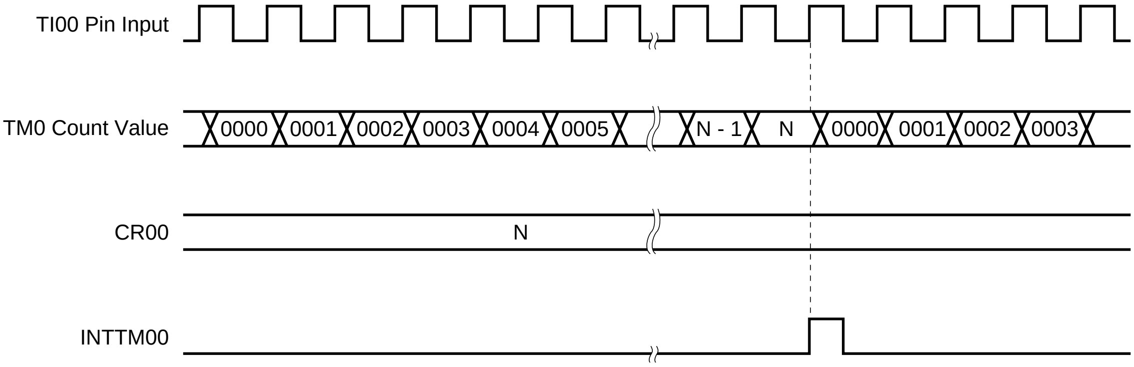


Figure 8-28. External Event Counter Operation Timings (with Rising Edge Specified)



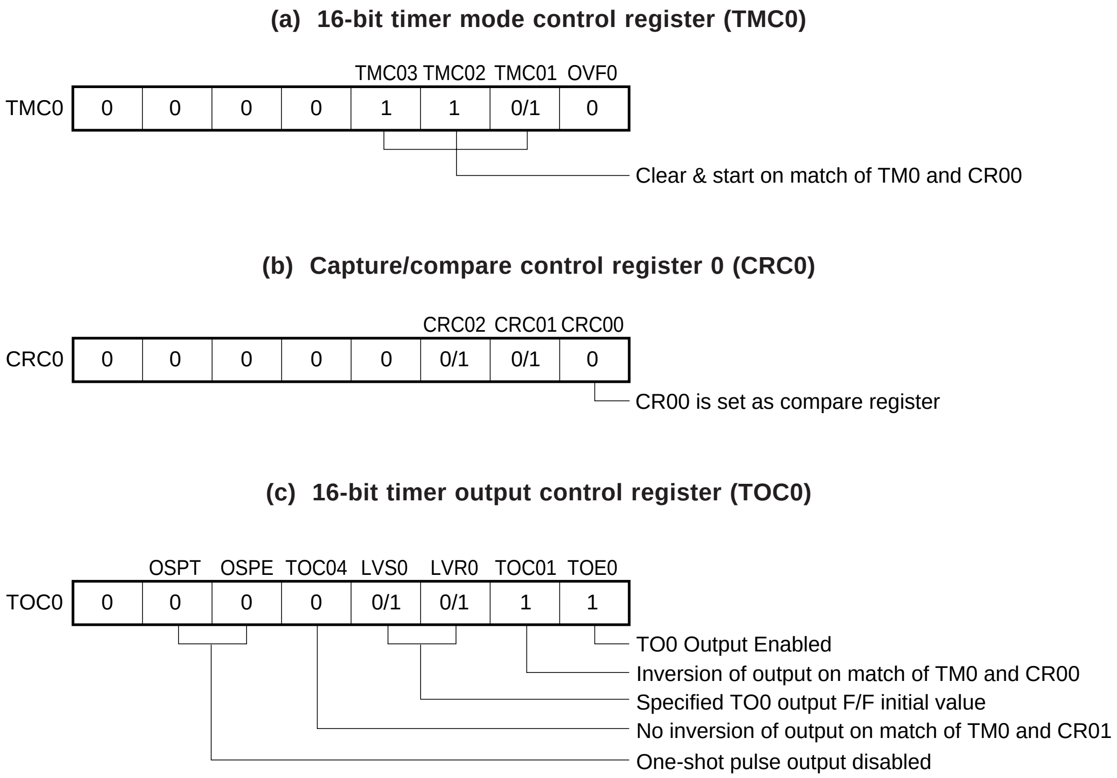
Caution When reading the external event counter count value, TM0 should be read.

8.5.6 Square-wave output operation

The 16-bit timer/event counter outputs a square wave with any selected frequency at intervals specified by the count value set in advance to the 16-bit capture/compare register 00 (CR00).

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

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



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

TMC03 TMC02 TMC01 OVF0
TMC0
0 0 0 0 1 1 0/1 0
Clear & start on match of TM0 and CR00

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

CRC02 CRC01 CRC00
CRC0
0 0 0 0 0 0/1 0/1 0
CR00 is set as compare register

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

OSPT OSPE TOC04 LVS0 LVR0 TOC01 TOE0
TOC0
0 0 0 0 0/1 0/1 1 1
TO0 Output Enabled
Inversion of output on match of TM0 and CR00
Specified TO0 output F/F initial value
No inversion of output on match of TM0 and CR01
One-shot pulse output disabled

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

Figure 8-30. Square-Wave Output Operation Timing

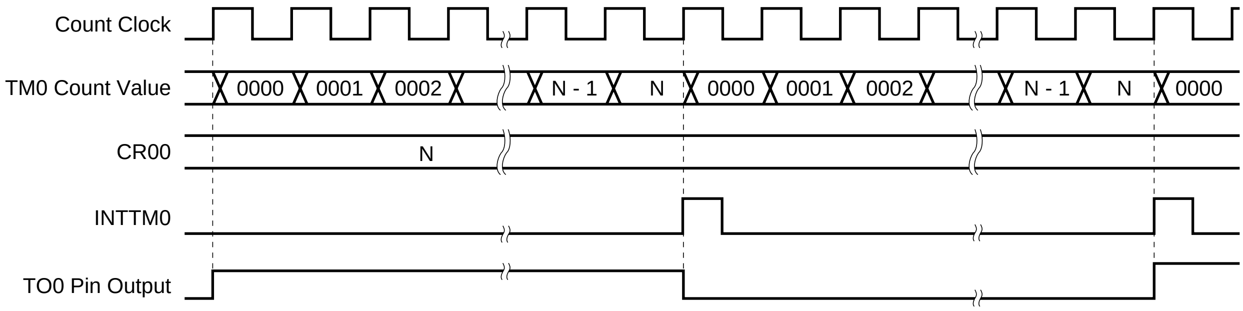


Table 8-7. 16-bit Timer/Event Counter Square-Wave Output Ranges

Minimum Pulse Time		Maximum Pulse Time		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
2 × TI00 input cycle		2^{16} × TI00 input cycle		TI00 input edge cycle	
—	2 × 1/f_X (400 ns)	—	2^{16} × 1/f_X (13.1 ms)	—	1/f_X (200 ns)
2 × 1/f_X (400 ns)	2^2 × 1/f_X (800 ns)	2^{16} × 1/f_X (13.1 ms)	2^{17} × 1/f_X (26.2 ms)	1/f_X (200 ns)	2 × 1/f_X (400 ns)
2^2 × 1/f_X (800 ns)	2^3 × 1/f_X (1.6 μs)	2^{17} × 1/f_X (26.2 ms)	2^{18} × 1/f_X (52.4 ms)	2 × 1/f_X (400 ns)	2^2 × 1/f_X (800 ns)
2^3 × 1/f_X (1.6 μs)	2^4 × 1/f_X (3.2 μs)	2^{18} × 1/f_X (52.4 ms)	2^{19} × 1/f_X (104.9 ms)	2^2 × 1/f_X (800 ns)	2^3 × 1/f_X (1.6 μs)
2 × watch timer output cycle		2^{16} × watch timer output cycle		Watch timer output edge cycle	

Remarks 1. f_X : Main system clock oscillation frequency

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

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

8.5.7 One-shot pulse output operation

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

(1) One-shot pulse output using software trigger

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

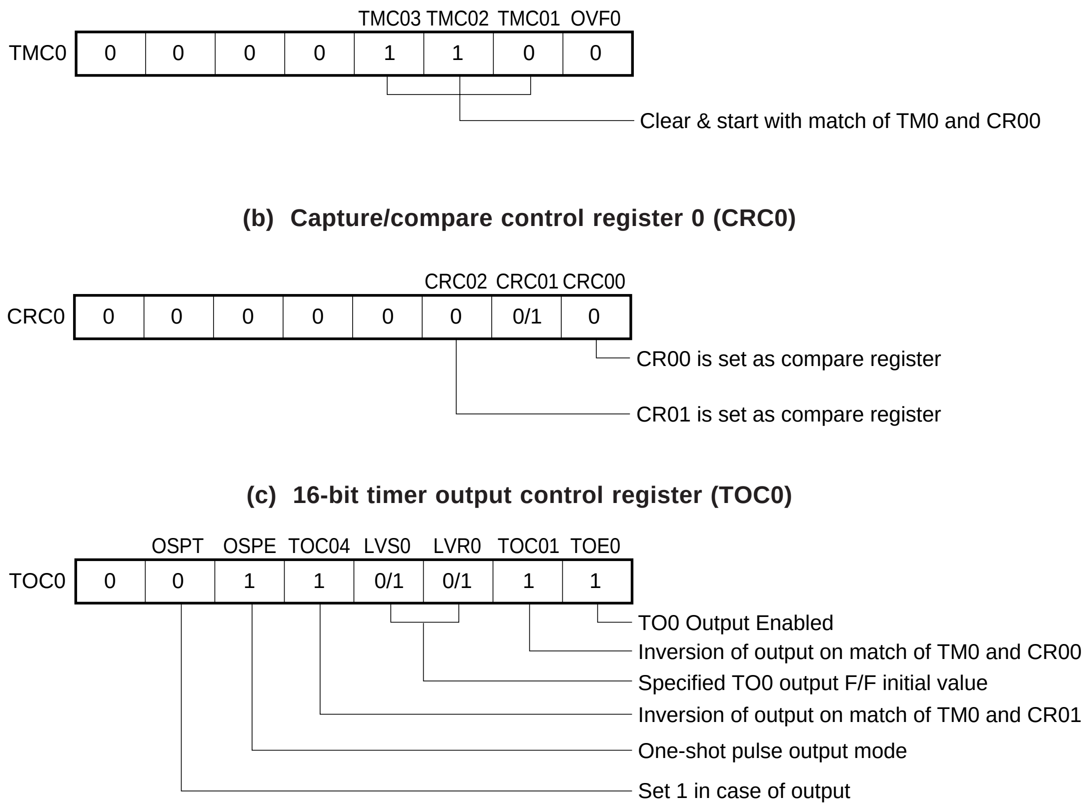
By setting 1 in OSPT, the 16-bit timer/event counter is cleared and started, and output is activated by the count value set beforehand in 16-bit capture/compare register 01 (CR01). Thereafter, output is inactivated by the count value set beforehand in 16-bit capture/compare register 00 (CR00).

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

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

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

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


TMC03 TMC02 TMC01 OVF0
TMC0 0 0 0 0 1 1 0 0
Clear & start with match of TM0 and CR00
CRC02 CRC01 CRC00
CRC0 0 0 0 0 0 0 0/1 0
CR00 is set as compare register
CR01 is set as compare register
OSPT OSPE TOC04 LVS0 LVR0 TOC01 TOE0
TOC0 0 0 1 1 0/1 0/1 1 1
TO0 Output Enabled
Inversion of output on match of TM0 and CR00
Specified TO0 output F/F initial value
Inversion of output on match of TM0 and CR01
One-shot pulse output mode
Set 1 in case of output

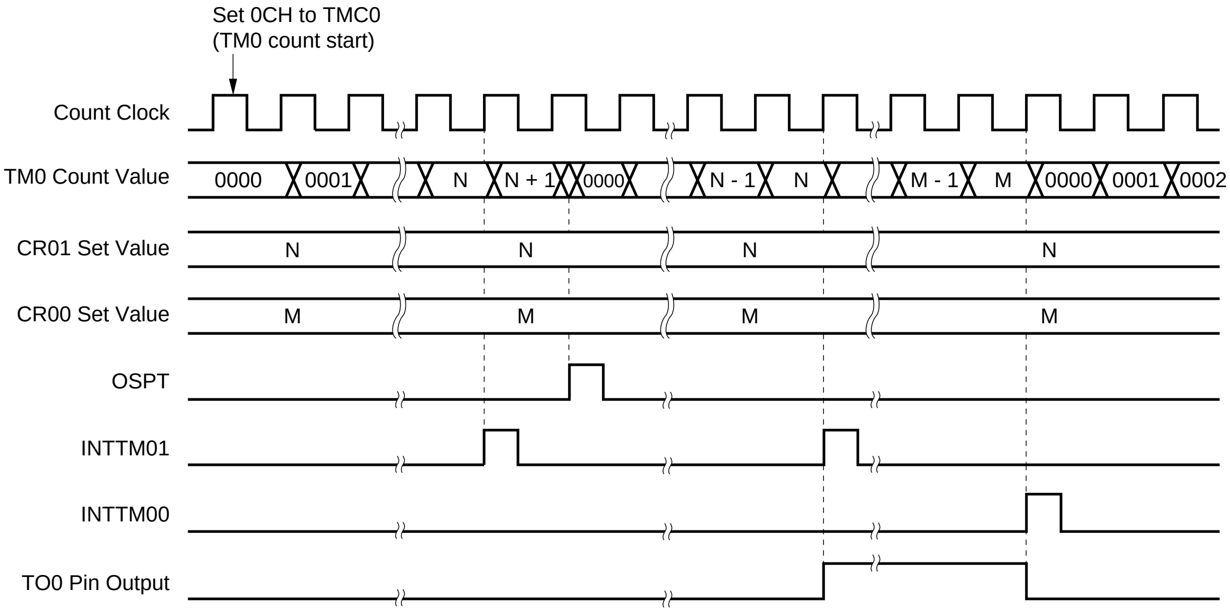
(b) Capture/compare control register 0 (CRC0)

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

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

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

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



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

(2) One-shot pulse output using external trigger

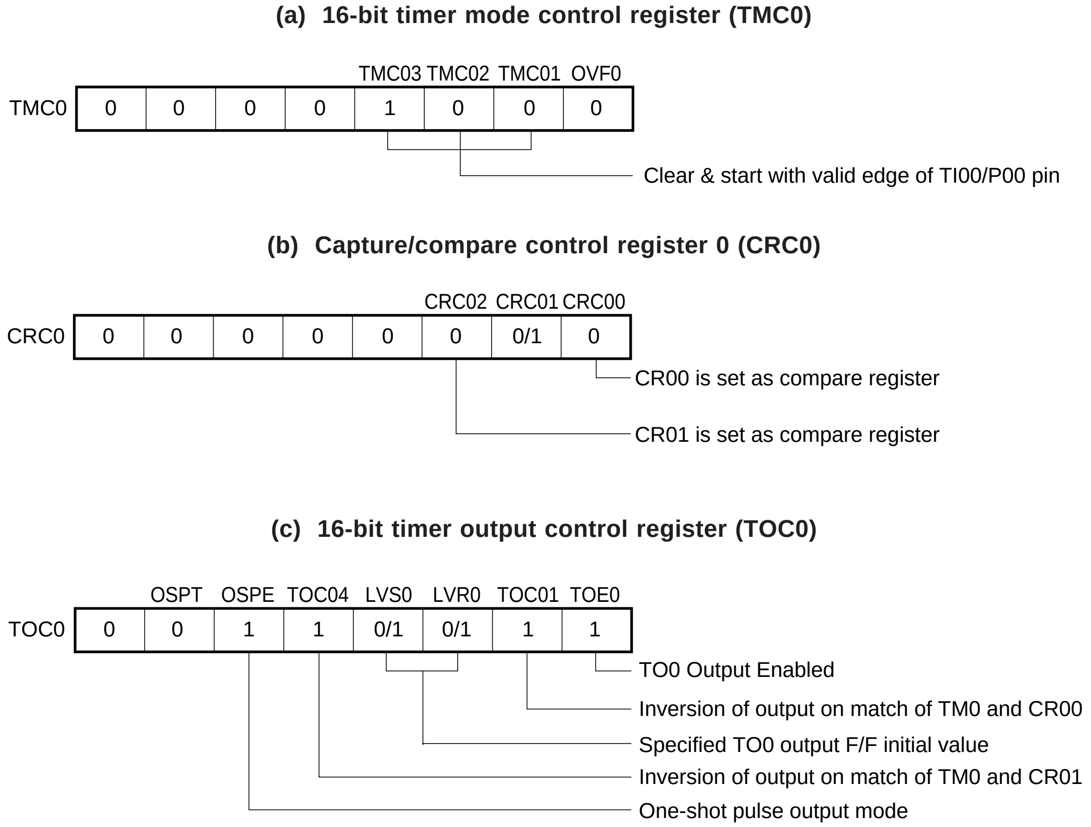
If the 16-bit timer mode control register (TMC0), capture/compare control register 0 (CRC0), and the 16-bit timer output control register (TOC0) are set as shown in Figure 8-33, a one-shot pulse is output from the TO0/ P30 pin with a TI00/P00 valid edge as an external trigger.

Any of three edge specifications can be selected —rising, falling, or both edges— as the valid edges for the TI00/P00 pin by means of bits 2 and 3 (ES10 and ES11) of external interrupt mode register 0 (INTM0).

When a valid edge is input to the TI00/P00 pin, the 16-bit timer/event counter is cleared and started, and output is activated by the count values set beforehand in 16-bit capture/compare register 01 (CR01). Thereafter, output is inactivated by the count value set beforehand in 16-bit capture/compare register 00 (CR00).

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

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



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

TMC03 TMC02 TMC01 OVF0
TMC0 0 0 0 0 1 0 0 0
Clear & start with valid edge of TI00/P00 pin

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

CRC02 CRC01 CRC00
CRC0 0 0 0 0 0 0 0/1 0
CR00 is set as compare register
CR01 is set as compare register

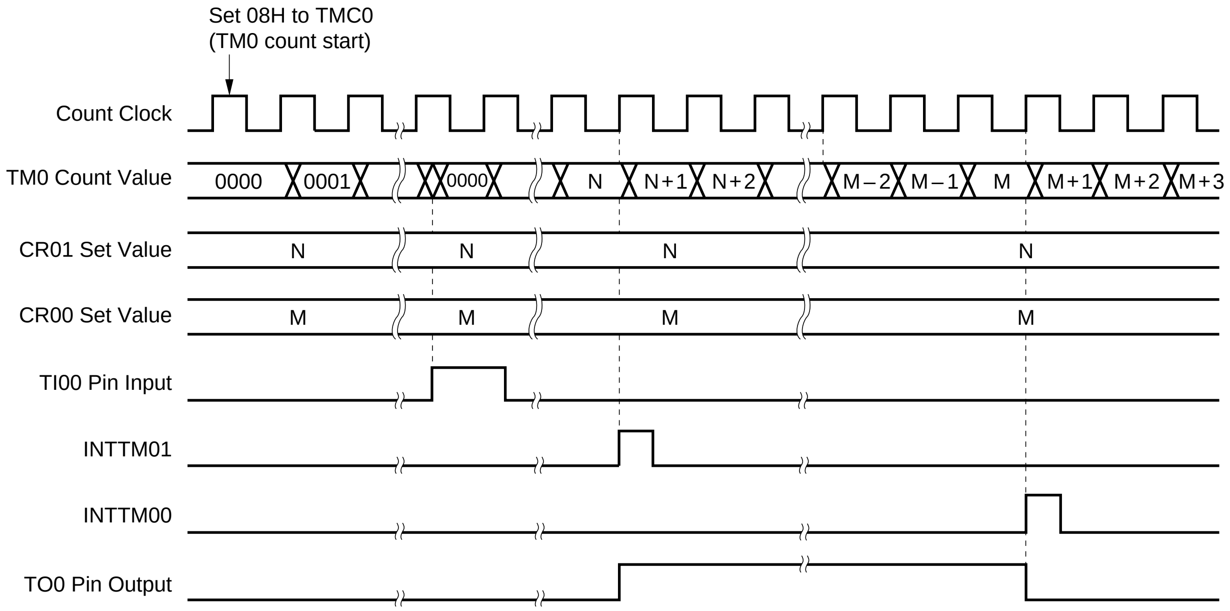
(c) 16-bit timer output control register (TOC0)

OSPT OSPE TOC04 LVS0 LVR0 TOC01 TOE0
TOC0 0 0 1 1 0/1 0/1 1 1
TO0 Output Enabled
Inversion of output on match of TM0 and CR00
Specified TO0 output F/F initial value
Inversion of output on match of TM0 and CR01
One-shot pulse output mode

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

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

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



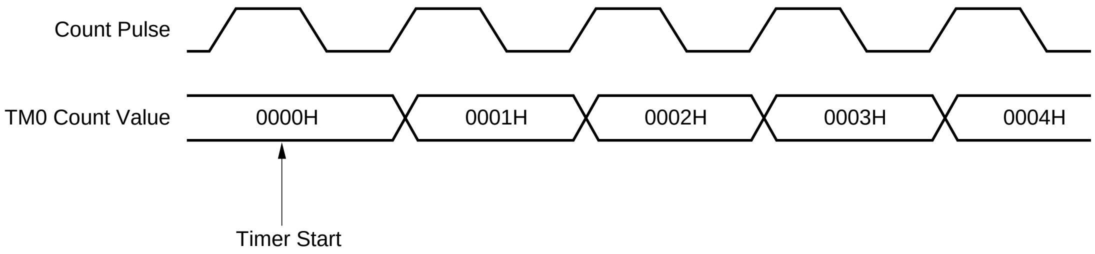
Caution **The 16-bit timer register starts operation at the moment a value other than 0, 0, 0 (operation stop mode) is set to TMC01 to TMC03, respectively.**

8.6 16-Bit Timer/Event Counter Operating Cautions

(1) Timer start errors

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

Figure 8-35. 16-bit Timer Register Start Timing



(2) 16-bit compare register setting

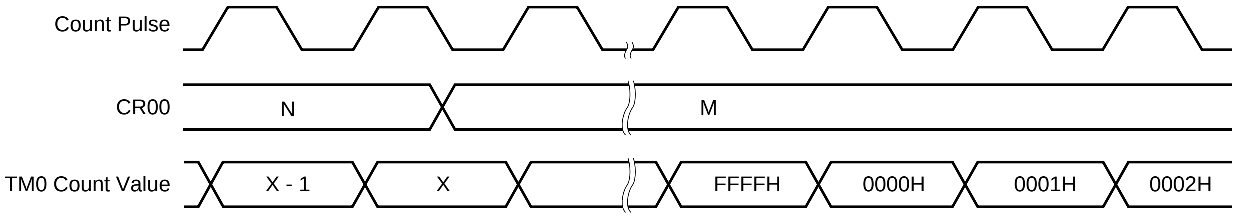
Set a value other than 0000H to the 16-bit capture/compare register 00 (CR00).

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

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

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

Figure 8-36. Timings After Change of Compare Register During Timer Count Operation

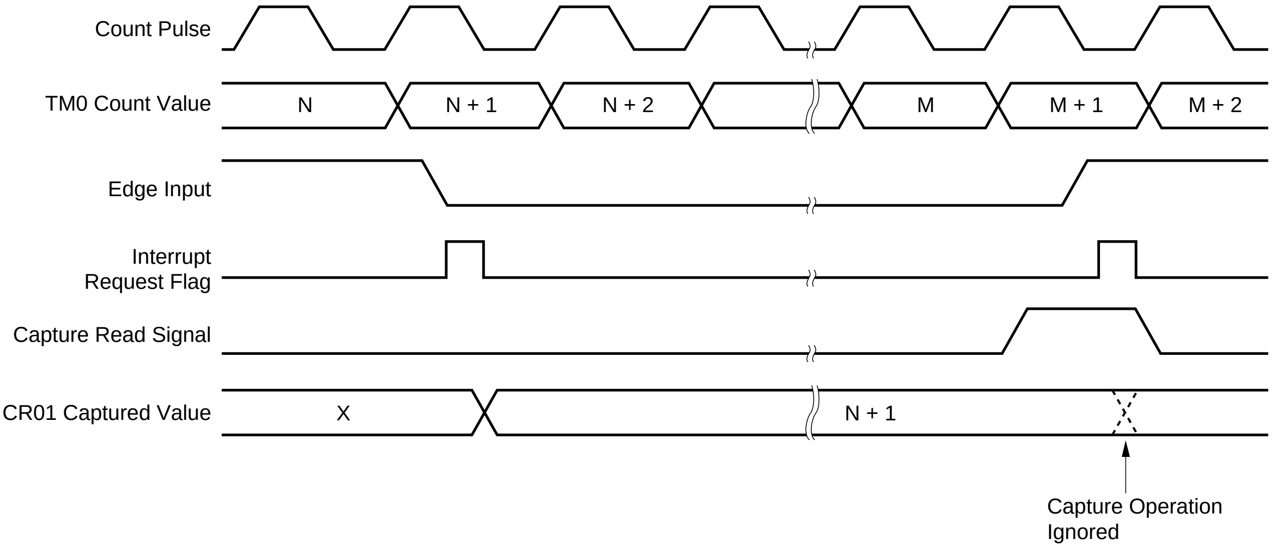


Remark N > X > M

(4) Capture register data retention timings

If the valid edge of the TI00/P00 pin is input during 16-bit capture/compare register 01 (CR01) read, CR01 holds data without carrying out capture operation. However, the interrupt request flag (PIF0) is set upon detection of the valid edge.

Figure 8-37. Capture Register Data Retention Timing



(5) Valid edge setting

Set the valid edge of the TI00/P00/INTP0 pin after setting bits 1 to 3 (TMC01 to TMC03) of the 16-bit timer mode control register (TMC0) to 0, 0 and 0, respectively, and then stopping timer operation. Valid edge setting is carried out with bits 2 and 3 (ES10 and ES11) of the external interrupt mode register 0 (INTM0).

(6) Re-trigger of one-shot pulse

(a) One-shot pulse output using software

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

(b) One-shot pulse output using external trigger

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

(7) Operation of OVF0 flag

OFV0 flag is set to 1 in the following case.

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

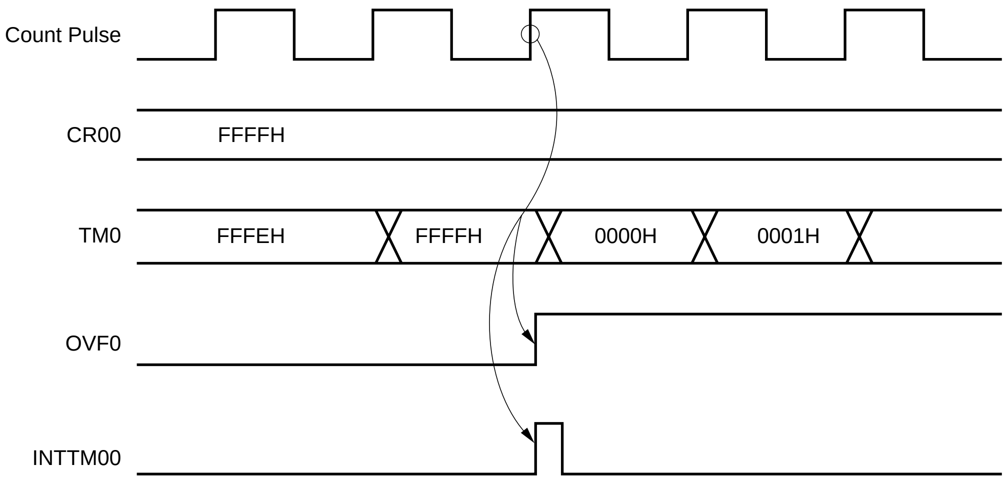
↓

CR00 is set to FFFFH.

↓

When TM0 is counted up from FFFFH to 0000H.

Figure 8-38. Operation Timing of OVF0 Flag



CHAPTER 9 8-BIT TIMER/EVENT COUNTER

9.1 8-Bit Timer/Event Counter Functions

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

9.1.1 8-bit timer/event counter mode

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

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

(1) 8-bit interval timer

Interrupt requests are generated at the preset time intervals.

Table 9-1. 8-bit Timer/Event Counters 1 and 2 Interval Times

Minimum Interval Time		Maximum Interval Time		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
$2 \times 1/f_X$ (400 ns)	$2^2 \times 1/f_X$ (800 ns)	$2^9 \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2 \times 1/f_X$ (400 ns)	$2^2 \times 1/f_X$ (800 ns)
$2^2 \times 1/f_X$ (800 ns)	$2^3 \times 1/f_X$ (1.6 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^2 \times 1/f_X$ (800 ns)	$2^3 \times 1/f_X$ (1.6 μs)
$2^3 \times 1/f_X$ (1.6 μs)	$2^4 \times 1/f_X$ (3.2 μs)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^3 \times 1/f_X$ (1.6 μs)	$2^4 \times 1/f_X$ (3.2 μs)
$2^4 \times 1/f_X$ (3.2 μs)	$2^5 \times 1/f_X$ (6.4 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{13} \times 1/f_X$ (1.64 ms)	$2^4 \times 1/f_X$ (3.2 μs)	$2^5 \times 1/f_X$ (6.4 μs)
$2^5 \times 1/f_X$ (6.4 μs)	$2^6 \times 1/f_X$ (12.8 μs)	$2^{13} \times 1/f_X$ (1.64 ms)	$2^{14} \times 1/f_X$ (3.28 ms)	$2^5 \times 1/f_X$ (6.4 μs)	$2^6 \times 1/f_X$ (12.8 μs)
$2^6 \times 1/f_X$ (12.8 μs)	$2^7 \times 1/f_X$ (25.6 μs)	$2^{14} \times 1/f_X$ (3.28 ms)	$2^{15} \times 1/f_X$ (6.55 ms)	$2^6 \times 1/f_X$ (12.8 μs)	$2^7 \times 1/f_X$ (25.6 μs)
$2^7 \times 1/f_X$ (25.6 μs)	$2^8 \times 1/f_X$ (51.2 μs)	$2^{15} \times 1/f_X$ (6.55 ms)	$2^{16} \times 1/f_X$ (13.1 ms)	$2^7 \times 1/f_X$ (25.6 μs)	$2^8 \times 1/f_X$ (51.2 μs)
$2^8 \times 1/f_X$ (51.2 μs)	$2^9 \times 1/f_X$ (102.4 μs)	$2^{16} \times 1/f_X$ (13.1 ms)	$2^{17} \times 1/f_X$ (26.2 ms)	$2^8 \times 1/f_X$ (51.2 μs)	$2^9 \times 1/f_X$ (102.4 μs)
$2^9 \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2^{17} \times 1/f_X$ (26.2 ms)	$2^{18} \times 1/f_X$ (52.4 ms)	$2^9 \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)
$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{19} \times 1/f_X$ (104.9 ms)	$2^{20} \times 1/f_X$ (209.7 ms)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)

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

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

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

(2) External event counter

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

(3) Square-wave output

A square wave with any selected frequency can be output.

Table 9-2. 8-bit Timer/Event Counters 1 and 2 Square-Wave Output Ranges

Minimum Pulse Time		Maximum Pulse Time		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
$2 \times 1/f_X$ (400 ns)	$2^{2} \times 1/f_X$ (800 ns)	$2^{9} \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2 \times 1/f_X$ (400 ns)	$2^{2} \times 1/f_X$ (800 ns)
$2^{2} \times 1/f_X$ (800 ns)	$2^{3} \times 1/f_X$ (1.6 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{2} \times 1/f_X$ (800 ns)	$2^{3} \times 1/f_X$ (1.6 μs)
$2^{3} \times 1/f_X$ (1.6 μs)	$2^{4} \times 1/f_X$ (3.2 μs)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{3} \times 1/f_X$ (1.6 μs)	$2^{4} \times 1/f_X$ (3.2 μs)
$2^{4} \times 1/f_X$ (3.2 μs)	$2^{5} \times 1/f_X$ (6.4 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{13} \times 1/f_X$ (1.64 ms)	$2^{4} \times 1/f_X$ (3.2 μs)	$2^{5} \times 1/f_X$ (6.4 μs)
$2^{5} \times 1/f_X$ (6.4 μs)	$2^{6} \times 1/f_X$ (12.8 μs)	$2^{13} \times 1/f_X$ (1.64 ms)	$2^{14} \times 1/f_X$ (3.28 ms)	$2^{5} \times 1/f_X$ (6.4 μs)	$2^{6} \times 1/f_X$ (12.8 μs)
$2^{6} \times 1/f_X$ (12.8 μs)	$2^{7} \times 1/f_X$ (25.6 μs)	$2^{14} \times 1/f_X$ (3.28 ms)	$2^{15} \times 1/f_X$ (6.55 ms)	$2^{6} \times 1/f_X$ (12.8 μs)	$2^{7} \times 1/f_X$ (25.6 μs)
$2^{7} \times 1/f_X$ (25.6 μs)	$2^{8} \times 1/f_X$ (51.2 μs)	$2^{15} \times 1/f_X$ (6.55 ms)	$2^{16} \times 1/f_X$ (13.1 ms)	$2^{7} \times 1/f_X$ (25.6 μs)	$2^{8} \times 1/f_X$ (51.2 μs)
$2^{8} \times 1/f_X$ (51.2 μs)	$2^{9} \times 1/f_X$ (102.4 μs)	$2^{16} \times 1/f_X$ (13.1 ms)	$2^{17} \times 1/f_X$ (26.2 ms)	$2^{8} \times 1/f_X$ (51.2 μs)	$2^{9} \times 1/f_X$ (102.4 μs)
$2^{9} \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2^{17} \times 1/f_X$ (26.2 ms)	$2^{18} \times 1/f_X$ (52.4 ms)	$2^{9} \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)
$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{19} \times 1/f_X$ (104.9 ms)	$2^{20} \times 1/f_X$ (209.7 ms)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)

Remarks 1. f_X : Main system clock oscillation frequency

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

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

9.1.2 16-bit timer/event counter mode

(1) 16-bit interval timer

Interrupt requests can be generated at the preset time intervals.

Table 9-3. Interval Times When 8-bit Timer/Event Counters 1 and 2 Are Used as 16-bit Timer/Event Counters

Minimum Interval Time		Maximum Interval Time		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
$2 \times 1/f_X$ (400 ns)	$2^2 \times 1/f_X$ (800 ns)	$2^{17} \times 1/f_X$ (26.2 ms)	$2^{18} \times 1/f_X$ (52.4 ms)	$2 \times 1/f_X$ (400 ns)	$2^2 \times 1/f_X$ (800 ns)
$2^2 \times 1/f_X$ (800 ns)	$2^3 \times 1/f_X$ (1.6 μs)	$2^{18} \times 1/f_X$ (52.4 ms)	$2^{19} \times 1/f_X$ (104.9 ms)	$2^2 \times 1/f_X$ (800 ns)	$2^3 \times 1/f_X$ (1.6 μs)
$2^3 \times 1/f_X$ (1.6 μs)	$2^4 \times 1/f_X$ (3.2 μs)	$2^{19} \times 1/f_X$ (104.9 ms)	$2^{20} \times 1/f_X$ (209.7 ms)	$2^3 \times 1/f_X$ (1.6 μs)	$2^4 \times 1/f_X$ (3.2 μs)
$2^4 \times 1/f_X$ (3.2 μs)	$2^5 \times 1/f_X$ (6.4 μs)	$2^{20} \times 1/f_X$ (209.7 ms)	$2^{21} \times 1/f_X$ (419.4 ms)	$2^4 \times 1/f_X$ (3.2 μs)	$2^5 \times 1/f_X$ (6.4 μs)
$2^5 \times 1/f_X$ (6.4 μs)	$2^6 \times 1/f_X$ (12.8 μs)	$2^{21} \times 1/f_X$ (419.4 ms)	$2^{22} \times 1/f_X$ (838.9 ms)	$2^5 \times 1/f_X$ (6.4 μs)	$2^6 \times 1/f_X$ (12.8 μs)
$2^6 \times 1/f_X$ (12.8 μs)	$2^7 \times 1/f_X$ (25.6 μs)	$2^{22} \times 1/f_X$ (838.9 ms)	$2^{23} \times 1/f_X$ (1.7 s)	$2^6 \times 1/f_X$ (12.8 μs)	$2^7 \times 1/f_X$ (25.6 μs)
$2^7 \times 1/f_X$ (25.6 μs)	$2^8 \times 1/f_X$ (51.2 μs)	$2^{23} \times 1/f_X$ (1.7 s)	$2^{24} \times 1/f_X$ (3.4 s)	$2^7 \times 1/f_X$ (25.6 μs)	$2^8 \times 1/f_X$ (51.2 μs)
$2^8 \times 1/f_X$ (51.2 μs)	$2^9 \times 1/f_X$ (102.4 μs)	$2^{24} \times 1/f_X$ (3.4 s)	$2^{25} \times 1/f_X$ (6.7 s)	$2^8 \times 1/f_X$ (51.2 μs)	$2^9 \times 1/f_X$ (102.4 μs)
$2^9 \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2^{25} \times 1/f_X$ (6.7 s)	$2^{26} \times 1/f_X$ (13.4 s)	$2^9 \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)
$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{27} \times 1/f_X$ (26.8 s)	$2^{28} \times 1/f_X$ (53.7 s)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)

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

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

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

(2) External event counter

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

(3) Square-wave output

A square wave with any selected frequency can be output.

Table 9-4. Square-Wave Output Ranges When 8-bit Timer/Event Counters 1 and 2 Are Used as 16-bit Timer/Event Counters

Minimum Pulse Time		Maximum Pulse Time		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
$2 \times 1/f_X$ (400 ns)	$2^2 \times 1/f_X$ (800 ns)	$2^{17} \times 1/f_X$ (26.2 ms)	$2^{18} \times 1/f_X$ (52.4 ms)	$2 \times 1/f_X$ (400 ns)	$2^2 \times 1/f_X$ (800 ns)
$2^2 \times 1/f_X$ (800 ns)	$2^3 \times 1/f_X$ (1.6 μs)	$2^{18} \times 1/f_X$ (52.4 ms)	$2^{19} \times 1/f_X$ (104.9 ms)	$2^2 \times 1/f_X$ (800 ns)	$2^3 \times 1/f_X$ (1.6 μs)
$2^3 \times 1/f_X$ (1.6 μs)	$2^4 \times 1/f_X$ (3.2 μs)	$2^{19} \times 1/f_X$ (104.9 ms)	$2^{20} \times 1/f_X$ (209.7 ms)	$2^3 \times 1/f_X$ (1.6 μs)	$2^4 \times 1/f_X$ (3.2 μs)
$2^4 \times 1/f_X$ (3.2 μs)	$2^5 \times 1/f_X$ (6.4 μs)	$2^{20} \times 1/f_X$ (209.7 ms)	$2^{21} \times 1/f_X$ (419.4 ms)	$2^4 \times 1/f_X$ (3.2 μs)	$2^5 \times 1/f_X$ (6.4 μs)
$2^5 \times 1/f_X$ (6.4 μs)	$2^6 \times 1/f_X$ (12.8 μs)	$2^{21} \times 1/f_X$ (419.4 ms)	$2^{22} \times 1/f_X$ (838.9 ms)	$2^5 \times 1/f_X$ (6.4 μs)	$2^6 \times 1/f_X$ (12.8 μs)
$2^6 \times 1/f_X$ (12.8 μs)	$2^7 \times 1/f_X$ (25.6 μs)	$2^{22} \times 1/f_X$ (838.9 ms)	$2^{23} \times 1/f_X$ (1.7 s)	$2^6 \times 1/f_X$ (12.8 μs)	$2^7 \times 1/f_X$ (25.6 μs)
$2^7 \times 1/f_X$ (25.6 μs)	$2^8 \times 1/f_X$ (51.2 μs)	$2^{23} \times 1/f_X$ (1.7 s)	$2^{24} \times 1/f_X$ (3.4 s)	$2^7 \times 1/f_X$ (25.6 μs)	$2^8 \times 1/f_X$ (51.2 μs)
$2^8 \times 1/f_X$ (51.2 μs)	$2^9 \times 1/f_X$ (102.4 μs)	$2^{24} \times 1/f_X$ (3.4 s)	$2^{25} \times 1/f_X$ (6.7 s)	$2^8 \times 1/f_X$ (51.2 μs)	$2^9 \times 1/f_X$ (102.4 μs)
$2^9 \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2^{25} \times 1/f_X$ (6.7 s)	$2^{26} \times 1/f_X$ (13.4 s)	$2^9 \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)
$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{27} \times 1/f_X$ (26.8 s)	$2^{28} \times 1/f_X$ (53.7 s)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)

Remarks 1. f_X : Main system clock oscillation frequency

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

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

9.2 8-Bit Timer/Event Counter Configurations

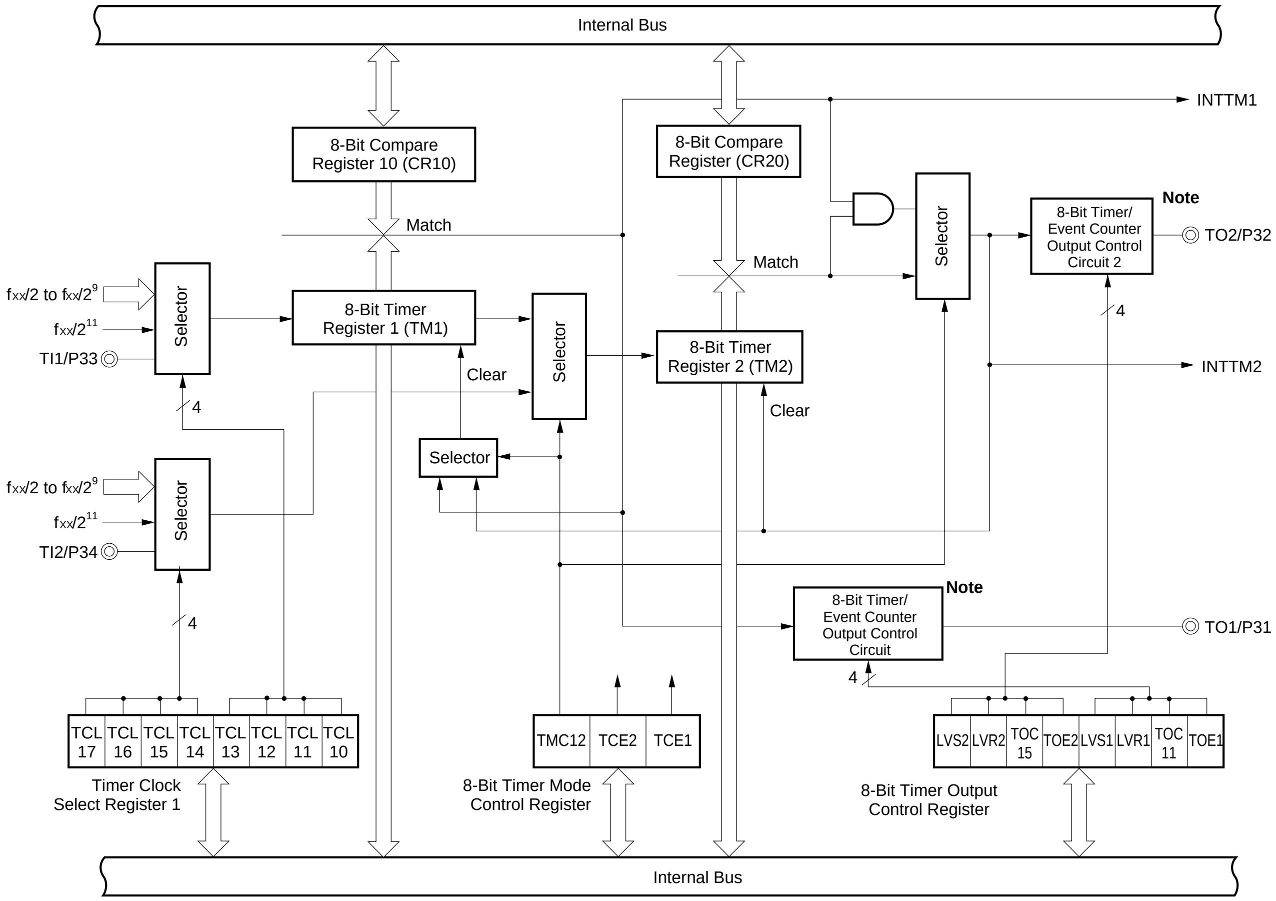
The 8-bit timer/event counter consists of the following hardware.

Table 9-5. 8-bit Timer/Event Counter Configurations

Item	Configuration
Timer register	8 bits × 2 (TM1, TM2)
Register	Compare register: 8 bits × 2 (CR10, CR20)
Timer output	2 (TO1, TO2)
Control register	Timer clock select register 1 (TCL1) 8-bit timer mode control register 1 (TMC1) 8-bit timer output control register (TOC1) Port mode register 3 (PM3) **Note**

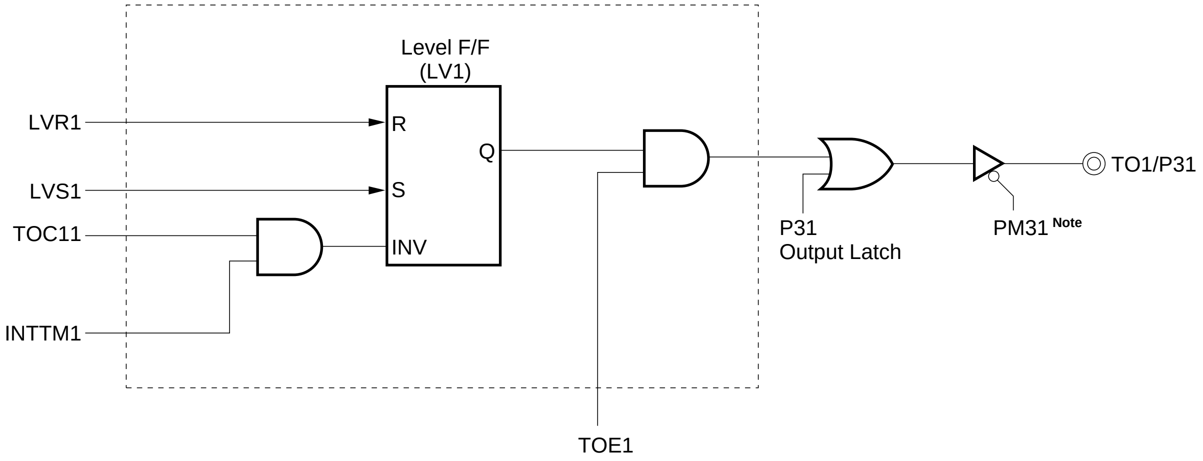
Note Refer to **Figure 6-9 P30 to P37 Block Diagram**.

Figure 9-1. 8-bit Timer/Event Counter Block Diagram


Internal Bus
INTTM1
8-Bit Compare Register 10 (CR10)
8-Bit Compare Register (CR20)
Note
8-Bit Timer/ Event Counter Output Control Circuit 2
TO2/P32
Match
Match
Selector
fxx/2 to fxx/2^9
fxx/2^11
TI1/P33
Selector
8-Bit Timer Register 1 (TM1)
Selector
8-Bit Timer Register 2 (TM2)
4
INTTM2
Clear
Clear
4
Selector
fxx/2 to fxx/2^9
fxx/2^11
TI2/P34
Selector
4
Note
8-Bit Timer/ Event Counter Output Control Circuit
TO1/P31
4
TCL 17
TCL 16
TCL 15
TCL 14
TCL 13
TCL 12
TCL 11
TCL 10
TMC12
TCE2
TCE1
LVS2
LVR2
TOC 15
TOE2
LVS1
LVR1
TOC 11
TOE1
Timer Clock Select Register 1
8-Bit Timer Mode Control Register
8-Bit Timer Output Control Register
Internal Bus

Note Refer to Figures 9-2 and 9-3 for details of 8-bit timer/event counter output control circuits 1 and 2, respectively.

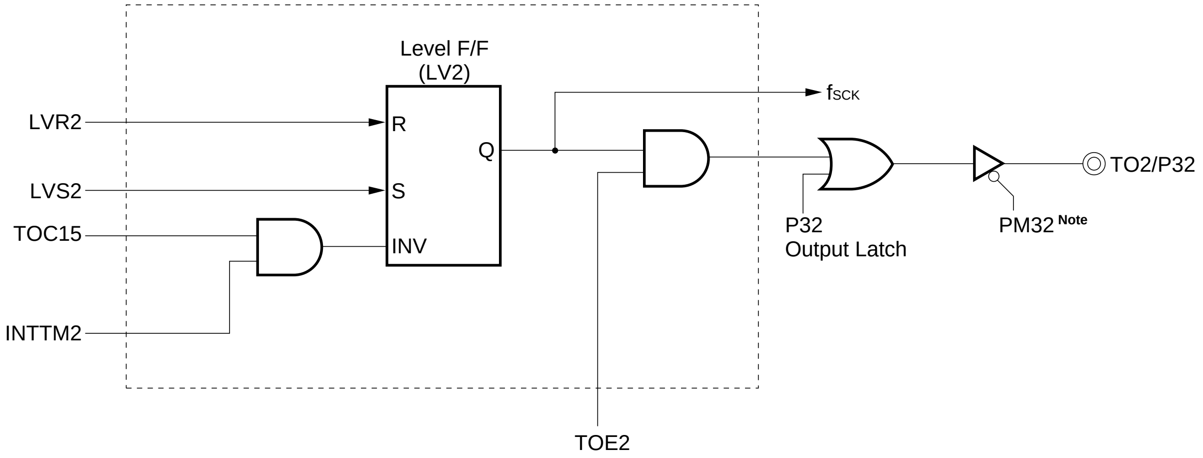
Figure 9-2. Block Diagram of 8-bit Timer/Event Counter Output Control Circuit 1



Note Bit 1 of port mode register 3 (PM3)

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

Figure 9-3. Block Diagram of 8-bit Timer/Event Counter Output Control Circuit 2



Note Bit 2 of port mode register 3 (PM3)

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

2. f_{SCK} : Serial clock frequency

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

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

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

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

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

2. When the new values of CR10 and CR20 are less than the count values of the 8-bit timer registers (TM1 and TM2), TM1 and TM2 continue counting, overflow, and start counting again from 0. If the new values of CR10 and CR20 are less than the old values, therefore, it is necessary to restart the timers after changing the values of CR10 and CR20.

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

These are 8-bit registers to count count pulses.

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

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

9.3 8-Bit Timer/Event Counter Control Registers

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

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

(1) Timer clock select register 1 (TCL1)

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

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

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

Figure 9-4. Timer Clock Select Register 1 Format

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

TCL13	TCL12	TCL11	TCL10	8-Bit Timer Register 1 Count Clock Selection		
					MCS = 1	MCS = 0
0	0	0	0	TI1 falling edge		
0	0	0	1	TI1 rising edge		
0	1	1	0	$f_{XX}/2$	$f_X/2$ (2.5 MHz)	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
1	1	1	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)
1	1	1	1	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)
Other than above				Setting prohibited		

TCL17	TCL16	TCL15	TCL14	8-Bit Timer Register 2 Count Clock Selection		
					MCS = 1	MCS = 0
0	0	0	0	TI2 falling edge		
0	0	0	1	TI2 rising edge		
0	1	1	0	$f_{XX}/2$	$f_X/2$ (2.5 MHz)	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
1	1	1	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)
1	1	1	1	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)
Other than above				Setting prohibited		

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

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

2. f_X : Main system clock oscillation frequency

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

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

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

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

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

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

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

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

Figure 9-5. 8-bit Timer Mode Control Register 1 Format

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

TCE1	8-Bit Timer Register 1 Operation Control
0	Operation stopped (TM1 is cleared to 0)
1	Operation enabled

TCE2	8-Bit Timer Register 2 Operation Control
0	Operation stop (TM2 is cleared to 0)
1	Operation enabled

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

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

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

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

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

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

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

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

Figure 9-6. 8-bit Timer Output Control Register Format

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

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

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

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

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

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

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

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

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

(4) Port mode register 3 (PM3)

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

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

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

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

Figure 9-7. Port Mode Register 3 Format

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

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

9.4 8-Bit Timer/Event Counters 1 and 2 Operations

9.4.1 8-bit timer/event counter mode

(1) Interval timer operations

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

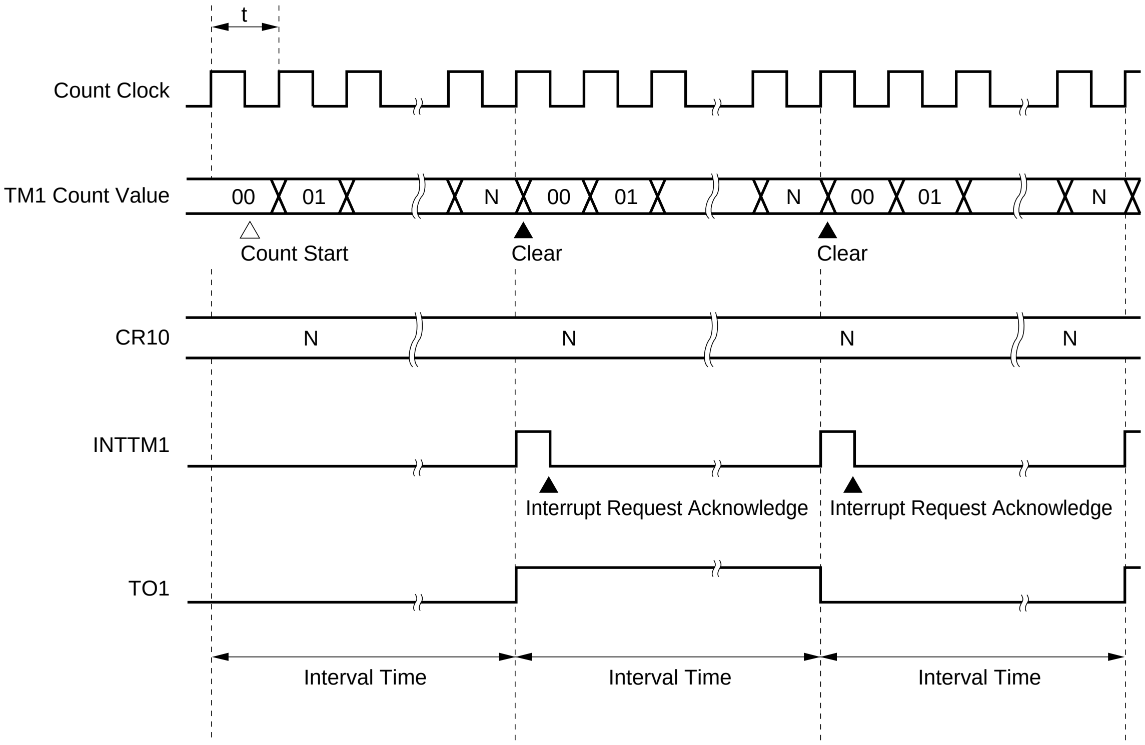
When the count values of the 8-bit timer registers 1 and 2 (TM1 and TM2) match the values set to CR10 and CR20, counting continues with the TM1 and TM2 values cleared to 0 and the interrupt request signals (INTTM1 and INTTM2) are generated.

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

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

For the operation when the value of the compare register is changed during a timer count operation, refer to **9.5 (3) Operation after compare register change during timer count operation**.

Figure 9-8. Interval Timer Operation Timings



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

Table 9-6. 8-bit Timer/Event Counter 1 Interval Time

TCL13	TCL12	TCL11	TCL10	Minimum Interval Time		Maximum Interval Time		Resolution	
				MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
0	0	0	0	TI1 input cycle		2^8 × TI1 input cycle		TI1 input edge cycle	
0	0	0	1	TI1 input cycle		2^8 × TI1 input cycle		TI1 input edge cycle	
0	1	1	0	$2 \times 1/f_X$ (400 ns)	$2^2 \times 1/f_X$ (800 ns)	$2^9 \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2 \times 1/f_X$ (400 ns)	$2^2 \times 1/f_X$ (800 ns)
0	1	1	1	$2^2 \times 1/f_X$ (800 ns)	$2^3 \times 1/f_X$ (1.6 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^2 \times 1/f_X$ (800 ns)	$2^3 \times 1/f_X$ (1.6 μs)
1	0	0	0	$2^3 \times 1/f_X$ (1.6 μs)	$2^4 \times 1/f_X$ (3.2 μs)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^3 \times 1/f_X$ (1.6 μs)	$2^4 \times 1/f_X$ (3.2 μs)
1	0	0	1	$2^4 \times 1/f_X$ (3.2 μs)	$2^5 \times 1/f_X$ (6.4 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{13} \times 1/f_X$ (1.64 ms)	$2^4 \times 1/f_X$ (3.2 μs)	$2^5 \times 1/f_X$ (6.4 μs)
1	0	1	0	$2^5 \times 1/f_X$ (6.4 μs)	$2^6 \times 1/f_X$ (12.8 μs)	$2^{13} \times 1/f_X$ (1.64 ms)	$2^{14} \times 1/f_X$ (3.28 ms)	$2^5 \times 1/f_X$ (6.4 μs)	$2^6 \times 1/f_X$ (12.8 μs)
1	0	1	1	$2^6 \times 1/f_X$ (12.8 μs)	$2^7 \times 1/f_X$ (25.6 μs)	$2^{14} \times 1/f_X$ (3.28 ms)	$2^{15} \times 1/f_X$ (6.55 ms)	$2^6 \times 1/f_X$ (12.8 μs)	$2^7 \times 1/f_X$ (25.6 μs)
1	1	0	0	$2^7 \times 1/f_X$ (25.6 μs)	$2^8 \times 1/f_X$ (51.2 μs)	$2^{15} \times 1/f_X$ (6.55 ms)	$2^{16} \times 1/f_X$ (13.1 ms)	$2^7 \times 1/f_X$ (25.6 μs)	$2^8 \times 1/f_X$ (51.2 μs)
1	1	0	1	$2^8 \times 1/f_X$ (51.2 μs)	$2^9 \times 1/f_X$ (102.4 μs)	$2^{16} \times 1/f_X$ (13.1 ms)	$2^{17} \times 1/f_X$ (26.2 ms)	$2^8 \times 1/f_X$ (51.2 μs)	$2^9 \times 1/f_X$ (102.4 μs)
1	1	1	0	$2^9 \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2^{17} \times 1/f_X$ (26.2 ms)	$2^{18} \times 1/f_X$ (52.4 ms)	$2^9 \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)
1	1	1	1	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{19} \times 1/f_X$ (104.9 ms)	$2^{20} \times 1/f_X$ (209.7 ms)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)
Other than above				Setting prohibited					

Remarks 1. f_X : Main system clock oscillation frequency

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

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

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

Table 9-7. 8-bit Timer/Event Counter 2 Interval Time

TCL17	TCL16	TCL15	TCL14	Minimum Interval Time		Maximum Interval Time		Resolution	
				MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
0	0	0	0	TI2 input cycle		$2^{8} \times$ TI2 input cycle		TI2 input edge cycle	
0	0	0	1	TI2 input cycle		$2^{8} \times$ TI2 input cycle		TI2 input edge cycle	
0	1	1	0	$2 \times 1/f_X$ (400 ns)	$2^{2} \times 1/f_X$ (800 ns)	$2^{9} \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2 \times 1/f_X$ (400 ns)	$2^{2} \times 1/f_X$ (800 ns)
0	1	1	1	$2^{2} \times 1/f_X$ (800 ns)	$2^{3} \times 1/f_X$ (1.6 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{2} \times 1/f_X$ (800 ns)	$2^{3} \times 1/f_X$ (1.6 μs)
1	0	0	0	$2^{3} \times 1/f_X$ (1.6 μs)	$2^{4} \times 1/f_X$ (3.2 μs)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{3} \times 1/f_X$ (1.6 μs)	$2^{4} \times 1/f_X$ (3.2 μs)
1	0	0	1	$2^{4} \times 1/f_X$ (3.2 μs)	$2^{5} \times 1/f_X$ (6.4 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{13} \times 1/f_X$ (1.64 ms)	$2^{4} \times 1/f_X$ (3.2 μs)	$2^{5} \times 1/f_X$ (6.4 μs)
1	0	1	0	$2^{5} \times 1/f_X$ (6.4 μs)	$2^{6} \times 1/f_X$ (12.8 μs)	$2^{13} \times 1/f_X$ (1.64 ms)	$2^{14} \times 1/f_X$ (3.28 ms)	$2^{5} \times 1/f_X$ (6.4 μs)	$2^{6} \times 1/f_X$ (12.8 μs)
1	0	1	1	$2^{6} \times 1/f_X$ (12.8 μs)	$2^{7} \times 1/f_X$ (25.6 μs)	$2^{14} \times 1/f_X$ (3.28 ms)	$2^{15} \times 1/f_X$ (6.55 ms)	$2^{6} \times 1/f_X$ (12.8 μs)	$2^{7} \times 1/f_X$ (25.6 μs)
1	1	0	0	$2^{7} \times 1/f_X$ (25.6 μs)	$2^{8} \times 1/f_X$ (51.2 μs)	$2^{15} \times 1/f_X$ (6.55 ms)	$2^{16} \times 1/f_X$ (13.1 ms)	$2^{7} \times 1/f_X$ (25.6 μs)	$2^{8} \times 1/f_X$ (51.2 μs)
1	1	0	1	$2^{8} \times 1/f_X$ (51.2 μs)	$2^{9} \times 1/f_X$ (102.4 μs)	$2^{16} \times 1/f_X$ (13.1 ms)	$2^{17} \times 1/f_X$ (26.2 ms)	$2^{8} \times 1/f_X$ (51.2 μs)	$2^{9} \times 1/f_X$ (102.4 μs)
1	1	1	0	$2^{9} \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2^{17} \times 1/f_X$ (26.2 ms)	$2^{18} \times 1/f_X$ (52.4 ms)	$2^{9} \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)
1	1	1	1	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{19} \times 1/f_X$ (104.9 ms)	$2^{20} \times 1/f_X$ (209.7 ms)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)
Other than above				Setting prohibited					

Remarks 1. f_X : Main system clock oscillation frequency

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

3. TCL14 to TCL17 : Bits 4 to 7 of timer clock select register 1 (TCL1)

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

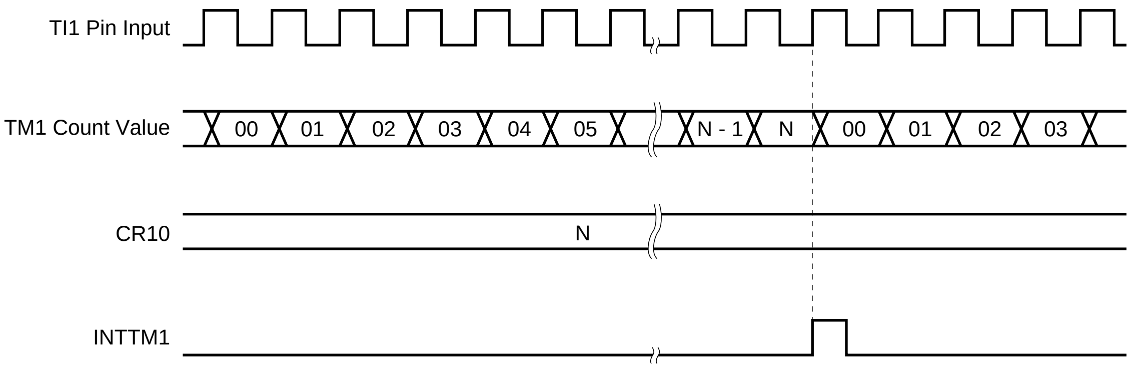
(2) External event counter operation

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

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

When the TM1 and TM2 counted values match the values of 8-bit compare registers 10, 20 (CR10 and CR20), TM1 and TM2 are cleared to 0 and the interrupt request signals (INTTM1 and INTTM2) are generated.

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



Remark N = 00H to FFH

(3) Square-wave output operation

The 8-bit timer/event counters 1 and 2 output a square wave with any selected frequency at intervals specified by the value set in advance to 8-bit compare registers 10 and 20 (CR10 and CR20).

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

Table 9-8. 8-bit Timer/Event Counters 1 and 2 Square-Wave Output Ranges

Minimum Pulse Time		Maximum Pulse Time		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
$2 \times 1/f_X$ (400 ns)	$2^{2} \times 1/f_X$ (800 ns)	$2^{9} \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2 \times 1/f_X$ (400 ns)	$2^{2} \times 1/f_X$ (800 ns)
$2^{2} \times 1/f_X$ (800 ns)	$2^{3} \times 1/f_X$ (1.6 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{2} \times 1/f_X$ (800 ns)	$2^{3} \times 1/f_X$ (1.6 μs)
$2^{3} \times 1/f_X$ (1.6 μs)	$2^{4} \times 1/f_X$ (3.2 μs)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{3} \times 1/f_X$ (1.6 μs)	$2^{4} \times 1/f_X$ (3.2 μs)
$2^{4} \times 1/f_X$ (3.2 μs)	$2^{5} \times 1/f_X$ (6.4 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{13} \times 1/f_X$ (1.64 ms)	$2^{4} \times 1/f_X$ (3.2 μs)	$2^{5} \times 1/f_X$ (6.4 μs)
$2^{5} \times 1/f_X$ (6.4 μs)	$2^{6} \times 1/f_X$ (12.8 μs)	$2^{13} \times 1/f_X$ (1.64 ms)	$2^{14} \times 1/f_X$ (3.28 ms)	$2^{5} \times 1/f_X$ (6.4 μs)	$2^{6} \times 1/f_X$ (12.8 μs)
$2^{6} \times 1/f_X$ (12.8 μs)	$2^{7} \times 1/f_X$ (25.6 μs)	$2^{14} \times 1/f_X$ (3.28 ms)	$2^{15} \times 1/f_X$ (6.55 ms)	$2^{6} \times 1/f_X$ (12.8 μs)	$2^{7} \times 1/f_X$ (25.6 μs)
$2^{7} \times 1/f_X$ (25.6 μs)	$2^{8} \times 1/f_X$ (51.2 μs)	$2^{15} \times 1/f_X$ (6.55 ms)	$2^{16} \times 1/f_X$ (13.1 ms)	$2^{7} \times 1/f_X$ (25.6 μs)	$2^{8} \times 1/f_X$ (51.2 μs)
$2^{8} \times 1/f_X$ (51.2 μs)	$2^{9} \times 1/f_X$ (102.4 μs)	$2^{16} \times 1/f_X$ (13.1 ms)	$2^{17} \times 1/f_X$ (26.2 ms)	$2^{8} \times 1/f_X$ (51.2 μs)	$2^{9} \times 1/f_X$ (102.4 μs)
$2^{9} \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2^{17} \times 1/f_X$ (26.2 ms)	$2^{18} \times 1/f_X$ (52.4 ms)	$2^{9} \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)
$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{19} \times 1/f_X$ (104.9 ms)	$2^{20} \times 1/f_X$ (209.7 ms)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)

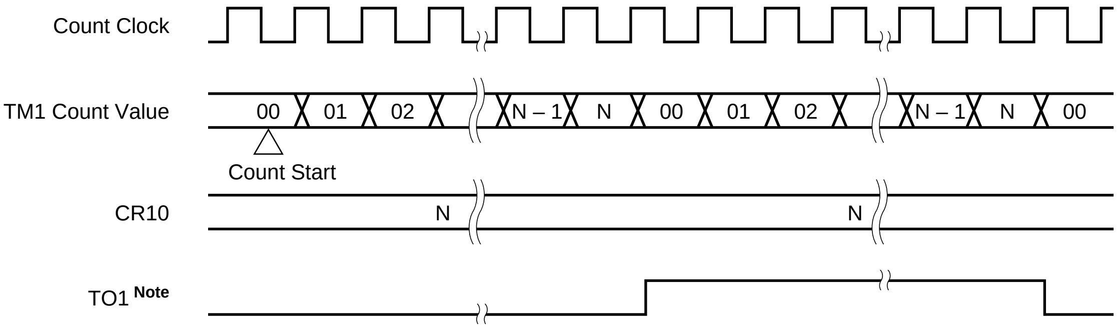
Remarks 1. f_X : Main system clock oscillation frequency

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

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

★

Figure 9-10. Square Wave Output Operation Timing



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

9.4.2 16-bit timer/event counter mode

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

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

The counting operation is enabled or disabled in this mode by using bit 0 (TCE1) of TMC1.

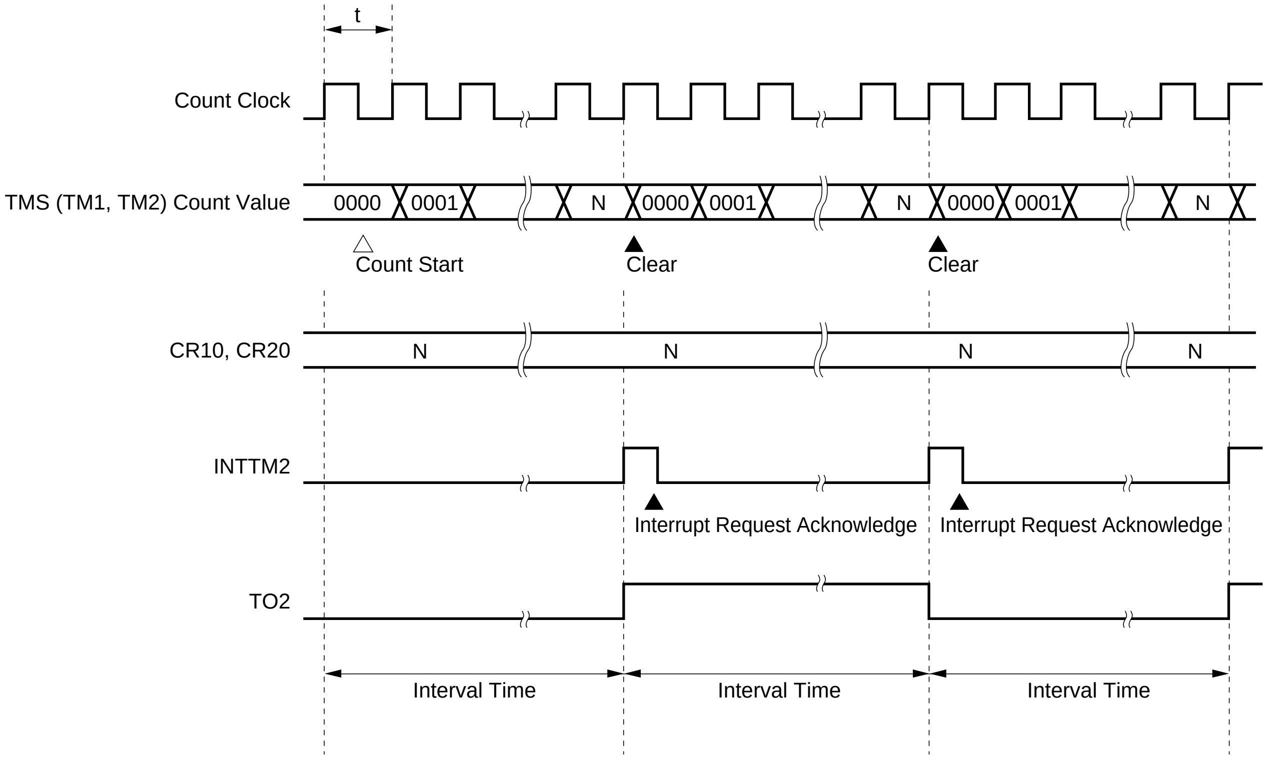
(1) Operation as interval timer

The 16-bit timer/event counter operates as an interval timer that repeatedly generates an interrupt request at intervals of the count values set in advance to the 2 channels of the 8-bit compare registers (CR10 and CR20). When setting a count value, assign the value of the high-order 8 bits to CR20 and the value of the low-order 8 bits to CR10. For the count values that can be set (interval time), refer to **Table 9-9**.

When the value of 8-bit timer register 1 (TM1) coincides with the value of CR10 and the value of 8-bit timer register 2 (TM1) coincides with the value of CR20, the values of TM1 and TM2 are cleared to 0, and at the same time, an interrupt request signal (INTTM2) is generated. For the operation timing of the interval timer, refer to **Figure 9-11**.

Select the count clock by using bits 0 through 3 (TCL10 through TCL13) of the timer clock select register 1 (TCL1). The overflow signal of TM1 is used as the count clock for TM2.

Figure 9-11. Interval Timer Operation Timing



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

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

Table 9-9. Interval Times When 2-channel 8-bit Timer/Event Counters (TM1 and TM2) Are Used as 16-bit Timer/Event Counter

TCL13	TCL12	TCL11	TCL10	Minimum Interval Time		Maximum Interval Time		Resolution	
				MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
0	0	0	0	TI1 input cycle		2^{8} × TI1 input cycle		TI1 input edge cycle	
0	0	0	1	TI1 input cycle		2^{8} × TI1 input cycle		TI1 input edge cycle	
0	1	1	0	$2 \times 1/f_X$ (400 ns)	$2^{2} \times 1/f_X$ (800 ns)	$2^{17} \times 1/f_X$ (26.2 ms)	$2^{18} \times 1/f_X$ (52.4 ms)	$2 \times 1/f_X$ (400 ns)	$2^{2} \times 1/f_X$ (800 ns)
0	1	1	1	$2^{2} \times 1/f_X$ (800 ns)	$2^{3} \times 1/f_X$ (1.6 μs)	$2^{18} \times 1/f_X$ (52.4 ms)	$2^{19} \times 1/f_X$ (104.9 ms)	$2^{2} \times 1/f_X$ (800 ns)	$2^{3} \times 1/f_X$ (1.6 μs)
1	0	0	0	$2^{3} \times 1/f_X$ (1.6 μs)	$2^{4} \times 1/f_X$ (3.2 μs)	$2^{19} \times 1/f_X$ (104.9 ms)	$2^{20} \times 1/f_X$ (209.7 ms)	$2^{3} \times 1/f_X$ (1.6 μs)	$2^{4} \times 1/f_X$ (3.2 μs)
1	0	0	1	$2^{4} \times 1/f_X$ (3.2 μs)	$2^{5} \times 1/f_X$ (6.4 μs)	$2^{20} \times 1/f_X$ (209.7 ms)	$2^{21} \times 1/f_X$ (419.4 ms)	$2^{4} \times 1/f_X$ (3.2 μs)	$2^{5} \times 1/f_X$ (6.4 μs)
1	0	1	0	$2^{5} \times 1/f_X$ (6.4 μs)	$2^{6} \times 1/f_X$ (12.8 μs)	$2^{21} \times 1/f_X$ (419.4 ms)	$2^{22} \times 1/f_X$ (838.9 ms)	$2^{5} \times 1/f_X$ (6.4 μs)	$2^{6} \times 1/f_X$ (12.8 μs)
1	0	1	1	$2^{6} \times 1/f_X$ (12.8 μs)	$2^{7} \times 1/f_X$ (25.6 μs)	$2^{22} \times 1/f_X$ (838.9 ms)	$2^{23} \times 1/f_X$ (1.7 s)	$2^{6} \times 1/f_X$ (12.8 μs)	$2^{7} \times 1/f_X$ (25.6 μs)
1	1	0	0	$2^{7} \times 1/f_X$ (25.6 μs)	$2^{8} \times 1/f_X$ (51.2 μs)	$2^{23} \times 1/f_X$ (1.7 s)	$2^{24} \times 1/f_X$ (3.4 s)	$2^{7} \times 1/f_X$ (25.6 μs)	$2^{8} \times 1/f_X$ (51.2 μs)
1	1	0	1	$2^{8} \times 1/f_X$ (51.2 μs)	$2^{9} \times 1/f_X$ (102.4 μs)	$2^{24} \times 1/f_X$ (3.4 s)	$2^{25} \times 1/f_X$ (6.7 s)	$2^{8} \times 1/f_X$ (51.2 μs)	$2^{9} \times 1/f_X$ (102.4 μs)
1	1	1	0	$2^{9} \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2^{25} \times 1/f_X$ (6.7 s)	$2^{26} \times 1/f_X$ (13.4 s)	$2^{9} \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)
1	1	1	1	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{27} \times 1/f_X$ (26.8 s)	$2^{28} \times 1/f_X$ (53.7 s)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)
Other than above				Setting prohibited					

Remarks 1. f_X : Main system clock oscillation frequency

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

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

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

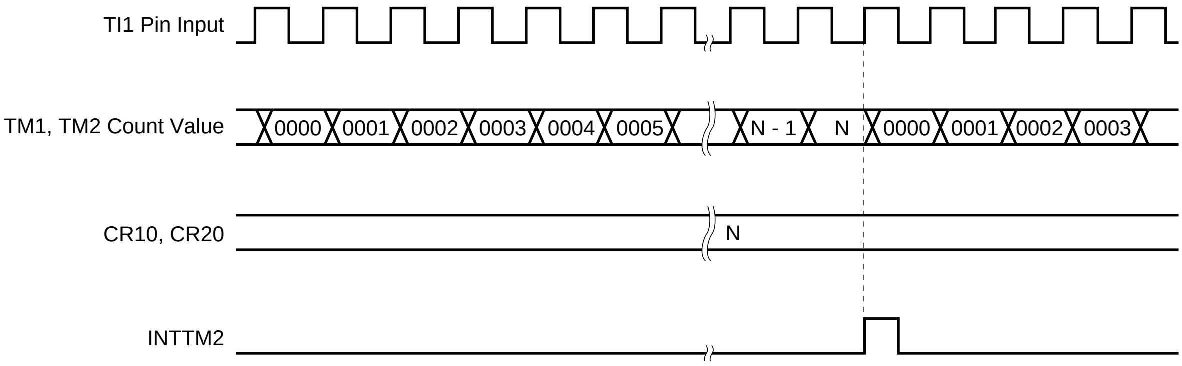
(2) External event counter operations

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

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

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

Figure 9-12. External Event Counter Operation Timings (with Rising Edge Specified)



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

(3) Square-wave output operation

The 8-bit timer/event counters 1 and 2 output a square wave with any selected frequency at intervals specified by the value set in advance to 8-bit compare registers 10 and 20 (CR10 and CR20). To set a count value, set the value of the high-order 8 bits to CR20, and the value of the low-order 8 bits to CR10.

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

Table 9-10. Square-Wave Output Ranges When 2-channel 8-bit Timer/Event Counters (TM1 and TM2) Are Used as 16-bit Timer/Event Counter

Minimum Pulse Time		Maximum Pulse Time		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
$2 \times 1/f_X$ (400 ns)	$2^{2} \times 1/f_X$ (800 ns)	$2^{17} \times 1/f_X$ (26.2 ms)	$2^{18} \times 1/f_X$ (52.4 ms)	$2 \times 1/f_X$ (400 ns)	$2^{2} \times 1/f_X$ (800 ns)
$2^{2} \times 1/f_X$ (800 ns)	$2^{3} \times 1/f_X$ (1.6 μs)	$2^{18} \times 1/f_X$ (52.4 ms)	$2^{19} \times 1/f_X$ (104.9 ms)	$2^{2} \times 1/f_X$ (800 ns)	$2^{3} \times 1/f_X$ (1.6 μs)
$2^{3} \times 1/f_X$ (1.6 μs)	$2^{4} \times 1/f_X$ (3.2 μs)	$2^{19} \times 1/f_X$ (104.9 ms)	$2^{20} \times 1/f_X$ (209.7 ms)	$2^{3} \times 1/f_X$ (1.6 μs)	$2^{4} \times 1/f_X$ (3.2 μs)
$2^{4} \times 1/f_X$ (3.2 μs)	$2^{5} \times 1/f_X$ (6.4 μs)	$2^{20} \times 1/f_X$ (209.7 ms)	$2^{21} \times 1/f_X$ (419.4 ms)	$2^{4} \times 1/f_X$ (3.2 μs)	$2^{5} \times 1/f_X$ (6.4 μs)
$2^{5} \times 1/f_X$ (6.4 μs)	$2^{6} \times 1/f_X$ (12.8 μs)	$2^{21} \times 1/f_X$ (419.4 ms)	$2^{22} \times 1/f_X$ (838.9 ms)	$2^{5} \times 1/f_X$ (6.4 μs)	$2^{6} \times 1/f_X$ (12.8 μs)
$2^{6} \times 1/f_X$ (12.8 μs)	$2^{7} \times 1/f_X$ (25.6 μs)	$2^{22} \times 1/f_X$ (838.9 ms)	$2^{23} \times 1/f_X$ (1.7 s)	$2^{6} \times 1/f_X$ (12.8 μs)	$2^{7} \times 1/f_X$ (25.6 μs)
$2^{7} \times 1/f_X$ (25.6 μs)	$2^{8} \times 1/f_X$ (51.2 μs)	$2^{23} \times 1/f_X$ (1.7 s)	$2^{24} \times 1/f_X$ (3.4 s)	$2^{7} \times 1/f_X$ (25.6 μs)	$2^{8} \times 1/f_X$ (51.2 μs)
$2^{8} \times 1/f_X$ (51.2 μs)	$2^{9} \times 1/f_X$ (102.4 μs)	$2^{24} \times 1/f_X$ (3.4 s)	$2^{25} \times 1/f_X$ (6.7 s)	$2^{8} \times 1/f_X$ (51.2 μs)	$2^{9} \times 1/f_X$ (102.4 μs)
$2^{9} \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2^{25} \times 1/f_X$ (6.7 s)	$2^{26} \times 1/f_X$ (13.4 s)	$2^{9} \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)
$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{27} \times 1/f_X$ (26.8 s)	$2^{28} \times 1/f_X$ (53.7 s)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^{12} \times 1/f_X$ (819.2 μs)

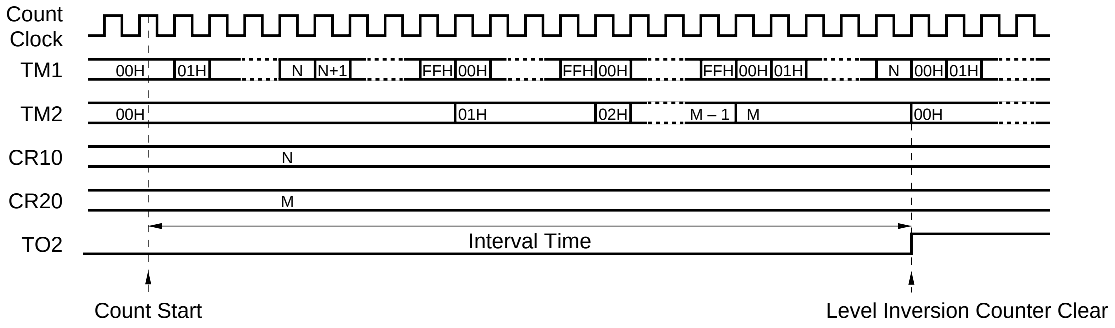
Remarks 1. f_X : Main system clock oscillation frequency

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

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

★

Figure 9-13. Square Wave Output Operation Timing

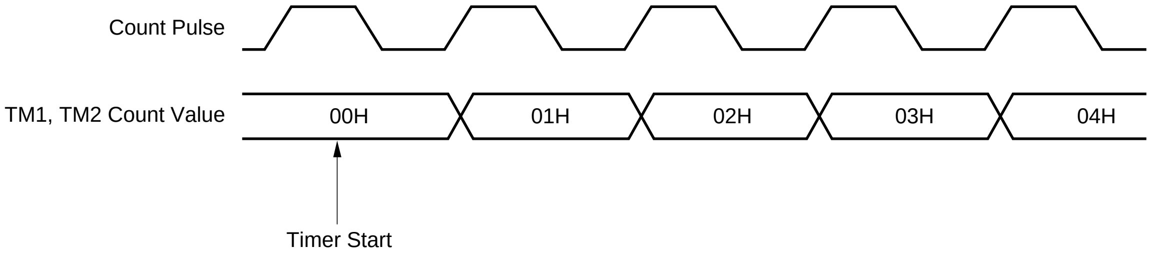


9.5 8-Bit Timer/Event Counters 1 and 2 Precautions

(1) Timer start errors

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

Figure 9-14. 8-bit Timer Registers 1 and 2 Start Timing



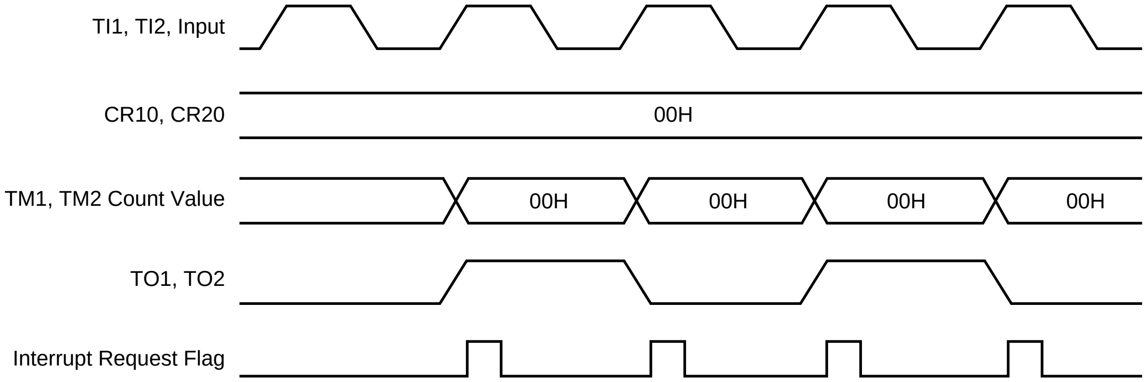
(2) 8-bit compare register 10 and 20 setting

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

Thus, when these 8-bit compare registers are used as event counters, one-pulse count operation can be carried out.

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

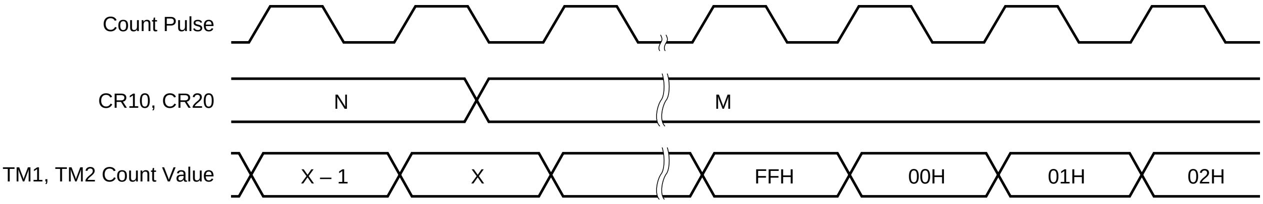
Figure 9-15. External Event Counter Operation Timing



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

If the values after the 8-bit compare registers 10 and 20 (CR10 and CR20) are changed are smaller than those of 8-bit timer registers (TM1 and TM2), TM1 and TM2 continue counting, overflow and then restart counting from 0. Thus, if the value (M) after CR10 and CR20 change is smaller than value (N) before the change, it is necessary to restart the timer after changing CR10 and CR20.

Figure 9-16. Timing After Compare Register Change During Timer Count Operation



Remark N > X > M

[MEMO]

CHAPTER 10 WATCH TIMER

10.1 Watch Timer Functions

The watch timer has the following functions.

- Watch timer
- Interval timer

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

(1) Watch timer

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

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

(2) Interval timer

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

Table 10-1. Interval Timer Interval Time

Interval Time	When operated at f_{XX} = 5.0 MHz	When operated at f_{XX} = 4.19 MHz	When operated at f_{XT} = 32.768 kHz
$2^4 \times 1/f_W$	410 μs	488 μs	488 μs
$2^5 \times 1/f_W$	819 μs	977 μs	977 μs
$2^6 \times 1/f_W$	1.64 ms	1.95 ms	1.95 ms
$2^7 \times 1/f_W$	3.28 ms	3.91 ms	3.91 ms
$2^8 \times 1/f_W$	6.55 ms	7.81 ms	7.81 ms
$2^9 \times 1/f_W$	13.1 ms	15.6 ms	15.6 ms

Remark f_{XX} : Main system clock frequency (f_X or $f_X/2$)
f_X : Main system clock oscillation frequency
f_{XT} : Subsystem clock oscillation frequency
f_W : Watch timer clock frequency ($f_{XX}/2^7$ or f_{XT})

10.2 Watch Timer Configuration

The watch timer consists of the following hardware.

Table 10-2. Watch Timer Configuration

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

10.3 Watch Timer Control Registers

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

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

(1) Timer clock select register 2 (TCL2) (refer to **Figure 10-2.**)
This register sets the watch timer count clock.
TCL2 is set with an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears TCL2 to 00H.

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

Figure 10-1. Watch Timer Block Diagram

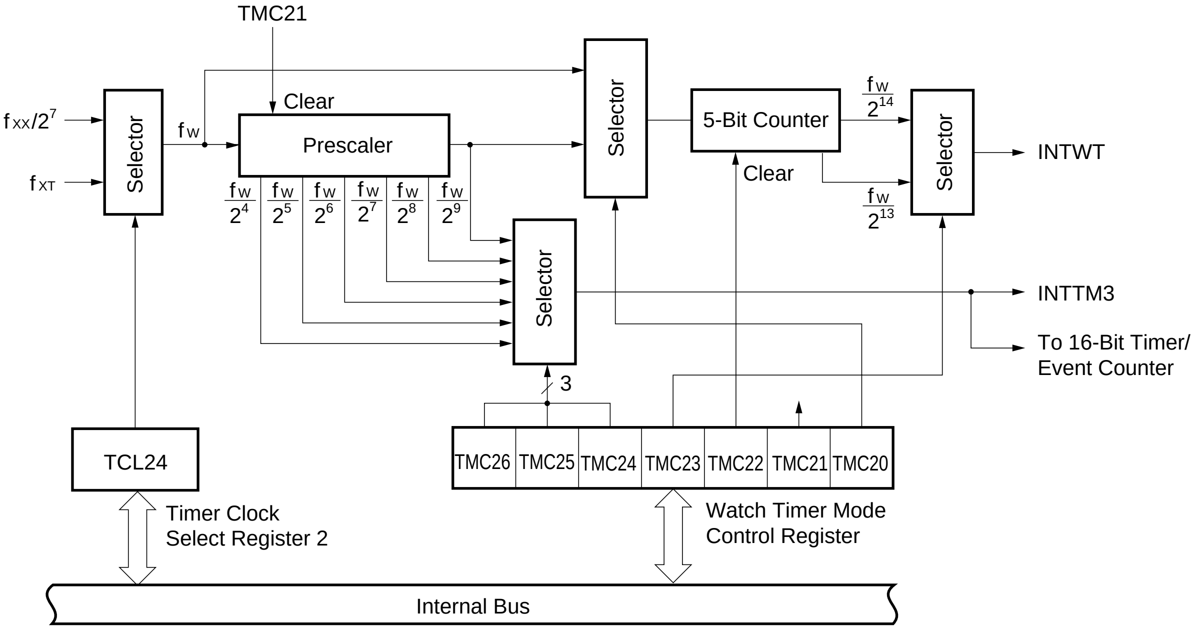


Figure 10-2. Timer Clock Select Register 2 Format

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

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

TCL24	Watch Timer Count Clock Selection		
		MCS = 1	MCS = 0
0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	f_{XT} (32.768 kHz)		

TCL27	TCL26	TCL25	Buzzer Output Frequency Selection		
				MCS = 1	MCS = 0
0	×	×	Buzzer output disable		
1	0	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)
1	0	1	$f_{XX}/2^{10}$	$f_X/2^{10}$ (4.9 kHz)	$f_X/2^{11}$ (2.4 kHz)
1	1	0	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)
1	1	1	Setting prohibited		

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

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

2. f_X : Main system clock oscillation frequency

3. f_{XT} : Subsystem clock oscillation frequency

4. × : don't care

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

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

(2) Watch timer mode control register (TMC2)

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

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

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

Figure 10-3. Watch Timer Mode Control Register Format

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

TMC20	Watch Operating Mode Selection
0	Normal operating mode (flag set at $f_W/2^{14}$)
1	Fast feed operating mode (flag set at $f_W/2^5$)

TMC21	Prescaler Operation Control
0	Clear after operation stop
1	Operation enable

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

TMC23	Watch Flag Set Time Selection		
	f_{XX} = 5.0-MHz Operation	f_{XX} = 4.19-MHz Operation	f_{XT} = 32.768-kHz Operation
0	$2^{14}/f_W$ (0.4 sec)	$2^{14}/f_W$ (0.5 sec)	$2^{14}/f_W$ (0.5 sec)
1	$2^{13}/f_W$ (0.2 sec)	$2^{13}/f_W$ (0.25 sec)	$2^{13}/f_W$ (0.25 sec)

TMC26	TMC25	TMC24	Prescaler Interval Time Selection		
			f_{XX} = 5.0-MHz Operation	f_{XX} = 4.19-MHz Operation	f_{XT} = 32.768-kHz Operation
0	0	0	$2^4/f_W$ (410 μs)	$2^4/f_W$ (488 μs)	$2^4/f_W$ (488 μs)
0	0	1	$2^5/f_W$ (819 μs)	$2^5/f_W$ (977 μs)	$2^5/f_W$ (977 μs)
0	1	0	$2^6/f_W$ (1.64 ms)	$2^6/f_W$ (1.95 ms)	$2^6/f_W$ (1.95 ms)
0	1	1	$2^7/f_W$ (3.28 ms)	$2^7/f_W$ (3.91 ms)	$2^7/f_W$ (3.91 ms)
1	0	0	$2^8/f_W$ (6.55 ms)	$2^8/f_W$ (7.81 ms)	$2^8/f_W$ (7.81 ms)
1	0	1	$2^9/f_W$ (13.1 ms)	$2^9/f_W$ (15.6 ms)	$2^9/f_W$ (15.6 ms)
Other than above			Setting prohibited		

Caution When the watch timer is used, the prescaler should not be cleared frequently.

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

2. f_{XX} : Main system clock frequency (f_X or $f_X/2$)

3. f_X : Main system clock oscillation frequency

4. f_{XT} : Subsystem clock oscillation frequency

10.4 Watch Timer Operations

10.4.1 Watch timer operation

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

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

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

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

10.4.2 Interval timer operation

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

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

Table 10-3. Interval Timer Interval Time

TMC26	TMC25	TMC24	Interval Time	When operated at f_{XX} = 5.0 MHz	When operated at f_{XX} = 4.19 MHz	When operated at f_{XT} = 32.768 kHz
0	0	0	$2^4 \times 1/f_W$	410 μs	488 μs	488 μs
0	0	1	$2^5 \times 1/f_W$	819 μs	977 μs	977 μs
0	1	0	$2^6 \times 1/f_W$	1.64 ms	1.95 ms	1.95 ms
0	1	1	$2^7 \times 1/f_W$	3.28 ms	3.91 ms	3.91 ms
1	0	0	$2^8 \times 1/f_W$	6.55 ms	7.81 ms	7.81 ms
1	0	1	$2^9 \times 1/f_W$	13.1 ms	15.6 ms	15.6 ms
Other than above			Setting prohibited			

Remark

f_{XX}	:	Main system clock frequency (f_X or $f_X/2$)
f_X	:	Main system clock oscillation frequency
f_{XT}	:	Subsystem clock oscillation frequency
f_W	:	Watch timer clock frequency ($f_{XX}/2^7$ or f_{XT})
TMC24 to TMC26	:	Bits 4 to 6 of watch timer mode control register (TMC2)

CHAPTER 11 WATCHDOG TIMER

11.1 Watchdog Timer Functions

The watchdog timer has the following functions.

- Watchdog timer
- Interval timer

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

(1) Watchdog timer mode

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

Table 11-1. Watchdog Timer Program Runaway Detection Times

Runaway Detection Time	MCS = 1	MCS = 0
2^{11} × 1/fxx	2^{11} × 1/fx (410 μs)	2^{12} × 1/fx (819 μs)
2^{12} × 1/fxx	2^{12} × 1/fx (819 μs)	2^{13} × 1/fx (1.64 ms)
2^{13} × 1/fxx	2^{13} × 1/fx (1.64 ms)	2^{14} × 1/fx (3.28 ms)
2^{14} × 1/fxx	2^{14} × 1/fx (3.28 ms)	2^{15} × 1/fx (6.55 ms)
2^{15} × 1/fxx	2^{15} × 1/fx (6.55 ms)	2^{16} × 1/fx (13.1 ms)
2^{16} × 1/fxx	2^{16} × 1/fx (13.1 ms)	2^{17} × 1/fx (26.2 ms)
2^{17} × 1/fxx	2^{17} × 1/fx (26.2 ms)	2^{18} × 1/fx (52.4 ms)
2^{19} × 1/fxx	2^{19} × 1/fx (104.9 ms)	2^{20} × 1/fx (209.7 ms)

Remarks 1. fxx : Main system clock frequency (fx or fx/2)

2. fx : Main system clock oscillation frequency

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

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

(2) Interval timer mode

Interrupt requests are generated at the preset time intervals.

Table 11-2. Interval Times

Interval Time	MCS = 1	MCS = 0
$2^{11} \times 1/f_{XX}$	$2^{11} \times 1/f_X$ (410 μs)	$2^{12} \times 1/f_X$ (819 μs)
$2^{12} \times 1/f_{XX}$	$2^{12} \times 1/f_X$ (819 μs)	$2^{13} \times 1/f_X$ (1.64 ms)
$2^{13} \times 1/f_{XX}$	$2^{13} \times 1/f_X$ (1.64 ms)	$2^{14} \times 1/f_X$ (3.28 ms)
$2^{14} \times 1/f_{XX}$	$2^{14} \times 1/f_X$ (3.28 ms)	$2^{15} \times 1/f_X$ (6.55 ms)
$2^{15} \times 1/f_{XX}$	$2^{15} \times 1/f_X$ (6.55 ms)	$2^{16} \times 1/f_X$ (13.1 ms)
$2^{16} \times 1/f_{XX}$	$2^{16} \times 1/f_X$ (13.1 ms)	$2^{17} \times 1/f_X$ (26.2 ms)
$2^{17} \times 1/f_{XX}$	$2^{17} \times 1/f_X$ (26.2 ms)	$2^{18} \times 1/f_X$ (52.4 ms)
$2^{19} \times 1/f_{XX}$	$2^{19} \times 1/f_X$ (104.9 ms)	$2^{20} \times 1/f_X$ (209.7 ms)

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

2. f_X : Main system clock oscillation frequency

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

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

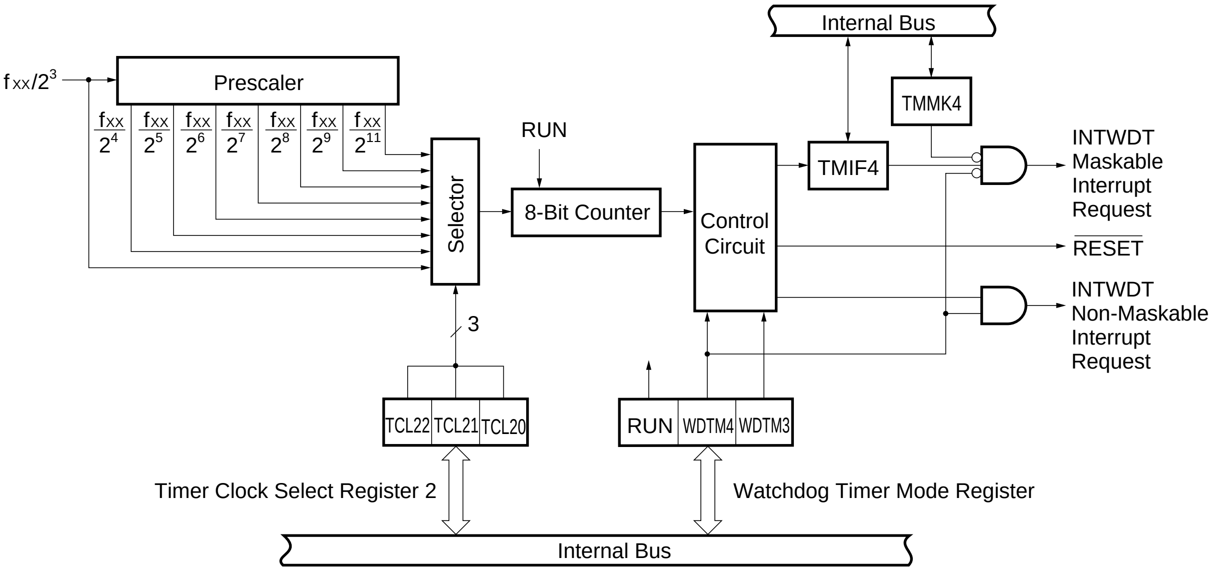
11.2 Watchdog Timer Configuration

The watchdog timer consists of the following hardware.

Table 11-3. Watchdog Timer Configuration

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

Figure 11-1. Watchdog Timer Block Diagram



11.3 Watchdog Timer Control Registers

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

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

(1) Timer clock select register 2 (TCL2)

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

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

Figure 11-2. Timer Clock Select Register 2 Format

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

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

TCL24	Watch Timer Count Clock Selection		
		MCS = 1	MCS = 0
0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	f_{XT} (32.768 kHz)		

TCL27	TCL26	TCL25	Buzzer Output Frequency Selection		
				MCS = 1	MCS = 0
0	×	×	Buzzer output disable		
1	0	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)
1	0	1	$f_{XX}/2^{10}$	$f_X/2^{10}$ (4.9 kHz)	$f_X/2^{11}$ (2.4 kHz)
1	1	0	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)
1	1	1	Setting prohibited		

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

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

2. f_X : Main system clock oscillation frequency

3. f_{XT} : Subsystem clock oscillation frequency

4. × : don't care

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

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

(2) Watchdog timer mode register (WDTM)

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

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

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

Figure 11-3. Watchdog Timer Mode Register Format

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
WDTM	RUM	0	0	WDTM4	WDTM3	0	0	0	FFF9H	00H	R/W

WDTM4	WDTM3	Watchdog Timer Operation Mode Selection Note 1
0	×	Interval timer mode Note 2 (Maskable interrupt request occurs upon generation of an overflow.)
1	0	Watchdog timer mode 1 (Non-maskable interrupt request occurs upon generation of an overflow.)
1	1	Watchdog timer mode 2 (Reset operation is activated upon generation of an overflow.)

RUN	Watchdog Timer Operation Mode Selection Note 3
0	Count stop
1	Counter is cleared and counting starts.

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

2. The watchdog timer starts operating as an interval timer as soon as RUN has been set to 1.

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

Cautions 1. When 1 is set in RUN so that the watchdog timer is cleared, the actual overflow time is up to 0.5% shorter than the time set by timer clock select register 2 (TCL2).

2. To use watchdog timer modes 1 and 2, make sure that the interrupt request flag (TMIF4) is 0, and then set WDTM4 to 1.
If WDTM4 is set to 1 when TMIF4 is 1, the non-maskable interrupt request occurs, regardless of the contents of WDTM3.

Remark ×: don’t care

11.4 Watchdog Timer Operations

11.4.1 Watchdog timer operation

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

The watchdog timer count clock (program runaway detection time interval) can be selected with bits 0 to 2 (TCL20 to TCL22) of the timer clock select register 2 (TCL2).

Watchdog timer starts by setting bit 7 (RUN) of WDTM to 1. After the watchdog timer is started, set RUN to 1 within the set runaway detection time interval. The watchdog timer can be cleared and counting is started by setting RUN to 1. If RUN is not set to 1 and the program runaway detection time is past, system reset or a non-maskable interrupt request is generated according to the WDTM bit 3 (WDTM3) value.

By setting RUN to 1, the watchdog timer can be cleared.

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 11-4. Watchdog Timer Program Runaway Detection Time

TCL22	TCL21	TCL20	Runaway Detection Time	MCS = 1	MCS = 0
0	0	0	2^{11} × 1/fxx	2^{11} × 1/fx (410 μs)	2^{12} × 1/fx (819 μs)
0	0	1	2^{12} × 1/fxx	2^{12} × 1/fx (819 μs)	2^{13} × 1/fx (1.64 ms)
0	1	0	2^{13} × 1/fxx	2^{13} × 1/fx (1.64 ms)	2^{14} × 1/fx (3.28 ms)
0	1	1	2^{14} × 1/fxx	2^{14} × 1/fx (3.28 ms)	2^{15} × 1/fx (6.55 ms)
1	0	0	2^{15} × 1/fxx	2^{15} × 1/fx (6.55 ms)	2^{16} × 1/fx (13.1 ms)
1	0	1	2^{16} × 1/fxx	2^{16} × 1/fx (13.1 ms)	2^{17} × 1/fx (26.2 ms)
1	1	0	2^{17} × 1/fxx	2^{17} × 1/fx (26.2 ms)	2^{18} × 1/fx (52.4 ms)
1	1	1	2^{19} × 1/fxx	2^{19} × 1/fx (104.9 ms)	2^{20} × 1/fx (209.7 ms)

Remarks 1. fxx : Main system clock frequency (fx or fx/2)

2. fx : Main system clock oscillation frequency

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

4. TCL20 to TCL22 : Bits 0 to 2 of timer clock select register 2 (TCL2)

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

11.4.2 Interval timer operation

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

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

When the watchdog timer operated as interval timer, the interrupt mask flag (TMMK4) and priority specify flag (TMPR4) are validated and the maskable interrupt request (INTWDT) can be generated. Among maskable interrupt requests, the INTWDT default has the highest priority.

The interval timer continues operating in the HALT mode but it stops in STOP mode. Thus, set bit 7 (RUN) of WDTM to 1 before the STOP mode is set, clear the interval timer and then execute the STOP instruction.

Cautions 1. Once bit 4 (WDTM4) of WDTM is set to 1 (with the watchdog timer mode selected), the interval timer mode is not set unless $\overline{\text{RESET}}$ input is applied.

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

3. When the subsystem clock is selected for CPU clock, watchdog timer count operation is stopped.

Table 11-5. Interval Timer Interval Time

TCL22	TCL21	TCL20	Interval Time	MCS = 1	MCS = 0
0	0	0	$2^{11} \times 1/f_{XX}$	$2^{11} \times 1/f_X$ (410 μs)	$2^{12} \times 1/f_X$ (819 μs)
0	0	1	$2^{12} \times 1/f_{XX}$	$2^{12} \times 1/f_X$ (819 μs)	$2^{13} \times 1/f_X$ (1.64 ms)
0	1	0	$2^{13} \times 1/f_{XX}$	$2^{13} \times 1/f_X$ (1.64 ms)	$2^{14} \times 1/f_X$ (3.28 ms)
0	1	1	$2^{14} \times 1/f_{XX}$	$2^{14} \times 1/f_X$ (3.28 ms)	$2^{15} \times 1/f_X$ (6.55 ms)
1	0	0	$2^{15} \times 1/f_{XX}$	$2^{15} \times 1/f_X$ (6.55 ms)	$2^{16} \times 1/f_X$ (13.1 ms)
1	0	1	$2^{16} \times 1/f_{XX}$	$2^{16} \times 1/f_X$ (13.1 ms)	$2^{17} \times 1/f_X$ (26.2 ms)
1	1	0	$2^{17} \times 1/f_{XX}$	$2^{17} \times 1/f_X$ (26.2 ms)	$2^{18} \times 1/f_X$ (52.4 ms)
1	1	1	$2^{19} \times 1/f_{XX}$	$2^{19} \times 1/f_X$ (104.9 ms)	$2^{20} \times 1/f_X$ (209.7 ms)

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

2. f_X : Main system clock oscillation frequency

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

4. TCL20 to TCL22 : Bits 0 to 2 of timer clock select register 2 (TCL2)

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

CHAPTER 12 CLOCK OUTPUT CONTROL CIRCUIT

12.1 Clock Output Control Circuit Functions

The clock output control circuit is intended for carrier output during remote controlled transmission and clock output for supply to peripheral LSI. Clocks selected with the timer clock select register 0 (TCL0) are output from the PCL/P35 pin.

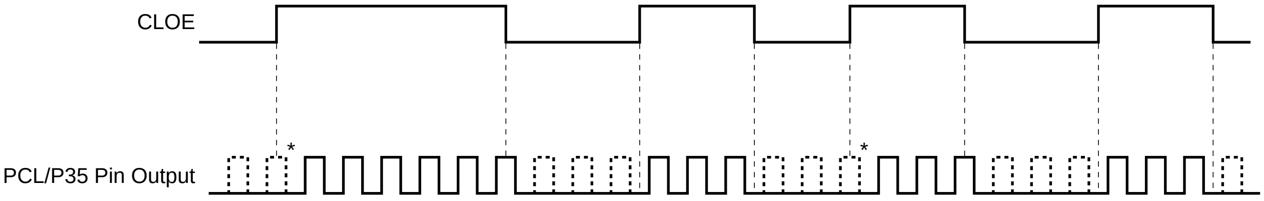
Follow the procedure below to output clock pulses.

(1) Select the clock pulse output frequency (with clock pulse output disabled) with bits 0 to 3 (TCL00 to TCL03) of TCL0.
(2) Set the P35 output latch to 0.
(3) Set bit 5 (PM35) of port mode register 3 (PM3) to 0 (set to output mode).
(4) Set bit 7 (CLOE) of timer clock select register 0 (TCL0) to 1.

Caution Clock output cannot be used when setting P35 output latch to 1.

Remark When clock output enable/disable is switched, the clock output control circuit does not output pulses with small widths (See the portions marked with * in **Figure 12-1**).

Figure 12-1. Remote Controlled Output Application Example



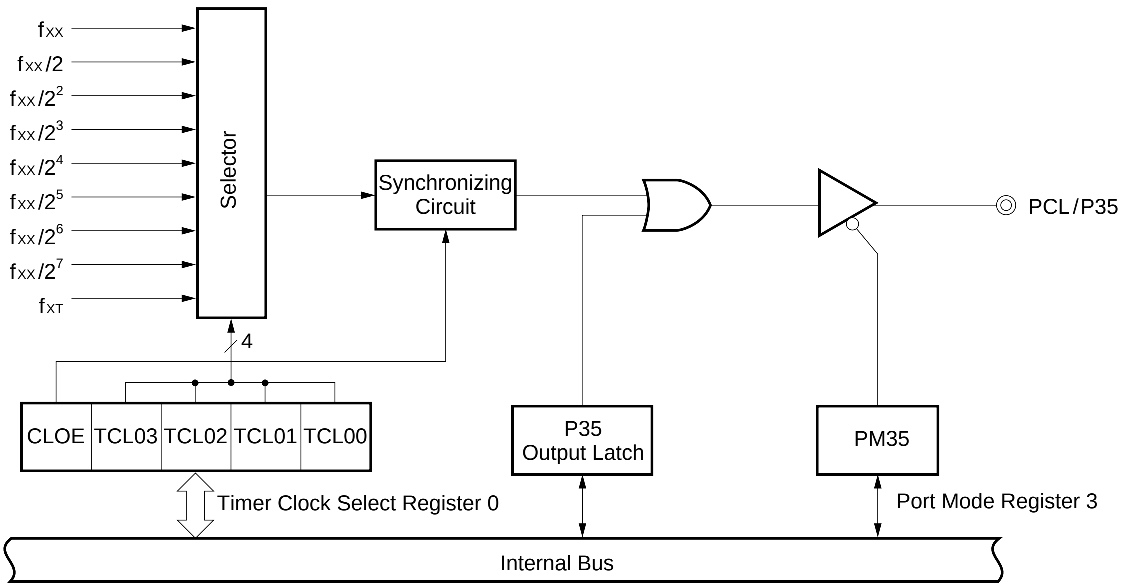
12.2 Clock Output Control Circuit Configuration

The clock output control circuit consists of the following hardware.

Table 12-1. Clock Output Control Circuit Configuration

Item	Configuration
Control register	Timer clock select register 0 (TCL0) Port mode register 3 (PM3)

Figure 12-2. Clock Output Control Circuit Block Diagram



12.3 Clock Output Function Control Registers

The following two types of registers are used to control the clock output function.

- Timer clock select register 0 (TCL0)
- Port mode register 3 (PM3)

(1) Timer clock select register 0 (TCL0)

This register sets PCL output clock.

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

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

Remark Besides setting PCL output clock, TCL0 sets the 16-bit timer register count clock.

Figure 12-3. Timer Clock Select Register 0 Format

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

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

TCL06	TCL05	TCL04	16-Bit Timer Register Count Clock Selection		
				MCS = 1	MCS = 0
0	0	0	TI00 (Valid edge specifiable)		
0	0	1	$2f_{XX}$	Setting prohibited	f_X (5.0 MHz)
0	1	0	f_{XX}	f_X (5.0 MHz)	$f_X/2$ (2.5 MHz)
0	1	1	$f_{XX}/2$	$f_X/2$ (2.5 MHz)	$f_X/2^2$ (1.25 MHz)
1	0	0	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	1	1	Watch Timer Output (INTTM3)		
Other than above			Setting prohibited		

CLOE	PCL Output Control
0	Output disable
1	Output enable

Cautions 1. TI00/P00/INTP0 pin valid edge is set by external interrupt mode register 0 (INTM0), and the sampling clock frequency is selected by the sampling clock selection register (SCS).

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

3. To read the count value when TI00 has been specified as the TM0 count clock, the value should be read from TM0, not from 16-bit capture/compare register (CR01).

4. When rewriting TCL0 to other data, stop the clock operation beforehand.

Remarks **1.** f_{XX} : Main system clock frequency (f_X or f_X/2)

2. f_X : Main system clock oscillation frequency

3. f_{XT} : Subsystem clock oscillation frequency

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

5. TM0 : 16-bit timer register

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

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

(2) Port mode register 3 (PM3)

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

When using the P35/PCL pin for clock output function, set PM35 and output latch of P35 to 0.

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

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

Figure 12-4. Port Mode Register 3 Format

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

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

CHAPTER 13 BUZZER OUTPUT CONTROL CIRCUIT

13.1 Buzzer Output Control Circuit Functions

The buzzer output control circuit outputs 1.2 kHz, 2.4 kHz, 4.9 kHz, or 9.8 kHz frequency square waves. The buzzer frequency selected with timer clock select register 2 (TCL2) is output from the BUZ/P36 pin.

Follow the procedure below to output the buzzer frequency.

(1) Select the buzzer output frequency with bits 5 to 7 (TCL25 to TCL27) of TCL2.
(2) Set the P36 output latch to 0.
(3) Set bit 6 (PM36) of port mode register 3 (PM3) to 0 (Set to output mode).

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

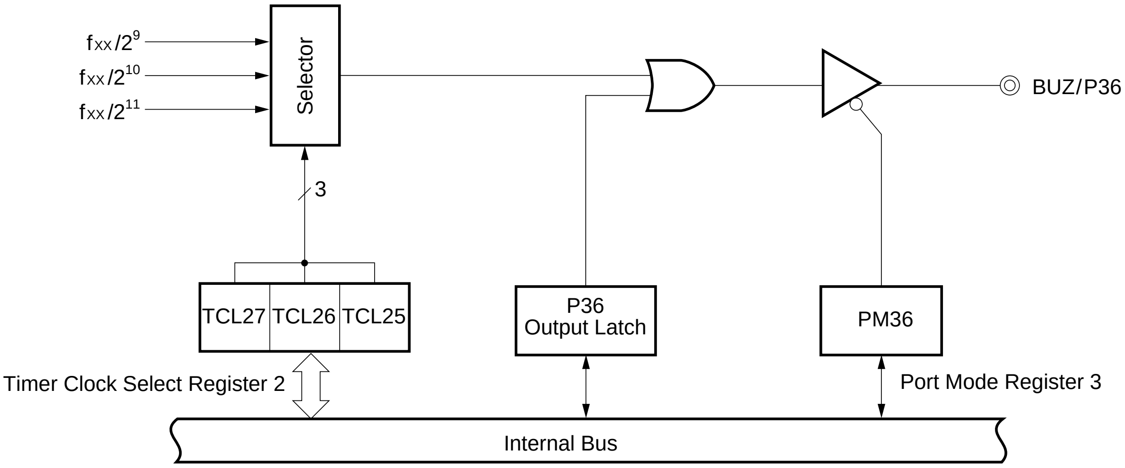
13.2 Buzzer Output Control Circuit Configuration

The buzzer output control circuit consists of the following hardware.

Table 13-1. Buzzer Output Control Circuit Configuration

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

Figure 13-1. Buzzer Output Control Circuit Block Diagram



13.3 Buzzer Output Function Control Registers

The following two types of registers are used to control the buzzer output function.

- Timer clock select register 2 (TCL2)
- Port mode register 3 (PM3)

(1) Timer clock select register 2 (TCL2)

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

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

Figure 13-2. Timer Clock Select Register 2 Format

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

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

TCL24	Watch Timer Count Clock Selection		
		MCS = 1	MCS = 0
0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	f_{XT} (32.768 kHz)		

TCL27	TCL26	TCL25	Buzzer Output Frequency Selection		
				MCS = 1	MCS = 0
0	×	×	Buzzer output disable		
1	0	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)
1	0	1	$f_{XX}/2^{10}$	$f_X/2^{10}$ (4.9 kHz)	$f_X/2^{11}$ (2.4 kHz)
1	1	0	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)
1	1	1	Setting prohibited		

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

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

2. f_X : Main system clock oscillation frequency

3. f_{XT} : Subsystem clock oscillation frequency

4. × : don't care

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

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

(2) Port mode register 3 (PM3)

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

When using the P36/BUZ pin for buzzer output function, set PM36 and output latch of P36 to 0.

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

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

Figure 13-3. Port Mode Register 3 Format

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

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

CHAPTER 14 A/D CONVERTER

14.1 A/D Converter Functions

The A/D converter converts an analog input into a digital value. It consists of 8 channels (ANI0 to ANI7) with an 8-bit resolution.

The conversion method is based on successive approximation and the conversion result is held in the 8-bit A/D conversion result register (ADCR).

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

(1) Hardware start

Conversion is started by trigger input (INTP3).

(2) Software start

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

One channel of analog input is selected from ANI0 to ANI7 and A/D conversion is carried out. In the case of hardware start, A/D conversion operation stops when an A/D conversion operation ends, and an interrupt request (INTAD) is generated. In the case of software start, the A/D conversion operation is repeated. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

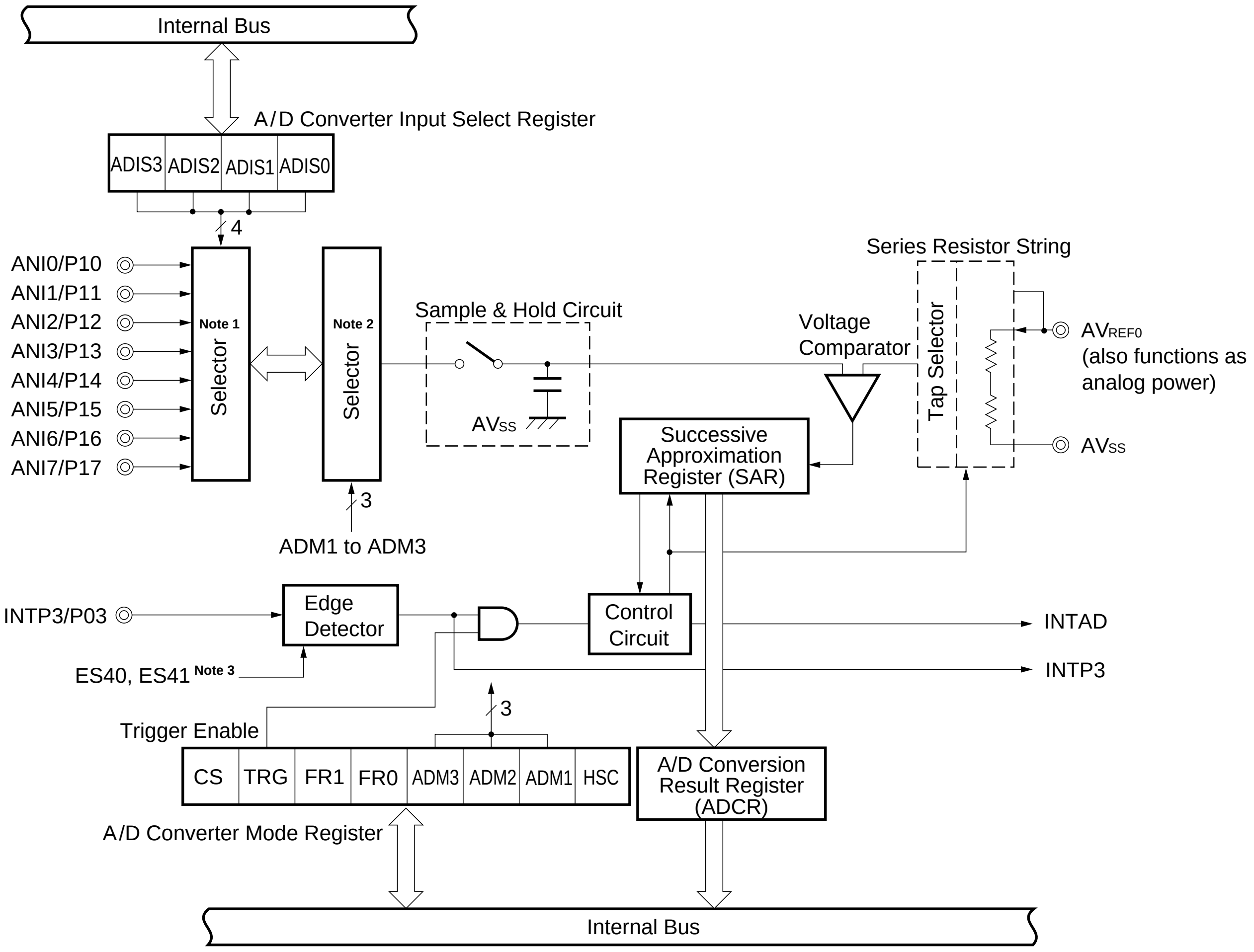
14.2 A/D Converter Configuration

The A/D converter consists of the following hardware.

Table 14-1. A/D Converter Configuration

Item	Configuration
Analog input	8 Channels (ANI0 to ANI7)
Control register	A/D converter mode register (ADM) A/D converter input select register (ADIS) External interrupt mode register 1 (INTM1)
Register	Successive approximation register (SAR) A/D conversion result register (ADCR)

Figure 14-1. A/D Converter Block Diagram



Notes 1. Selector to select the number of channels to be used for analog input.

2. Selector to select the channel for A/D conversion.

3. ES40, ES41: Bits 0 and 1 of external interrupt mode register 1 (INTM1)

(1) Successive approximation register (SAR)

This register compares the analog input voltage value to the voltage tap (compare voltage) value applied from the series resistor string and holds the result from the most significant bit (MSB).

When up to the least significant bit (LSB) is held (termination of A/D conversion), the SAR contents are transferred to the A/D conversion result register (ADCR).

(2) A/D conversion result register (ADCR)

This register holds the A/D conversion result. Each time A/D conversion terminates, the conversion result is loaded from the successive approximation register (SAR).

ADCR is read with an 8-bit memory manipulation instruction.

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

(3) Sample & hold circuit

The sample & hold circuit samples each analog input signal sequentially applied from the input circuit and sends it to the voltage comparator. This circuit holds the sampled analog input voltage value during A/D conversion.

(4) Voltage comparator

The voltage comparator compares the analog input to the series resistor string output voltage.

(5) Series resistor string

The series resistor string is connected between AV_{REF0} and AV_{SS}, and generates a voltage to be compared to the analog input.

(6) ANI0 to ANI7 pins

These are 8-channel analog input pins to input analog signals to undergo A/D conversion to the A/D converter.

Pins other than those selected as analog input by the A/D converter input select register (ADIS) can be used as input/output ports.

Cautions 1. Use ANI0 to ANI7 input voltages within the specified range. If a voltage higher than AV_{REF0} or lower than AV_{SS} is applied (even if within the absolute maximum ratings), the converted value of the corresponding channel becomes indeterminate and may adversely affect the converted values of other channels.

2. The analog input pins (ANI0 to ANI7) also function as input/output port (port 1) pins. When A/D conversion is performed with any of pins ANI0 to ANI7 selected, be sure not to execute an instruction that inputs data to port 1 while conversion is in progress, as this may reduce the conversion resolution.

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

(7) AV_{REF0} pin

This pin inputs the A/D converter reference voltage.
It converts signals input to ANI0 to ANI7 into digital signals according to the voltage applied between AV_{REF0} and AV_{SS}.
The current flowing in the series resistor string can be reduced by setting the voltage to be input to the AV_{REF0} pin to AV_{SS} level in standby mode.
This pin also serves as an analog power supply pin. Supply power to this pin when the A/D converter is used.

Caution A series resistor string of approximately 10 kΩ is connected between the AV_{REF0} pin and AV_{SS} pin. Therefore, if the output impedance of the reference voltage source is high, this will result in parallel connection to the series resistor string between AV_{REF0} pin and the AV_{SS} pin, resulting in a large reference voltage error.

(8) AV_{SS} pin

This is a GND potential pin of the A/D converter. Keep it at the same potential as the V_{SS0} pin when not using the A/D converter.

14.3 A/D Converter Control Registers

The following three types of registers are used to control the A/D converter.

- A/D converter mode register (ADM)
- A/D converter input select register (ADIS)
- External interrupt mode register 1 (INTM1)

(1) A/D converter mode register (ADM)

This register sets the analog input channel for A/D conversion, conversion time, conversion start/stop and external trigger.

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

$\overline{\text{RESET}}$ input sets ADM to 01H.

Figure 14-2. A/D Converter Mode Register Format

Symbol	⑦	⑥	5	4	3	2	1	0	Address	After Reset	R/W
ADM	CS	TRG	FR1	FR0	ADM3	ADM2	ADM1	HSC	FF80H	01H	R/W

ADM3	ADM2	ADM1	Analog Input Channel Selection
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

FR1	FR0	HSC	A/D Conversion Time Selection Note 1			
			f_X = 5.0-MHz Operation		f_X = 4.19-MHz Operation	
			MCS = 1	MCS = 0	MCS = 1	MCS = 0
0	0	1	80/f_X (Setting prohibited Note 2)	160/f_X (32.0 µs)	80/f_X (19.1 µs)	160/f_X (38.1 µs)
0	1	1	40/f_X (Setting prohibited Note 2)	80/f_X (Setting prohibited Note 2)	40/f_X (Setting prohibited Note 2)	80/f_X (19.1 µs)
1	0	0	50/f_X (Setting prohibited Note 2)	100/f_X (20.0 µs)	50/f_X (Setting prohibited Note 2)	100/f_X (23.8 µs)
1	0	1	100/f_X (20.0 µs)	200/f_X (40.0 µs)	100/f_X (23.8 µs)	200/f_X (47.7 µs)
Other than above			Setting prohibited			

TRG	External Trigger Selection
0	No external trigger (software starts)
1	Conversion started by external trigger (hardware starts)

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

Notes 1. Set so that the A/D conversion time is 19.1 µs or more.

2. Setting prohibited because A/D conversion time is less than 19.1 µs.

Cautions 1. The following sequence is recommended for power consumption reduction of A/D converter when the standby function is used: Clear bit 7 (CS) to 0 first to stop the A/D conversion operation, and then execute the HALT or STOP instruction.

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

Remarks 1. f_X : Main system clock oscillation frequency

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

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

This register determines whether the ANI0/P10 to ANI7/P17 pins should be used for analog input channels or ports. Pins other than those selected as analog input can be used as input/output ports.

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

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

Cautions 1. Set the analog input channel in the following order.

(1) Set the number of analog input channels with ADIS.

(2) Using A/D converter mode register (ADM), select one channel to undergo A/D conversion from among the channels set for analog input with ADIS.

2. No internal pull-up resistor can be used to the channels set for analog input with ADIS, irrespective of the value of bit 1 (PUO1) of the pull-up resistor option register L (PUOL).

Figure 14-3. A/D Converter Input Select Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADIS	0	0	0	0	ADIS3	ADIS2	ADIS1	ADIS0	FF84H	00H	R/W

ADIS3	ADIS2	ADIS1	ADIS0	Number of Analog Input Channel Selection
0	0	0	0	No analog input channel (P10 to P17)
0	0	0	1	1 channel (ANI0, P11 to P17)
0	0	1	0	2 channel (ANI0, ANI1, P12 to P17)
0	0	1	1	3 channel (ANI0 to ANI2, P13 to P17)
0	1	0	0	4 channel (ANI0 to ANI3, P14 to P17)
0	1	0	1	5 channel (ANI0 to ANI4, P15 to P17)
0	1	1	0	6 channel (ANI0 to ANI5, P16, P17)
0	1	1	1	7 channel (ANI0 to ANI6, P17)
1	0	0	0	8 channel (ANI0 to ANI7)
Other than above				Setting prohibited

(3) External interrupt mode register 1 (INTM1)

This register sets the valid edge for INTP3 to INTP5.

INTM1 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears INTM1 to 00H.

Figure 14-4. External Interrupt Mode Register 1 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
INTM1	0	0	ES61	ES60	ES51	ES50	ES41	ES40	FFEDH	00H	R/W

ES41	ES40	INTP3 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES51	ES50	INTP4 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES61	ES60	INTP5 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

14.4 A/D Converter Operations

14.4.1 Basic operations of A/D converter

(1) Set the number of analog input channels with A/D converter input select register (ADIS).

(2) From among the analog input channels set with ADIS, select one channel for A/D conversion with A/D converter mode register (ADM).

(3) Sample the voltage input to the selected analog input channel with the sample & hold circuit.

(4) Sampling for the specified period of time sets the sample & hold circuit to the hold state so that the circuit holds the input analog voltage until termination of A/D conversion.

(5) Bit 7 of the successive approximation register (SAR). The series resistor string voltage tap is set to (1/2) AV_{REF0} by the tap selector.

(6) The voltage difference between the series resistor string voltage tap and analog input is compared with a voltage comparator. If the analog input is greater than (1/2) AV_{REF0}, the MSB of SAR remains set. If the input is smaller than (1/2) AV_{REF0}, the MSB is reset.

(7) Next, bit 6 of SAR is automatically set and the operation proceeds to the next comparison. In this case, the series resistor string voltage tap is selected according to the preset value of bit 7 as described below.

- Bit 7 = 1 : (3/4) AV_{REF0}
- Bit 7 = 0 : (1/4) AV_{REF0}

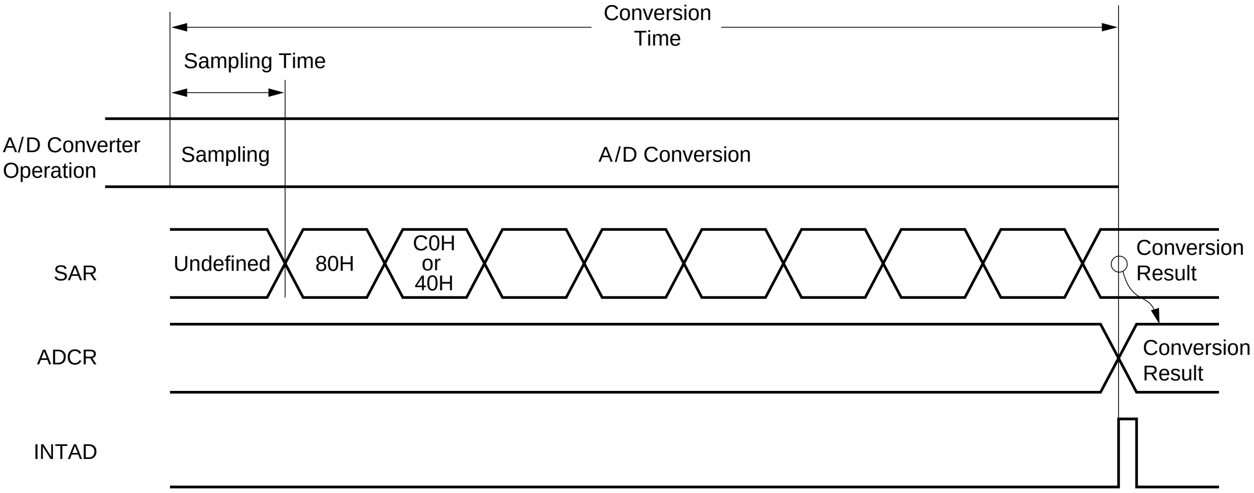
The voltage tap and analog input voltage are compared and bit 6 of SAR is manipulated with the result as follows.

- Analog input voltage ≥ Voltage tap : Bit 6 = 1
- ★ Analog input voltage < Voltage tap : Bit 6 = 0

(8) Comparison of this sort continues up to bit 0 of SAR.

(9) Upon completion of the comparison of 8 bits, any effective digital resultant value remains in SAR and the resultant value is transferred to and latched in the A/D conversion result register (ADCR).
At the same time, the A/D conversion termination interrupt request (INTAD) can also be generated.

Figure 14-5. A/D Converter Basic Operation



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

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

$\overline{\text{RESET}}$ input, makes ADCR undefined.

14.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI7) and the A/D conversion result (the value stored in A/D conversion result register (ADCR)) is shown by the following expression.

$$\mathrm{ADCR} = \mathrm{INT}\left(\frac{V_{\mathrm{IN}}}{AV_{\mathrm{REF0}}} \times 256 + 0.5\right)$$

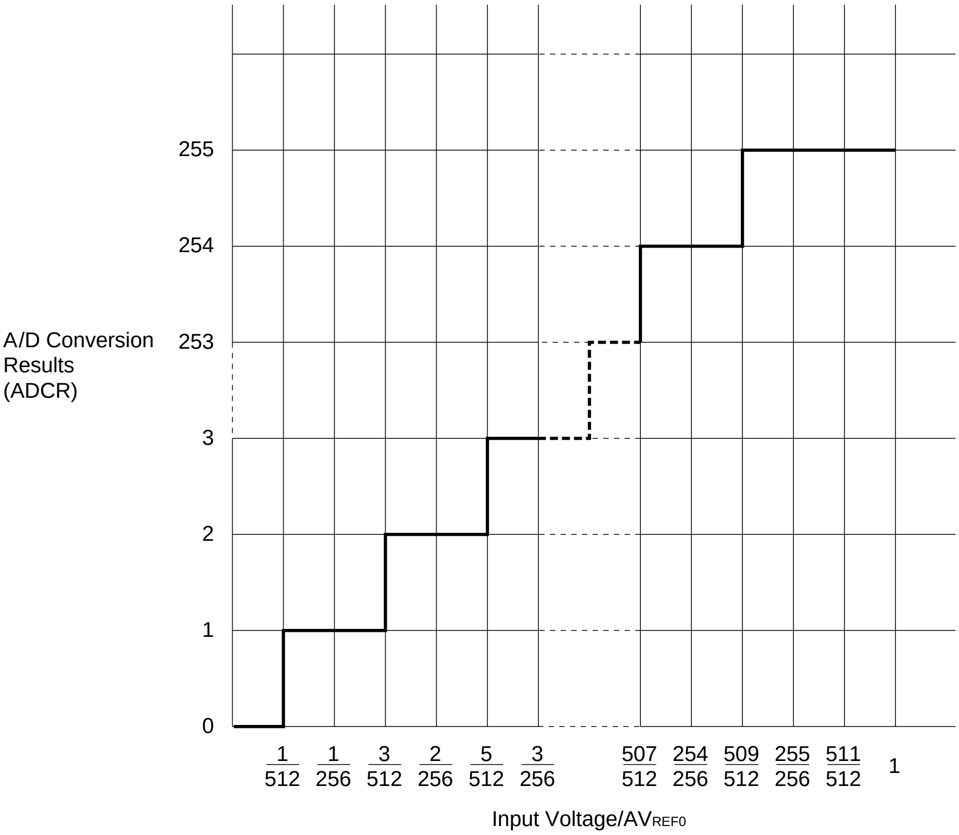
or

$$(\mathrm{ADCR} - 0.5) \times \frac{AV_{\mathrm{REF0}}}{256} \leq V_{\mathrm{IN}} < (\mathrm{ADCR} + 0.5) \times \frac{AV_{\mathrm{REF0}}}{256}$$

Where, INT( ) : Function which returns integer parts of value in parentheses.
V_{IN} : Analog input voltage
AV_{REF0} : AV_{REF0} pin voltage
ADCR : Value of A/D conversion result register (ADCR)

Figure 14-6 shows the relation between the analog input voltage and the A/D conversion result.

Figure 14-6. Relationships Between Analog Input Voltage and A/D Conversion Result



14.4.3 A/D converter operating mode

One analog input channel is selected from among ANI0 to ANI7 with the A/D converter input select register (ADIS) and A/D converter mode register (ADM) and A/D conversion is started.

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

- Hardware start: Conversion is started by trigger input (INTP3).
- Software start: Conversion is started by setting ADM.

The A/D conversion result is stored in the A/D conversion result register (ADCR) and the interrupt request signal (INTAD) is simultaneously generated.

(1) A/D conversion by hardware start

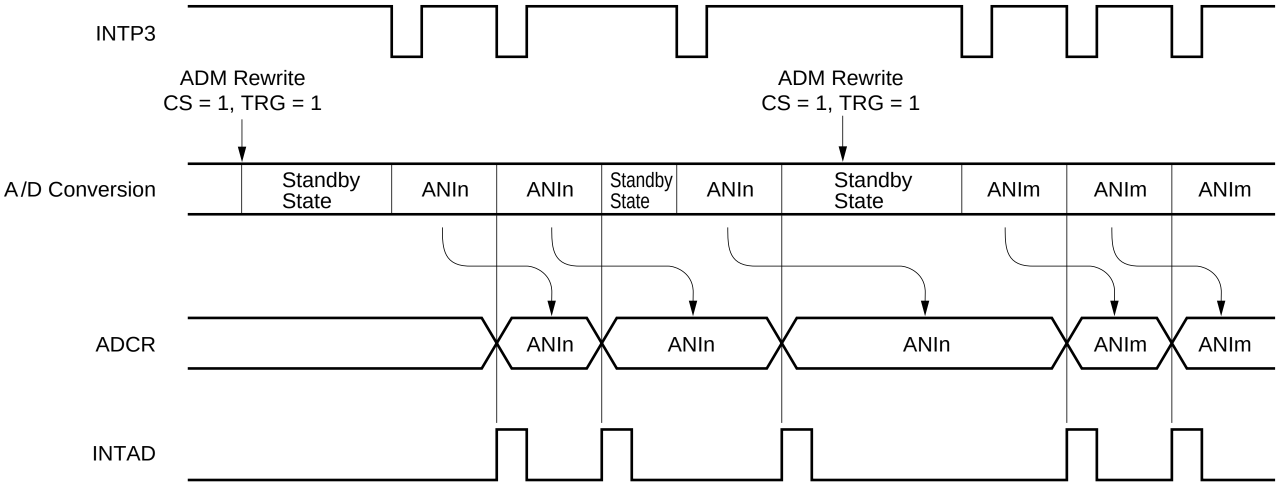
When bit 6 (TRG) and bit 7 (CS) of A/D converter mode register (ADM) are set to 1, the A/D conversion standby state is set. When the external trigger signal (INTP3) is input, the A/D conversion starts on the voltage applied to the analog input pins specified with bits 1 to 3 (ADM1 to ADM3) of ADM.

Upon termination of the A/D conversion, the conversion result is stored in the A/D conversion result register (ADCR) and the interrupt request signal (INTAD) is generated. After one A/D conversion operation is started and terminated, another operation is not started until a new external trigger signal is input.

If data with CS set to 1 is written to ADM again during A/D conversion, the converter suspends its A/D conversion operation and waits for a new external trigger signal to be input. When the external trigger input signal is reinput, A/D conversion is carried out from the beginning.

If data with CS set to 0 is written to ADM during A/D conversion, the A/D conversion operation stops immediately.

Figure 14-7. A/D Conversion by Hardware Start



Remarks 1. n = 0, 1, ..., 7

2. m = 0, 1, ..., 7

(2) A/D conversion operation in software start

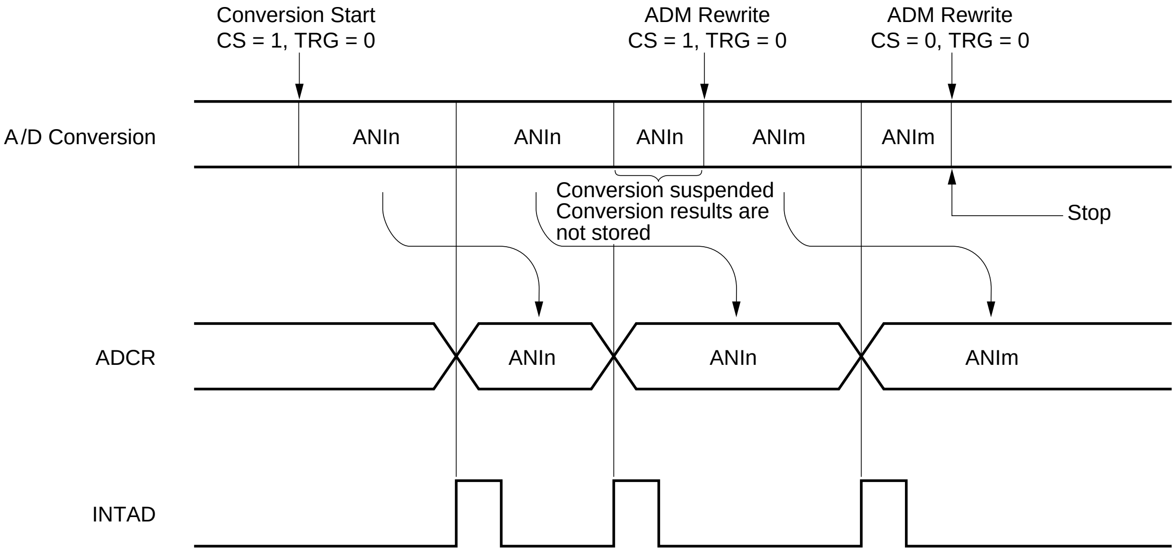
When bit 6 (TRG) and bit 7 (CS) of A/D converter mode register (ADM) are set to 0 and 1, respectively, the A/D conversion starts on the voltage applied to the analog input pins specified with bits 1 to 3 (ADM1 to ADM3) of ADM.

Upon termination of the A/D conversion, the conversion result is stored in the A/D conversion result register (ADCR) and the interrupt request signal (INTAD) is generated. After one A/D conversion operation is started and terminated, the next A/D conversion operation starts immediately. The A/D conversion operation continues repeatedly until new data is written to ADM.

If data with CS set to 1 is written to ADM again during A/D conversion, the converter suspends its A/D conversion operation and starts A/D conversion on the newly written data.

If data with CS set to 0 is written to ADM during A/D conversion, the A/D conversion operation stops immediately.

Figure 14-8. A/D Conversion by Software Start



Remarks 1. n = 0, 1, ..., 7

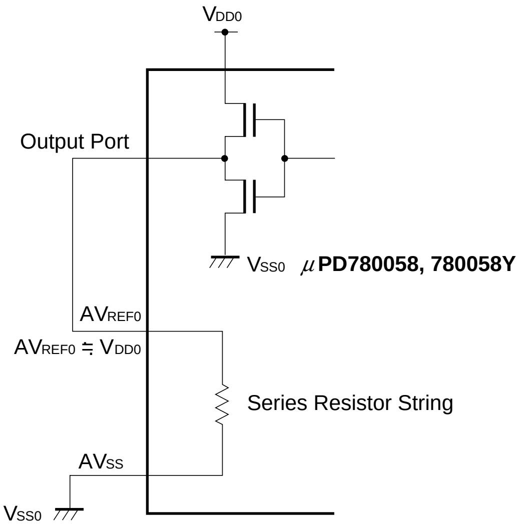
2. m = 0, 1, ..., 7

14.5 A/D Converter Cautions

(1) Power consumption in standby mode

The A/D converter operates on the main system clock. Therefore, its operation stops in STOP mode or in HALT mode with the subsystem clock. As a current still flows in the AV_{REF0} pin at this time, this current must be cut in order to minimize the overall system power consumption. In Figure 14-9, the power consumption can be reduced by outputting a low-level signal to the output port in standby mode. However, there is no precision to the actual AV_{REF0} voltage, and therefore the conversion values themselves lack precision and can only be used for relative comparison.

Figure 14-9. Example of Method of Reducing Current Consumption in Standby Mode



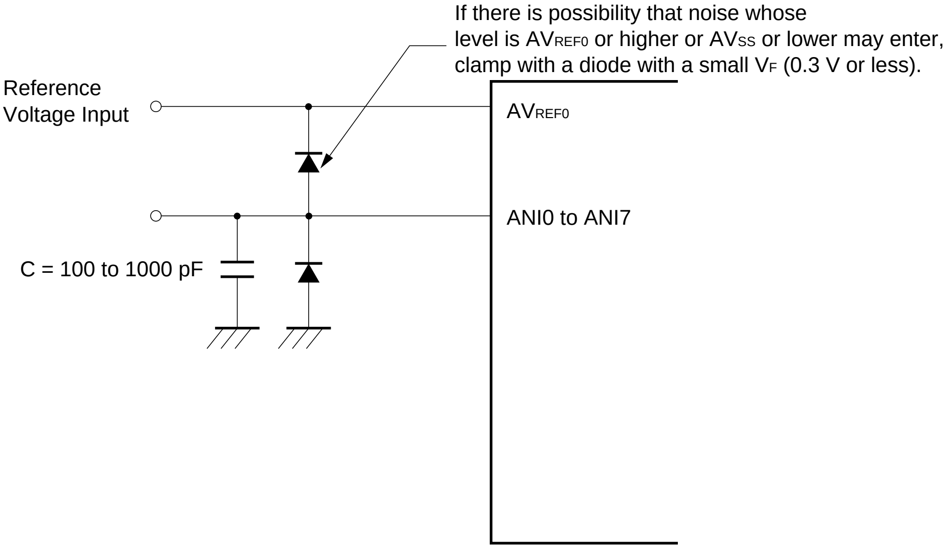
(2) Input range of ANI0 to ANI7

The input voltages of ANI0 to ANI7 should be within the specification range. In particular, if a voltage above AV_{REF0} or below AV_{SS} is input (even if within the absolute maximum rating range), the conversion value for that channel will be indeterminate. The conversion values of the other channels may also be affected.

(3) Noise countermeasures

In order to maintain 8-bit resolution, attention must be paid to noise on AV_{REF0} and ANI0 to ANI7 pins. Since the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally as shown in Figure 14-10 in order to reduce noise.

Figure 14-10. Analog Input Pin Disposition


If there is possibility that noise whose level is AV_{REF0} or higher or AV_{SS} or lower may enter, clamp with a diode with a small V_F (0.3 V or less).
Reference Voltage Input
AV_{REF0}
ANI0 to ANI7
C = 100 to 1000 pF

(4) Pins ANI0/P10 to ANI7/P17

The analog input pins ANI0 to ANI7 also function as input/output port (port 1) pins. When A/D conversion is performed with any of pins ANI0 to ANI7 selected, be sure not to execute an instruction that inputs data to port 1 while conversion is in progress, as this may reduce the conversion resolution.

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

(5) AV_{REF0} pin input impedance

A series resistor string of approximately 10 kΩ is connected between the AV_{REF0} pin and the AV_{SS} pin.

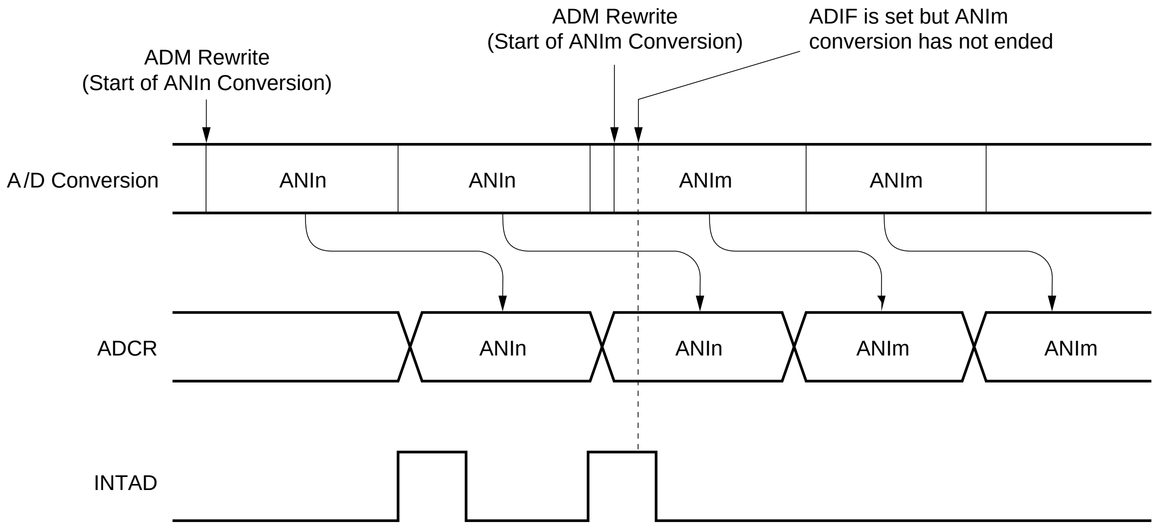
Therefore, if the output impedance of the reference voltage source is high, this will result in parallel connection to the series resistor string between the AV_{REF0} pin and the AV_{SS} pin, and there will be a large reference voltage error.

(6) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the A/D converter mode register (ADM) is changed. Caution is therefore required since, if a change of analog input pin is performed during A/D conversion, the A/D conversion result and ADIF for the pre-change analog input may be set just before the ADM rewrite. At this time, when ADIF is read immediately after the ADM rewrite, ADIF may be set despite the fact that the A/D conversion for the post-change analog input has not ended.

When the A/D conversion is stopped and then resumed, clear the ADIF before it is resumed.

Figure 14-11. A/D Conversion End Interrupt Request Generation Timing


ADM Rewrite
(Start of ANIm Conversion)
ADIF is set but ANIm conversion has not ended
ADM Rewrite
(Start of ANIn Conversion)
A/D Conversion
ANIn
ANIn
ANIm
ANIm
ADCR
ANIn
ANIn
ANIm
ANIm
INTAD

Remark n = 0, 1, ..., 7
m = 0, 1, ..., 7

(7) Conversion result immediately after A/D converter start

The first A/D conversion value immediately after A/D conversion is started may not satisfy ratings. Therefore, implement a countermeasure such as polling A/D conversion end interrupt requests (INTAD) to delete the first conversion result.

CHAPTER 15 D/A CONVERTER

15.1 D/A Converter Functions

The D/A converter converts a digital input into an analog value. It consists of two 8-bit resolution channels of voltage output type D/A converter.

The conversion method used is the R-2R resistor ladder method.

Start the D/A conversion by setting bits 0 and 1 (DACE0 and DACE1) of the D/A converter mode register (DAM).

There are two types of modes for the D/A converter, as follows.

(1) Normal mode

Outputs an analog voltage signal immediately after the D/A conversion.

(2) Real-time output mode

Outputs an analog voltage signal synchronously with the output trigger after the D/A conversion.

Since a sine wave can be generated in the mode, it is useful for an MSK modem for cordless telephone sets.

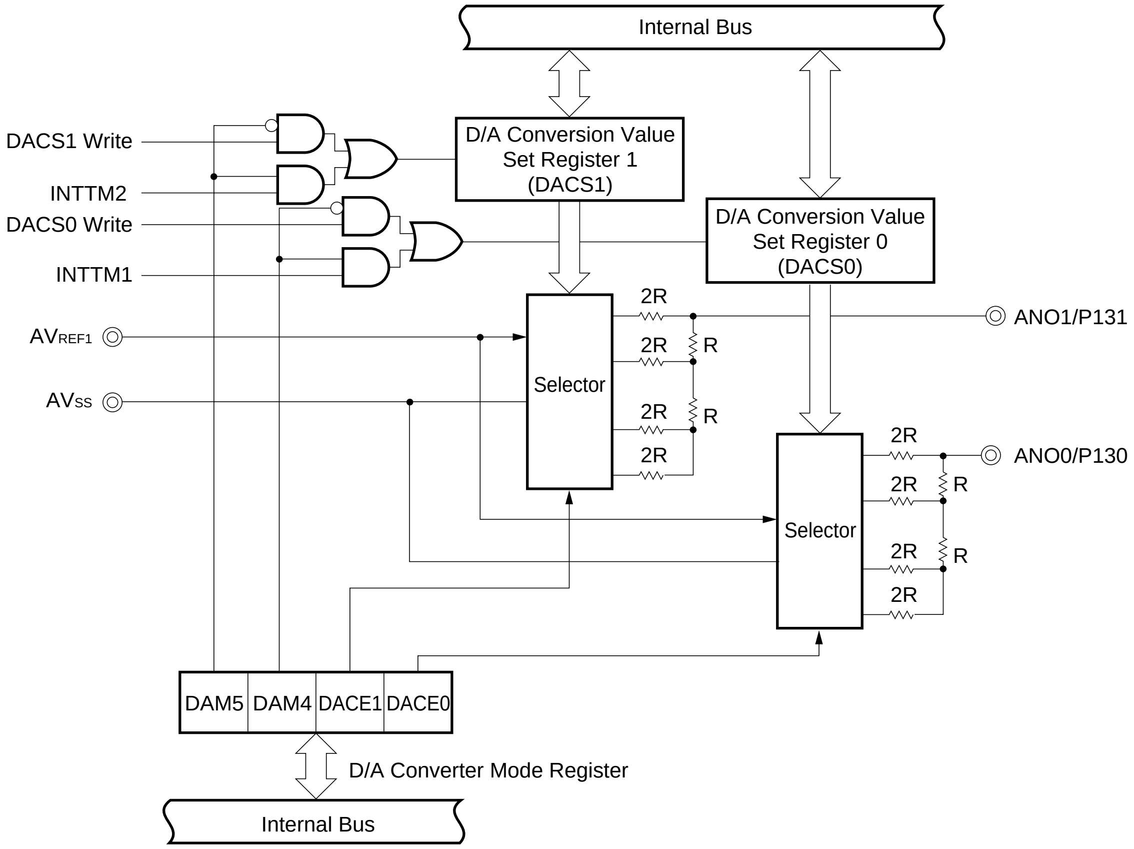
15.2 D/A Converter Configuration

The D/A converter consists of the following hardware.

Table 15-1. D/A Converter Configuration

Item	Configuration
Register	D/A conversion value set register 0 (DACS0) D/A conversion value set register 1 (DACS1)
Control register	D/A converter mode register (DAM)

Figure 15-1. D/A Converter Block Diagram



(1) D/A conversion value set register 0, 1 (DACS0, DACS1)

DACS0 and DACS1 are registers that set the values used to determine the analog voltages to be output to the ANO0 and ANO1 pins, respectively.

DACS0 and DACS1 are set with an 8-bit memory manipulation instruction.

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

Analog voltage output to the ANO0 and ANO1 pins is determined by the following expression.

$$\text{ANOn output voltage} = \text{AV}_{\text{REF1}} \times \frac{\text{DACSn}}{256}$$

where, n = 0, 1

Cautions 1. In the real-time output mode, when data that are set in DACS0 and DACS1 are read before an output trigger is generated, the previous data are read rather than the set data.

2. In the real-time output mode, data should be set to DACS0 and DACS1 after an output trigger and before the next output trigger.

15.3 D/A Converter Control Registers

The D/A converter mode register (DAM) controls the D/A converter. This register sets D/A converter operation enable/stop.

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

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

Figure 15-2. D/A Converter Mode Register Format

Symbol	7	6	5	4	3	2	①	⓪	Address	After Reset	R/W
DAM	0	0	DAM5	DAM4	0	0	DACE1	DACE0	FF98H	00H	R/W

DACE0	D/A Converter Channel 0 Control
0	D/A conversion stop
1	D/A conversion enable

DACE1	D/A Converter Channel 1 Control
0	D/A conversion stop
1	D/A conversion enable

DAM4	D/A Converter Channel 0 Operating Mode
0	Normal mode
1	Real-time output mode

DAM5	D/A Converter Channel 1 Operating Mode
0	Normal mode
1	Real-time output mode

Cautions 1. When using the D/A converter, an alternate function port pin should be set to the input mode, and a pull-up resistor should be disconnected.

2. Always set bits 2, 3, 6, and 7 to 0.

3. When D/A conversion is stopped, the output state is high-impedance.

4. The output triggers are INTTM1 and INTTM2 for channel 0 and channel 1, respectively, in the real-time output mode.

15.4 D/A Converter Operations

(1) The channel 0 operating mode and channel 1 operating mode are selected by bits 4 and 5 (DAM4 and DAM5), respectively, of the D/A converter mode register (DAM).

(2) Set the data corresponding to the analog voltages output to the ANO0/P130 and ANO1/P131 pins to the D/A conversion value setting registers 0 and 1 (DACS0 and DACS1), respectively.

(3) The D/A conversion of channel 0 or channel 1 can be started by setting bits 0 or 1 (DACE0 or DACE1) of DAM, respectively.

(4) In the normal mode, the analog voltage signals are output to the ANO0/P130 and ANO1/P131 pins immediately after the D/A conversion. In the real-time output mode, the analog voltage signals are output synchronously with the output triggers.

(5) In the normal mode, the analog voltage signals to be output are held until new data are set in DACS0 and DACS1. In the realtime output mode, new data are set in DACS0 and DACS1 and then they are held until the next trigger is generated.

Caution Set DACE0 and DACE1 after setting data in DACS0 and DACS1.

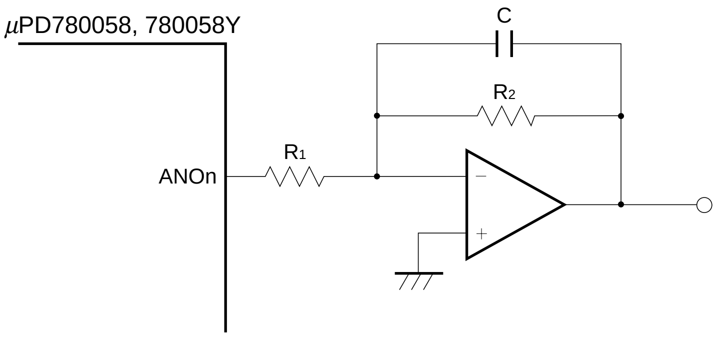
15.5 D/A Converter Cautions

(1) Output impedance of D/A converter

Because the output impedance of the D/A converter is high, use of current flowing from the ANOn pins (n = 0,1) is prohibited. If the input impedance of the load for the converter is low, insert a buffer amplifier between the load and the ANOn pins. In addition, wiring from the ANOn pins to the buffer amplifier or the load should be as short as possible (because of high output impedance). If the wiring may be long, design the ground pattern so as to be close to those lines or use some other expedient to achieve shorter wiring.

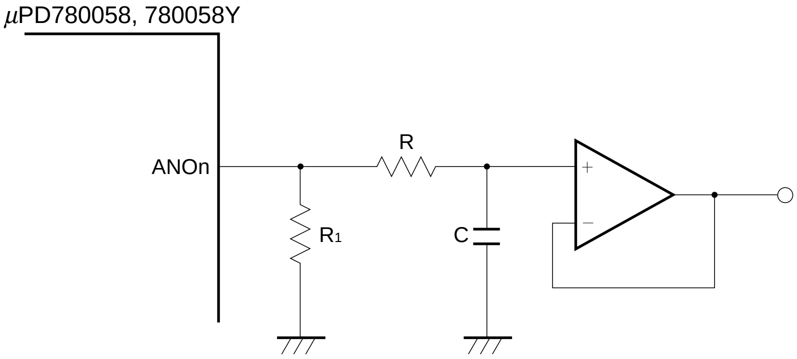
Figure 15-3. Use Example of Buffer Amplifier

(a) Inverting amplifier



• The input impedance of the buffer amplifier is R_1.

(b) Voltage-follower



• The input impedance of the buffer amplifier is R_1 .
• If R_1 is not connected, the output becomes undefined when $\overline{\text{RESET}}$ is low.

(2) Output voltage of D/A converter

Because the output voltage of the converter changes in steps, use the D/A converter output signals in general by connecting a low-pass filter.

(3) AV_{REF1} pin

When only either one of the D/A converter channels is used with $AV_{REF1} < V_{DD0}$, the other pins that are not used as analog outputs must be set as follows:

- Set PM13x bit of the port mode register 13 (PM13) to 1 (input mode) and connect the pin to V_{SS0}.
- Set PM13x bit of the port mode register 13 (PM13) to 0 (output mode) and the output latch to 0, to output low level from the pin.

CHAPTER 16 SERIAL INTERFACE CHANNEL 0 (μPD780058 Subseries)

The μPD780058 Subseries incorporates three channels of serial interfaces. Differences between channels 0, 1, and 2 are as follows (refer to **CHAPTER 18 SERIAL INTERFACE CHANNEL 1** for details of the serial interface channel 1. Refer to **CHAPTER 19 SERIAL INTERFACE CHANNEL 2** for details of the serial interface channel 2).

Table 16-1. Differences Between Channels 0, 1, and 2

Serial Transfer Mode		Channel 0	Channel 1	Channel 2
3-wire serial I/O	Clock selection	$f_{XX}/2$, $f_{XX}/2^2$, $f_{XX}/2^3$, $f_{XX}/2^4$, $f_{XX}/2^5$, $f_{XX}/2^6$, $f_{XX}/2^7$, $f_{XX}/2^8$, external clock, TO2 output	$f_{XX}/2$, $f_{XX}/2^2$, $f_{XX}/2^3$, $f_{XX}/2^4$, $f_{XX}/2^5$, $f_{XX}/2^6$, $f_{XX}/2^7$, $f_{XX}/2^8$, external clock, TO2 output	External clock, baud rate generator output
	Transfer method	MSB/LSB switchable as the start bit	MSB/LSB switchable as the start bit Automatic transmit/receive function	MSB/LSB switchable as the start bit
	Transfer end flag	Serial transfer end interrupt request flag (CSIIF0)	Serial transfer end interrupt request flag (CSIIF1)	Serial transfer end interrupt request flag (SRIF)
SBI (serial bus interface)		Use possible	None	None
2-wire serial I/O				
UART (Asynchronous serial interface)		None		Use possible Time-division transfer function

16.1 Serial Interface Channel 0 Functions

Serial interface channel 0 employs the following four modes.

- Operation stop mode
- 3-wire serial I/O mode
- SBI (serial bus interface) mode
- 2-wire serial I/O mode

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

(1) Operation stop mode

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

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

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

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

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

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

This mode is used for 8-bit data transfer with two or more devices using two lines of serial clock ($\overline{\text{SCK0}}$) and serial data bus (SB0 or SB1) (refer to **Figure 16-1**).

The SBI mode conforms to the NEC serial bus format, and transmits or receives three types of transfer data: “addresses”, “commands”, “data”.

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

Actually, the master device outputs an “address” to the serial bus to select one of the slave devices with which the master device is to communicate. After that, “commands” and “data” are transmitted or received between the master and slave devices (this is the serial transfer). The receiver can automatically identify the received data as an “address”, “command”, or “data” by hardware.

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

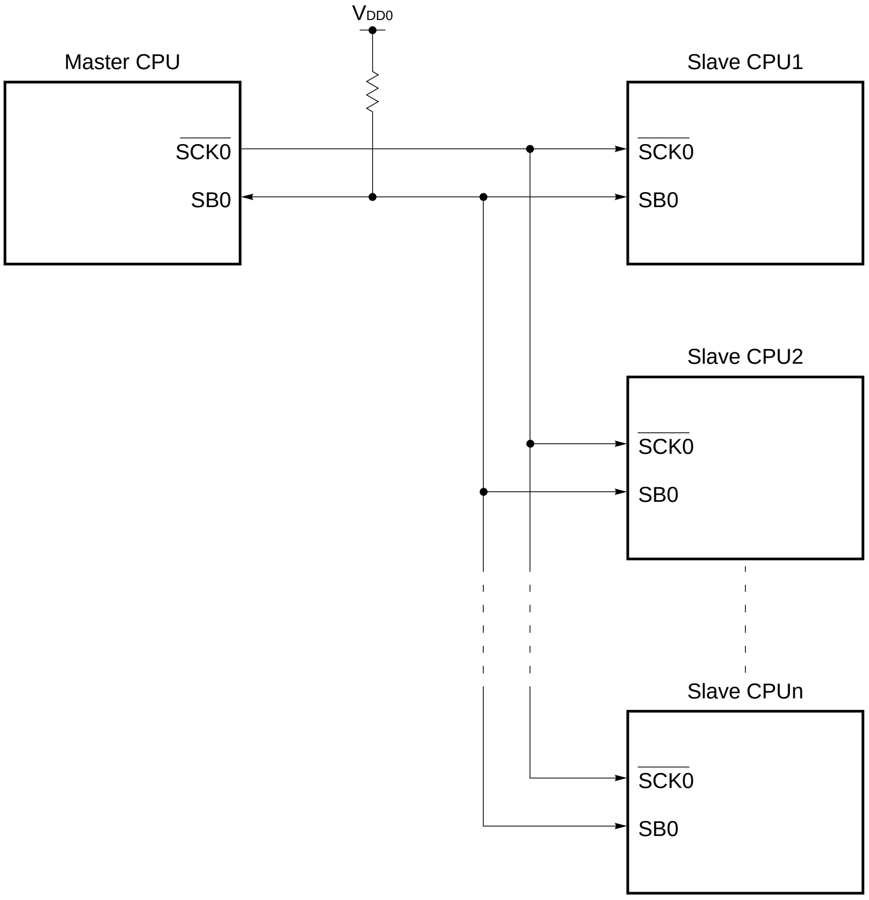
In this mode, the wake-up function for handshake and the output function of acknowledge and busy signals can also be used.

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

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

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

Figure 16-1. Serial Bus Interface (SBI) System Configuration Example



16.2 Serial Interface Channel 0 Configuration

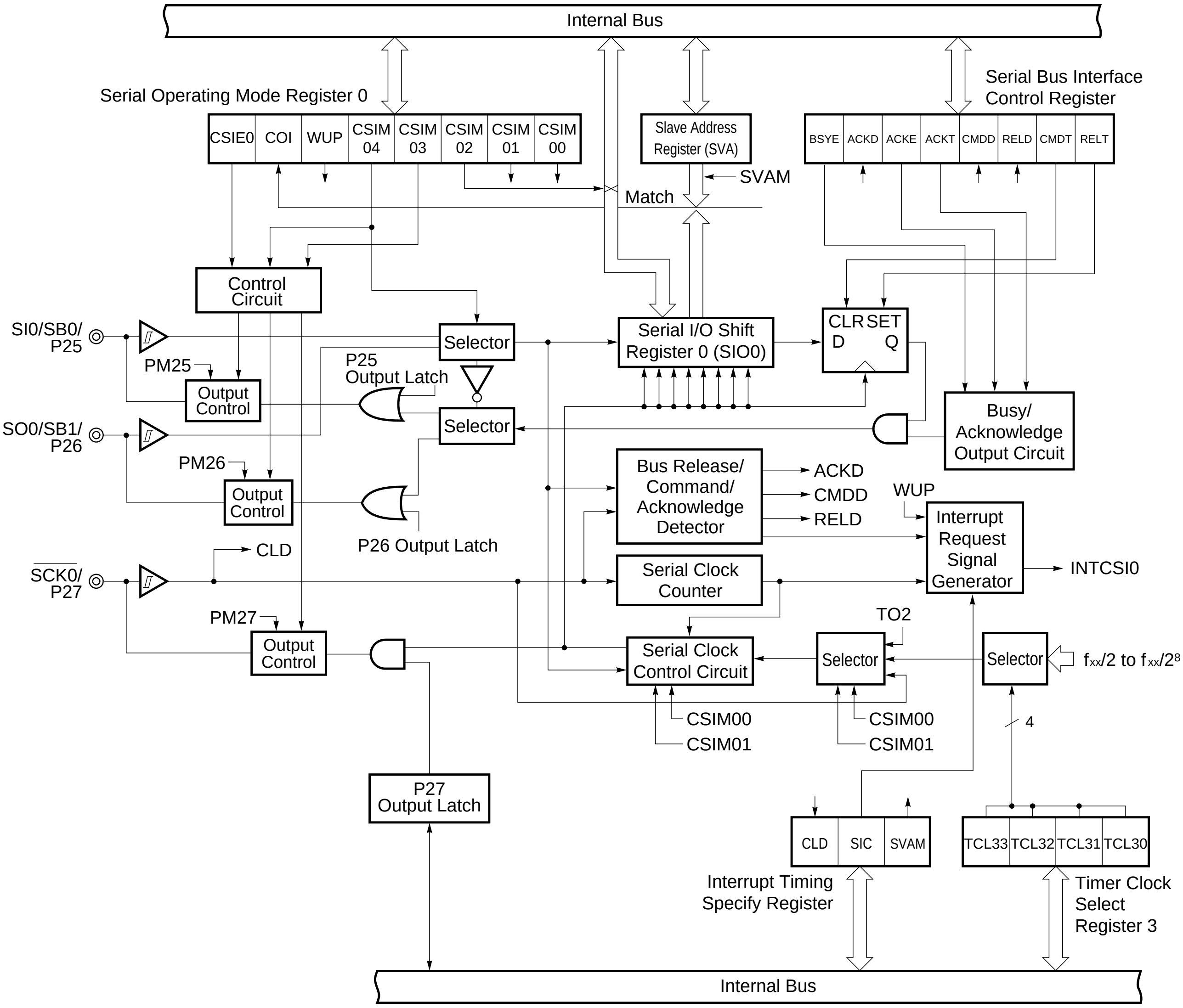
Serial interface channel 0 consists of the following hardware.

Table 16-2. Serial Interface Channel 0 Configuration

Item	Configuration
Register	Serial I/O shift register 0 (SIO0) Slave address register (SVA)
Control register	Timer clock select register 3 (TCL3) Serial operating mode register 0 (CSIM0) Serial bus interface control register (SBIC) Interrupt timing specify register (SINT) Port mode register 2 (PM2) **Note**

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

Figure 16-2. Serial Interface Channel 0 Block Diagram



Remark Output Control performs selection between CMOS output and N-ch open-drain output.

(1) Serial I/O shift register 0 (SIO0)

This is an 8-bit register to carry out parallel/serial conversion and to carry out serial transmission/reception (shift operation) in synchronization with the serial clock.

The SIO0 is set with an 8-bit memory manipulation instruction.

When bit 7 (CSIE0) of serial operating mode register 0 (CSIM0) is 1, writing data to SIO0 starts serial operation.

In transmission, data written to the SIO0 is output to the serial output (SO0) or serial data bus (SB0/SB1).

In reception, data is read from the serial input (SI0) or SB0/SB1 to SIO0.

Note that, if a bus is driven in the SBI mode or 2-wire serial I/O mode, the bus pin must serve for both input and output. Thus, in the case of a device for reception, write FFH to SIO0 in advance (except when address reception is carried out by setting bit 5 (WUP) of CSIM0 to 1).

In the SBI mode, the busy state can be cleared by writing data to SIO0. In this case, bit 7 (BSYE) of the serial bus interface control register (SBIC) is not cleared to 0.

$\overline{\text{RESET}}$ input makes SIO0 undefined.

(2) Slave address register (SVA)

This is an 8-bit register to set the slave address value for connection of a slave device to the serial bus.

The SVA is set with an 8-bit memory manipulation instruction. This register is not used in the 3-wire serial I/O mode.

The master device outputs a slave address for selection of a particular slave device to the connected slave device. These two data (the slave address output from the master device and the SVA value) are compared with an address comparator. If they match, the slave device has been selected. In that case, bit 6 (COI) of serial operating mode register 0 (CSIM0) becomes 1.

Address comparison can also be executed on the data of LSB-masked high-order 7 bits by setting bit 4 (SVAM) of the interrupt timing specify register (SINT) to 1.

If no matching is detected in address reception, bit 2 (RELD) of the serial bus interface control register (SBIC) is cleared to 0. In the SBI mode, the wake-up function can be used by setting the bit 5 (WUP) of CSIM0 to 1. In this case, the interrupt request signal (INTCSI0) is generated only when the slave address output by the master coincides with the value of the SVA, and it can be learned by this interrupt request that the master requests for communication. If the bit 5 (SIC) of the interrupt timing specify register (SINT) is set to 1, the wake-up function cannot be used even if WUP is set to 1 (an interrupt request signal is generated when bus release is detected). To use the wake-up function, clear SIC to 0.

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

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

(3) SO0 latch

This latch holds SI0/SB0/P25 and SO0/SB1/P26 pin levels. It can be directly controlled by software. In the SBI mode, this latch is set upon termination of the 8th serial clock.

(4) Serial clock counter

This counter counts the serial clocks to be output and input during transmission/reception and to check whether 8-bit data has been transmitted/received.

(5) Serial clock control circuit

This circuit controls serial clock supply to the serial I/O shift register 0 (SIO0). When the internal system clock is used, the circuit also controls clock output to the $\overline{\text{SCK0}}$/P27 pin.

(6) Interrupt request signal generator

This circuit controls interrupt request signal generation. It generates the interrupt request signal in the following cases.

- **In the 3-wire serial I/O mode and 2-wire serial I/O mode**
  This circuit generates an interrupt request signal every eight serial clocks.

- **In the SBI mode**
  When WUP is 0 .... Generates an interrupt request signal every eight serial clocks.
  When WUP is 1 .... Generates an interrupt request signal when the serial I/O shift register 0 (SIO0) value matches the slave address register (SVA) value after address reception.

Remark WUP is wake-up function specify bit. It is bit 5 of serial operating mode register 0 (CSIM0). When using the wake-up function (WUP = 1), clear the bit 5 (SIC) of the interrupt timing specify register (SINT) to 0.

(7) Busy/acknowledge output circuit and bus release/command/acknowledge detector

These two circuits output and detect various control signals in the SBI mode.

These do not operate in the 3-wire serial I/O mode and 2-wire serial I/O mode.

16.3 Serial Interface Channel 0 Control Registers

The following four types of registers are used to control serial interface channel 0.

- Timer clock select register 3 (TCL3)
- Serial operating mode register 0 (CSIM0)
- Serial bus interface control register (SBIC)
- Interrupt timing specify register (SINT)

(1) Timer clock select register 3 (TCL3)

This register sets the serial clock of serial interface channel 0.
TCL3 is set with an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets TCL3 to 88H.

Figure 16-3. Timer Clock Select Register 3 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33	TCL32	TCL31	TCL30	Serial Interface Channel 0 Serial Clock Selection		
					MCS = 1	MCS = 0
0	1	1	0	$f_{XX}/2$	Setting prohibited	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
Other than above				Setting prohibited		

TCL37	TCL36	TCL35	TCL34	Serial Interface Channel 1 Serial Clock Selection		
					MCS = 1	MCS = 0
0	1	1	0	$f_{XX}/2$	Setting prohibited	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
Other than above				Setting prohibited		

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

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

2. f_X : Main system clock oscillation frequency

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

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

(2) Serial operating mode register 0 (CSIM0)

This register sets serial interface channel 0 serial clock, operating mode, operation enable/stop wake-up function and displays the address comparator match signal.

CSIM0 is set with a 1-bit or an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM0 to 00H.

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

Figure 16-4. Serial Operating Mode Register 0 Format (1/2)

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W [Note 1]

R/W

CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection
0	×	Input Clock to $\overline{\text{SCK0}}$ pin from off-chip
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operation Mode	Start Bit	SI0/SB0/P25 Pin Function	SO0/SB1/P26 Pin Function	$\overline{\text{SCK0}}$/P27 Pin Function
0	×	0	1 [Note 2]	× [Note 2]	0	0	0	1	3-wire serial I/O mode	MSB	SI0 [Note 2] (Input)	SO0 (CMOS output)	$\overline{\text{SCK0}}$ (CMOS input/output)
		1								LSB			
1	0	0	× [Note 3]	× [Note 3]	0	0	0	1	SBI mode	MSB	P25 (CMOS input/output)	SB1 (N-ch open-drain input/output)	$\overline{\text{SCK0}}$ (CMOS input/output)
		1	0	0	× [Note 3]	× [Note 3]	0	1			SB0 (N-ch open-drain input/output)	P26 (CMOS input/output)	
1	1	0	× [Note 3]	× [Note 3]	0	0	0	1	2-wire serial I/O mode	MSB	P25 (CMOS input/output)	SB1 (N-ch open-drain input/output)	$\overline{\text{SCK0}}$ (N-ch open-drain input/output)
		1	0	0	× [Note 3]	× [Note 3]	0	1			SB0 (N-ch open-drain input/output)	P26 (CMOS input/output)	

(Cont'd)

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

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

3. Can be used freely as port function.

Remark		
×	:	don't care
PM××	:	Port mode register
P××	:	Port output latch

Figure 16-4. Serial Operating Mode Register 0 Format (2/2)

R/W	WUP	Wake-up Function Control [Note 1]
	0	Interrupt request signal generation with each serial transfer in any mode
	1	Interrupt request signal generation when the address received after bus release (when CMDD = RELD = 1) matches the slave address register (SVA) data in SBI mode

R	COI	Slave Address Comparison Result Flag [Note 2]
	0	Slave address register (SVA) not equal to serial I/O shift register 0 (SIO0) data
	1	Slave address register (SVA) equal to serial I/O shift register 0 (SIO0) data

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

Notes 1. When using the wake-up function (WUP = 1), clear the bit 5 (SIC) of the interrupt timing specify register (SINT) to 0.

2. When CSIE0 = 0, COI becomes 0.

(3) Serial bus interface control register (SBIC)

This register sets serial bus interface operation and displays statuses.

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

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

Figure 16-5. Serial Bus Interface Control Register Format (1/2)

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

R/W	RELT	Used for bus release signal output. When RELT = 1, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W	CMDT	Used for command signal output. When CMDT = 1, SO0 latch is cleared to 0. After SO0 latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R

RELD	Bus Release Detection
Clear Conditions (RELD = 0)	Set Conditions (RELD = 1)
• When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied	• When bus release signal (REL) is detected

R

CMDD	Command Detection
Clear Conditions (CMDD = 0)	Set Conditions (CMDD = 1)
• When transfer start instruction is executed • When bus release signal (REL) is detected • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied	• When command signal (CMD) is detected

R/W	ACKT	Acknowledge signal is output in synchronization with the falling edge of $\overline{\text{SCK0}}$ clock just after execution of the instruction that sets this bit to 1, and after acknowledge signal output, automatically cleared to 0. Used as ACKE = 0. Also cleared to 0 upon start of serial interface transfer or when CSIE0 = 0.

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

Remarks 1. Bits 0, 1, and 4 (RELT, CMDT, and ACKT) are 0 when read after data setting.

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

Figure 16-5. Serial Bus Interface Control Register Format (2/2)

R/W

ACKE	Acknowledge Signal Automatic Output Control	
0	Acknowledge signal automatic output disable (output with ACKT enable)	
1	Before completion of transfer	Acknowledge signal is output in synchronization with the falling edge of the 9th $\overline{\text{SCK0}}$ clock (automatically output when ACKE = 1).
	After completion of transfer	Acknowledge signal is output in synchronization with the falling edge of $\overline{\text{SCK0}}$ just after execution of the instruction that sets this bit to 1 (automatically output when ACKE = 1). However, not automatically cleared to 0 after acknowledge signal output.

R

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

R/W

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

★ **Note** The busy mode can be canceled by start of serial interface transfer. However, the BSYE flag is not cleared to 0.

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

(4) Interrupt timing specify register (SINT)

This register sets the bus release interrupt and address mask functions and displays the $\overline{\text{SCK0}}$/P27 pin level status.

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

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

Figure 16-6. Interrupt Timing Specify Register Format

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

R/W

SVAM	SVA Bit to be Used as Slave Address
0	Bits 0 to 7
1	Bits 1 to 7

R/W

SIC	INTCSI0 Interrupt Sourse Selection [Note 2]
0	CSIIF0 is set upon termination of serial interface channel 0 transfer
1	CSIIF0 is set upon bus release detection or termination of serial interface channel 0 transfer

R

CLD	$\overline{\text{SCK0}}$/P27 Pin Level [Note 3]
0	Low level
1	High level

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

2. When using wake-up function in the SBI mode, set SIC to 0.

3. When CSIE0 = 0, CLD becomes 0.

Caution Be sure to set bits 0 to 3 to 0.

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

16.4 Serial Interface Channel 0 Operations

The following four operating modes are available to the serial interface channel 0.

- Operation stop mode
- 3-wire serial I/O mode
- SBI mode
- 2-wire serial I/O mode

16.4.1 Operation stop mode

Serial transfer is not carried out in the operation stop mode. Thus, power consumption can be reduced. The serial I/O shift register 0 (SIO0) does not carry out shift operation either and thus it can be used as ordinary 8-bit register.

In the operation stop mode, the P25/SI0/SB0, P26/SO0/SB1, and P27/$\overline{\text{SCK0}}$ pins can be used as ordinary input/output ports.

(1) Register setting

The operation stop mode is set with the serial operating mode register 0 (CSIM0).

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM0 to 00H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W

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

16.4.2 3-wire serial I/O mode operation

The 3-wire serial I/O mode is valid for connection of peripheral I/O units and display controllers which incorporate a conventional synchronous clocked serial interface as is the case with the 75X/XL, 78K, and 17K Series.

Communication is carried out with three lines of serial clock ($\overline{\text{SCK0}}$), serial output (SO0), and serial input (SI0).

(1) Register setting

The 3-wire serial I/O mode is set with the serial operating mode register 0 (CSIM0) and serial bus interface control register (SBIC).

(a) Serial operating mode register 0 (CSIM0)

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM0 to 00H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W Note 1

R/W

CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection
0	×	Input clock to $\overline{\text{SCK0}}$ pin from off-chip
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operation Mode	Start Bit	SIO/SB0/P25 Pin Function	SO0/SB1/P26 Pin Function	$\overline{\text{SCK0}}$/P27 Pin Function
0	×	0	Note 2 1	Note 2 ×	0	0	0	1	3-wire serial I/O mode	MSB	SI0 Note 2 (Input)	SO0 (CMOS output)	$\overline{\text{SCK0}}$ (CMOS input/output)
		1								LSB			
1	0	SBI mode (see **16.4.3 SBI mode operation**.)											
1	1	2-wire serial I/O mode (see **16.4.4 2-wire serial I/O mode operation**.)											

R/W

WUP	Wake-up Function Control Note 3
0	Interrupt request signal generation with each serial transfer in any mode
1	Interrupt request signal generation when the address received after bus release (when CMDD = RELD = 1) matches the slave address register data in SBI mode

R/W

CSIE0	Serial Interface Channel 0 Operation Control
0	Operation stopped
1	Operation enabled

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

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

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

Remark		
×	:	don’t care
PM××	:	Port mode register
P××	:	Port output latch

(b) Serial bus interface control register (SBIC)

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

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

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

R/W	RELT	When RELT = 1, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W	CMDT	When CMDT = 1, SO0 latch is cleared to 0. After SO0 latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

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

(2) Communication operation

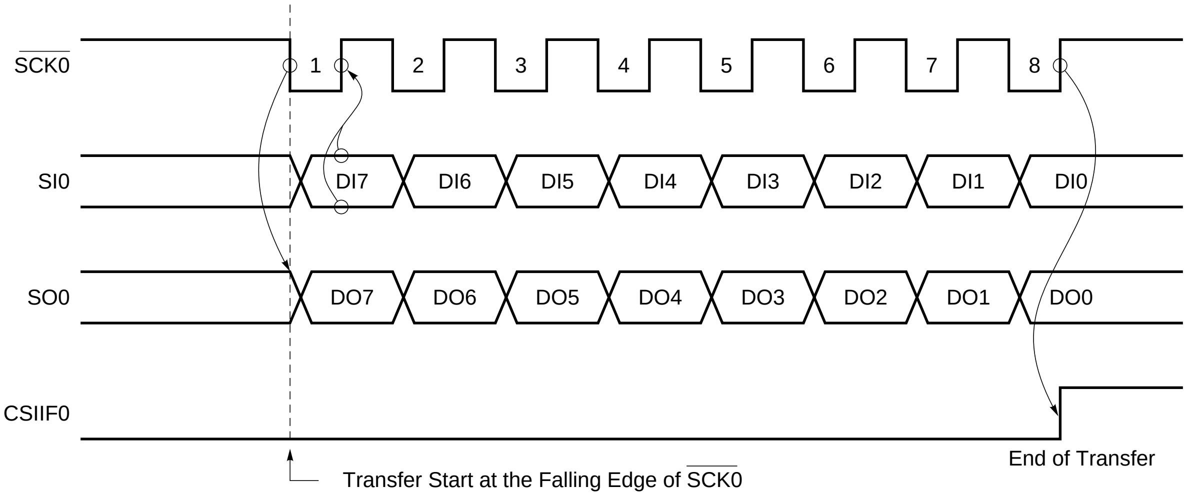
The 3-wire serial I/O mode is used for data transmission/reception in 8-bit units. Bit-wise data transmission/reception is carried out in synchronization with the serial clock.

Shift operation of the serial I/O shift register 0 (SIO0) is carried out at the falling edge of the serial clock ($\overline{\text{SCK0}}$).

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

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

Figure 16-7. 3-wire Serial I/O Mode Timings



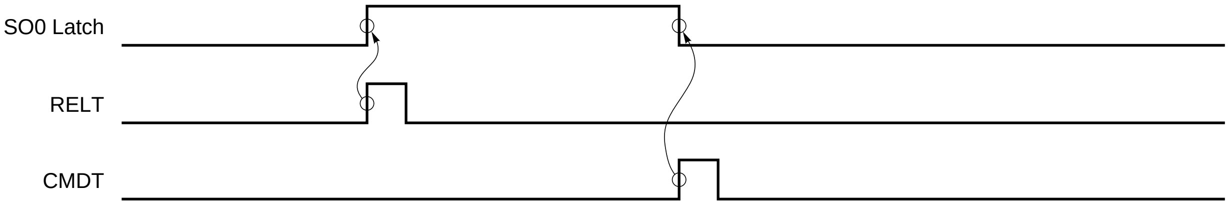
The SO0 pin is a CMOS output pin and outputs current SO0 latch statuses. Thus, the SO0 pin output status can be manipulated by setting bit 0 (RELT) and bit 1 (CMDT) of serial bus interface control register (SBIC). However, do not carry out this manipulation during serial transfer.

Control the $\overline{\text{SCK0}}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **16.4.5 $\overline{\text{SCK0}}$/P27 pin output manipulation**).

(3) Other signals

Figure 16-8 shows RELT and CMDT operations.

Figure 16-8. RELT and CMDT Operations



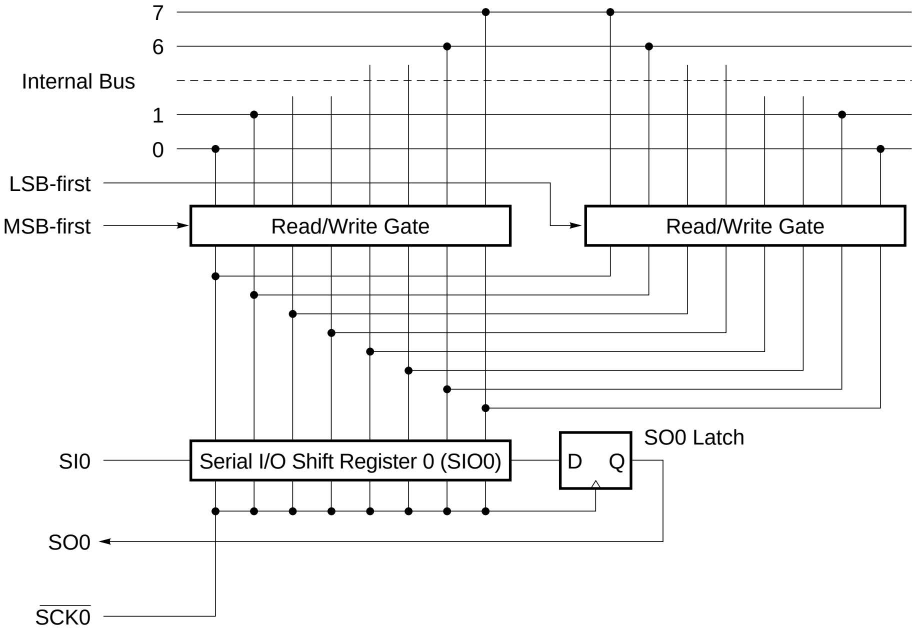
(4) MSB/LSB switching as the start bit

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

Figure 16-9 shows the configuration of the serial I/O shift register 0 (SIO0) and internal bus. As shown in the figure, MSB/LSB can be read/written in reverse form.

MSB/LSB switching as the start bit can be specified with bit 2 (CSIM02) of the serial operating mode register 0 (CSIM0).

Figure 16-9. Circuit of Switching in Transfer Bit Order



Start bit switching is realized by switching the bit order for data write to SIO0. The SIO0 shift order remains unchanged.

Thus, switching between MSB-first and LSB-first must be performed before writing data to SIO0.

(5) Transfer start

Serial transfer is started by setting transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied.

- Serial interface channel 0 operation control bit (CSIE0) = 1.
- Internal serial clock is stopped or $\overline{\text{SCK0}}$ is a high level after 8-bit serial transfer.

Caution If CSIE0 is set to “1” after data write to SIO0, transfer does not start.

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

16.4.3 SBI mode operation

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

SBI uses a single master device and employs the clocked serial I/O format with the addition of a bus configuration function. This function enables devices to communicate using only two lines. Thus, when making up a serial bus with two or more microcontrollers and peripheral ICs, the number of ports to be used and the number of wires on the board can be decreased.

The master device outputs three kinds of data to slave devices on the serial data bus: “addresses” to select a device to be communicated with, “commands” to instruct the selected device, and “data” which is actually required.

The slave device can identify the received data into “address”, “command”, or “data”, by hardware. An application program that controls serial interface channel 0 can be simplified by using this function.

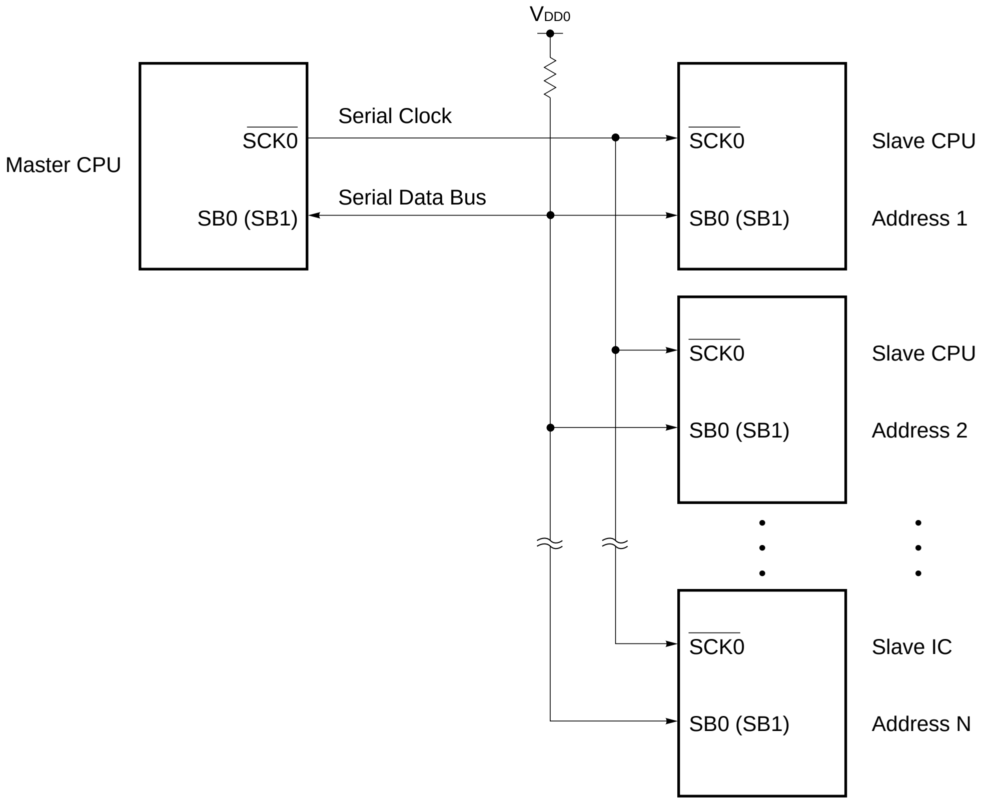
The SBI function is incorporated into various devices including 75X/XL Series and 78K Series.

Figure 16-10 shows a serial bus configuration example when a CPU having a serial interface compliant with SBI and peripheral ICs are used.

In SBI, the SB0 (SB1) serial data bus pin is an open-drain output pin and therefore the serial data bus line behaves in the same way as the wired-OR configuration. In addition, a pull-up resistor must be connected to the serial data bus line.

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

Figure 16-10. Example of Serial Bus Configuration with SBI



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

(1) SBI functions

In the conventional serial I/O format, when a serial bus is configured by connecting two or more devices, many ports and wiring are necessary, to provide chip select signal to identify command and data, and to judge the busy state, because only the data transfer function is available. If these operations are to be controlled by software, the software must be heavily loaded.

In the SBI, a serial bus can be configured with two signal lines of serial clock $\overline{\text{SCK0}}$ and serial data bus SB0 (SB1). Thus, use of SBI leads to reduction in the number of microcontroller ports and that of wirings and routings on the board.

The SBI functions are described below.

(a) Address/command/data identify function

Serial data is distinguished into addresses, commands, and data.

(b) Chip select function by address transmission

The master executes slave chip selection by address transmission.

(c) Wake-up function

The slave can easily judge address reception (chip select judgment) with the wake-up function (which can be set/reset by software).

When the wake-up function is set, the interrupt request signal (INTCSI0) is generated upon reception of a match address.

Thus, when communication is executed with two or more devices, the CPU except the selected slave devices can operate regardless of underway serial communications.

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

The acknowledge signal to check serial data reception is controlled.

(e) Busy signal ($\overline{\text{BUSY}}$) control function

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

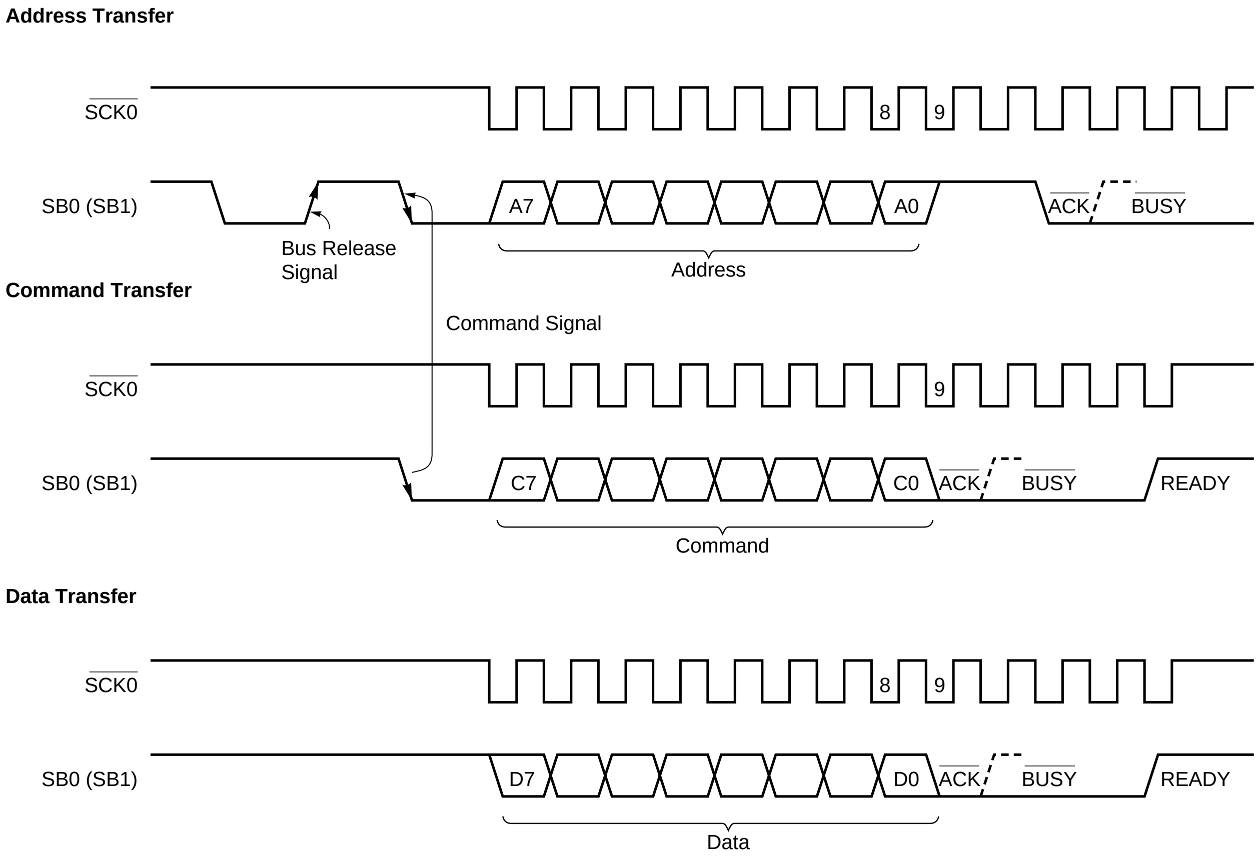
(2) SBI definition

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

Serial data to be transferred with SBI consists of three kinds of data: “address”, “command”, and “data”.

Figure 16-11 shows the address, command, and data transfer timings.

Figure 16-11. SBI Transfer Timings



Remark The broken line indicates READY status.

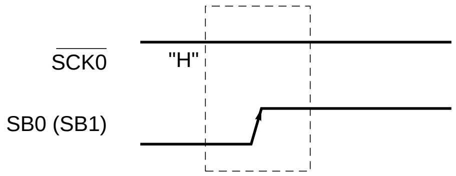
The bus release signal and the command signal are output by the master device. $\overline{\text{BUSY}}$ is output by the slave signal. $\overline{\text{ACK}}$ can be output by either the master or slave device (normally, the 8-bit data receiver outputs). Serial clocks continue to be output by the master device from 8-bit data transfer start to $\overline{\text{BUSY}}$ reset.

(a) Bus release signal (REL)

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

This signal is output by the master device.

Figure 16-12. Bus Release Signal



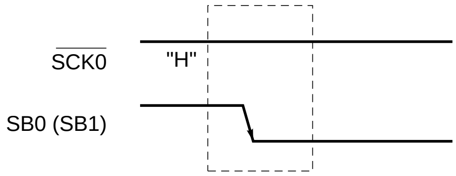
The bus release signal indicates that the master device is going to transmit an address to the slave device. The slave device incorporates hardware to detect the bus release signal.

★ **Caution The transition of the SB0 (SB1) line from low to high when the $\overline{\text{SCK0}}$ line is high is recognized as a bus release signal. If the transition timing of the bus is shifted due to the influence of board capacitance, transmitted data may be judged as a bus release signal. Exercise care in wiring so that noise is not superimposed on the signal lines.**

(b) Command signal (CMD)

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

Figure 16-13. Command Signal



The command signal indicates that the master is to transmit a command to the slave (however, the command signal following the bus release signal indicates that an address is transmitted).

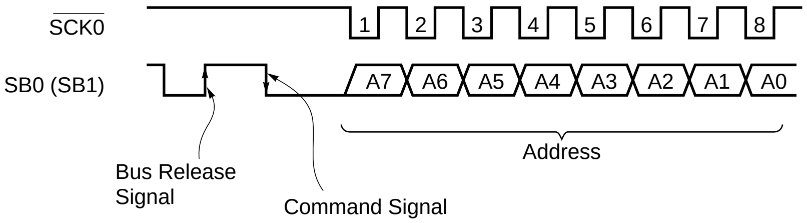
The slave device incorporates hardware to detect the command signal.

★ **Caution The transition of the SB0 (SB1) line from high to low when the $\overline{\text{SCK0}}$ line is high is recognized as a command signal. If the transition timing of the bus is shifted due to the influence of board capacitance, transmitted data may be judged as a command signal. Exercise care in wiring so that noise is not superimposed on the signal lines.**

(c) Address

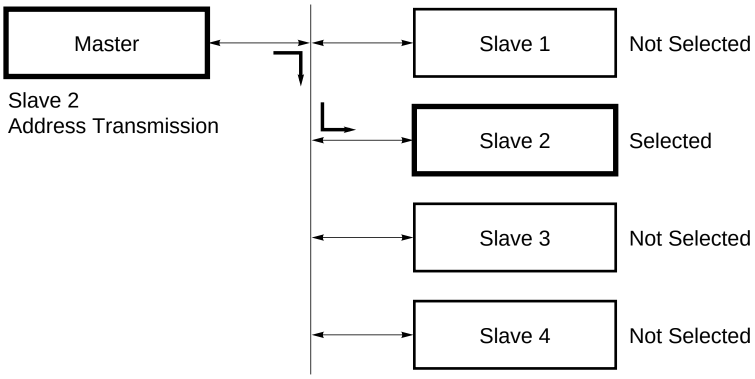
An address is 8-bit data which the master device outputs to the slave device connected to the bus line in order to select a particular slave device.

Figure 16-14. Addresses



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

Figure 16-15. Slave Selection with Address



(d) Command and data

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

Figure 16-16. Commands

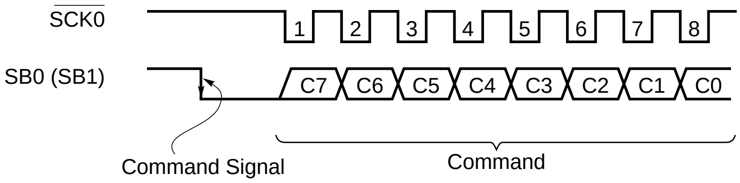
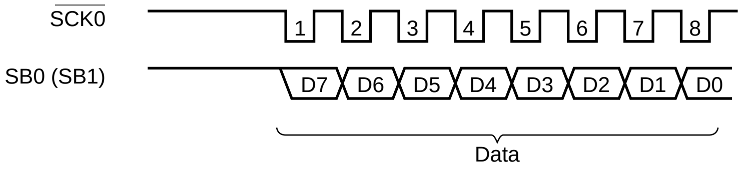


Figure 16-17. Data

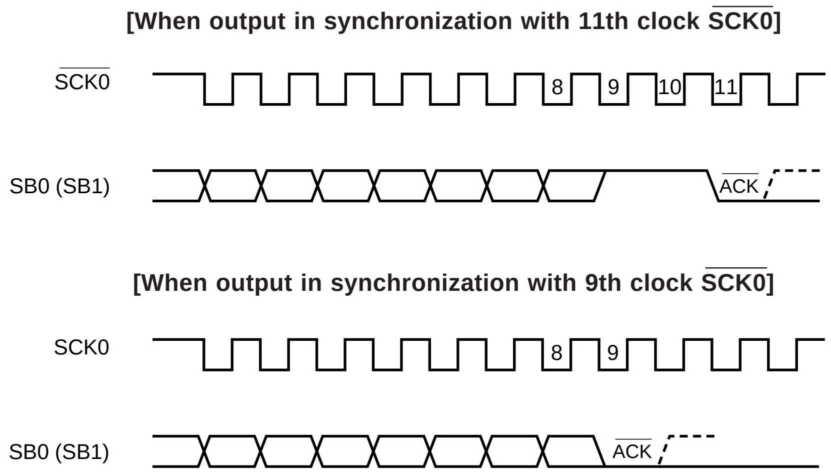


8-bit data following a command signal is defined as "command" data. 8-bit data without command signal is defined as "data". Command and data operation procedures are allowed to determine by user according to communications specifications.

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

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

Figure 16-18. Acknowledge Signal



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

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

Remark The broken line indicates READY status.

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

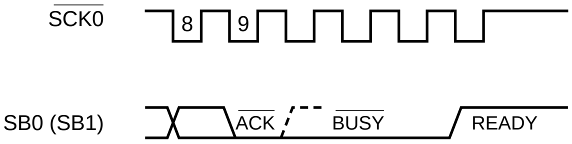
After 8-bit data transmission, the transmitter checks whether the receiver has returned the acknowledge signal. If the acknowledge signal is not returned for the preset period of time after data transmission, it can be judged that data reception has not been carried out correctly.

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

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

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

Figure 16-19. $\overline{\text{BUSY}}$ and READY Signals



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

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

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

Caution In the SBI mode, the $\overline{\text{BUSY}}$ signal is output until the next serial clock ($\overline{\text{SCK0}}$) falls after a command that reset the $\overline{\text{BUSY}}$ signal has been issued. If WUP is set to 1 during this period by mistake, the $\overline{\text{BUSY}}$ signal is not reset. Therefore, be sure to confirm that the SB0 (SB1) pin has gone high after resetting the $\overline{\text{BUSY}}$ signal, by setting WUP to 1.

(3) Register setting

The SBI mode is set with the serial operating mode register 0 (CSIM0), the serial bus interface control register (SBIC), and the interrupt timing specify register (SINT).

(a) Serial operating mode register 0 (CSIM0)

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM0 to 00H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W Note 1

R/W

CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection
0	×	Input clock to $\overline{\text{SCK0}}$ pin from off-chip
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operation Mode	Start Bit	SI0/SB0/P25 Pin Function	SO0/SB1/P26 Pin Function	$\overline{\text{SCK0}}$/P27 Pin Function
0	×	3-wire serial I/O mode (see **16.4.2 3-wire serial I/O mode operation**.)											
1	0	0	Note 2 ×	Note 2 ×	0	0	0	1	SBI mode	MSB	P25 (CMOS input/output)	SB1 (N-ch open-drain input/output)	$\overline{\text{SCK0}}$ (CMOS input/output)
		1	0	0	Note 2 ×	Note 2 ×	0	1			SB0 (N-ch open-drain input/output)	P26 (CMOS input/output)	
1	1	2-wire serial I/O mode (see **16.4.4 2-wire serial I/O mode operation**.)											

R/W

WUP	Wake-up Function Control Note 3
0	Interrupt request signal generation with each serial transfer in any mode
1	Interrupt request signal generation when the address received after bus release (when CMDD = RELD = 1) matches the slave address register (SVA) data in SBI mode

R

COI	Slave Address Comparison Result Flag Note 4
0	Slave address register (SVA) not equal to serial I/O shift register 0 (SIO0) data
1	Slave address register (SVA) equal to serial I/O shift register 0 (SIO0) data

R/W

CSIE0	Serial Interface Channel 0 Operation Control
0	Operation stopped
1	Operation enabled

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

2. Can be used as a port function.

3. When using the wake-up function (WUP = 1), clear the bit 5 (SIC) of the interrupt timing specify register (SINT) to 0.

4. When CSIE0 = 0, COI becomes 0.

Remark × : don't care

PM×× : Port mode register

P×× : Port output latch

(b) Serial bus interface control register (SBIC)

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

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

The shaded area is used in the SBI mode.

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

R/W

RELT	Used for bus release signal output. When RELT = 1, SO0 latch is set to (1). After SO0 latch setting, automatically cleared to (0). Also cleared to 0 when CSIE0 = 0.

R/W

CMDT	Used for command signal output. When CMDT = 1, SO0 latch is cleared to (0). After SO0 latch clearance, automatically cleared to (0). Also cleared to 0 when CSIE0 = 0.

R

RELD	Bus Release Detection
Clear Conditions (RELD = 0)	Set Conditions (RELD = 1)
• When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception (only when WUP = 1) • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied	• When bus release signal (REL) is detected

R

CMDD	Command Detection
Clear Conditions (CMDD = 0)	Set Conditions (CMDD = 1)
• When transfer start instruction is executed • When bus release signal (REL) is detected • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied	• When command signal (CMD) is detected

R/W

ACKT	Acknowledge signal is output in synchronization with the falling edge of $\overline{\text{SCK0}}$ clock just after execution of the instruction that sets this bit to 1 and, after acknowledge signal output, automatically cleared to (0). Used as ACKE = 0. Also cleared to (0) upon start of serial interface transfer or when CSIE0 = 0.

R/W

ACKE	Acknowledge Signal Automatic Output Control	
0	Acknowledge signal automatic output disable (output with ACKT enable)	
1	Before completion of transfer	Acknowledge signal is output in synchronization with the falling edge of the 9th $\overline{\text{SCK0}}$ clock (automatically output when ACKE = 1).
	After completion of transfer	Acknowledge signal is output in synchronization with falling edge of $\overline{\text{SCK0}}$ clock just after execution of the instruction that sets this bit to 1 (automatically output when ACKE = 1). However, not automatically cleared to 0 after acknowledge signal output.

(Cont'd)

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

Remarks 1. Bits 0, 1, and 4 (RELT, CMDT, and ACKT) are 0 when read after data setting.

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

R

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

R/W

BSYE[Note]	Synchronizing Busy Signal Output Control
0	Disables busy signal which is output in synchronization with the falling edge of $\overline{\text{SCK0}}$ clock just after execution of the instruction that clears this bit to 0 (sets READY status).
1	Outputs busy signal at the falling edge of $\overline{\text{SCK0}}$ clock following the acknowledge signal.

★ **Note** Busy mode can be cleared by start of serial interface transfer. However, BSYE flag is not cleared to 0.

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

(c) Interrupt timing specify register (SINT)

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

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

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

R/W

SVAM	SVA Bit to be Used as Slave Address
0	Bits 0 to 7
1	Bits 1 to 7

R/W

SIC	INTCSI0 Interrupt Source Selection Note 2
0	CSIIF0 is set upon termination of serial interface channel 0 transfer
1	CSIIF0 is set upon bus release detection or termination of serial interface channel 0 transfer

R

CLD	$\overline{\text{SCK0}}$/P27 Pin Level Note 3
0	Low level
1	High level

Caution Be sure to set bits 0 to 3 to 0.

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

2. When using wake-up function in the SBI mode, set SIC to 0.

3. When CSIE0 = 0, CLD becomes 0.

Remark SVA : Slave address register

CSIIF0 : Interrupt request flag corresponding to INTCSI0

CSIE0 : Bit 7 of serial operating mode register 0 (CSIM0)

(4) Various signals

Figures 16-20 to 16-25 show various signals and flag operations in SBI. Table 16-3 lists various signals in SBI.

Figure 16-20. RELT, CMDT, RELD, and CMDD Operations (Master)

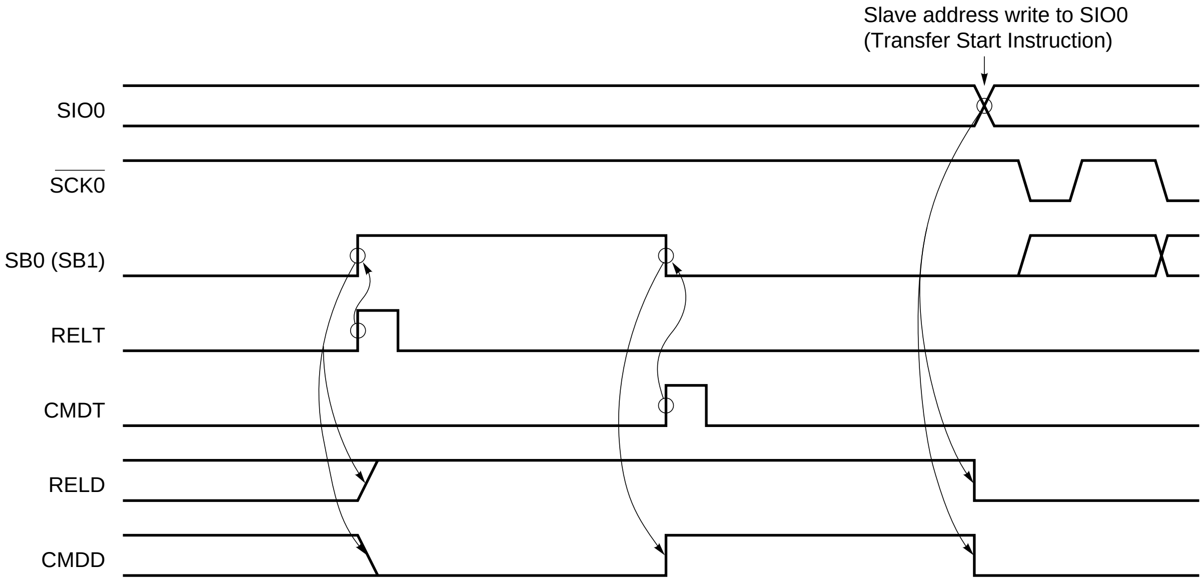


Figure 16-21. RELD and CMDD Operations (Slave)

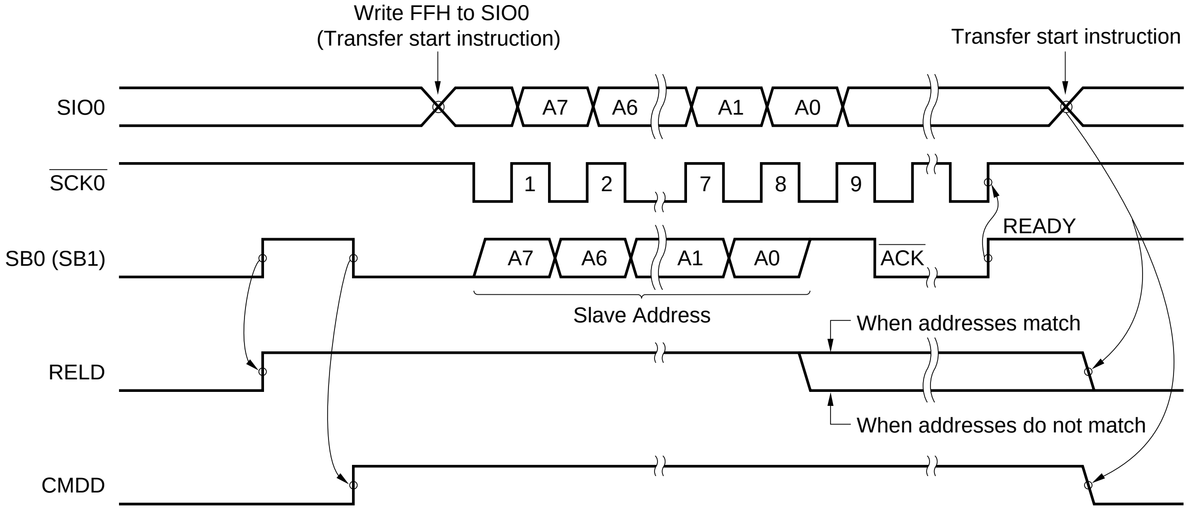
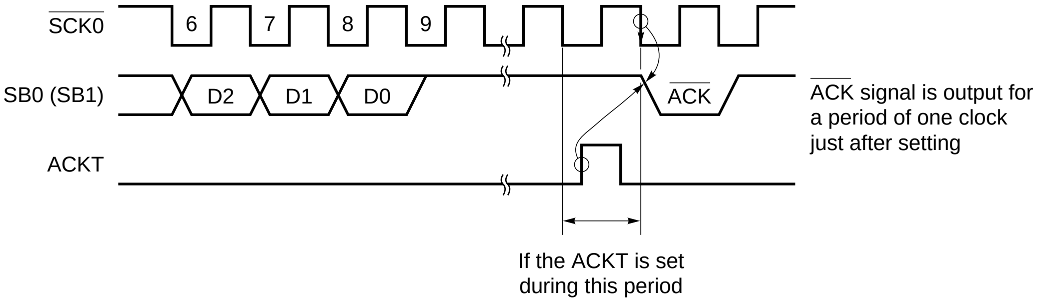
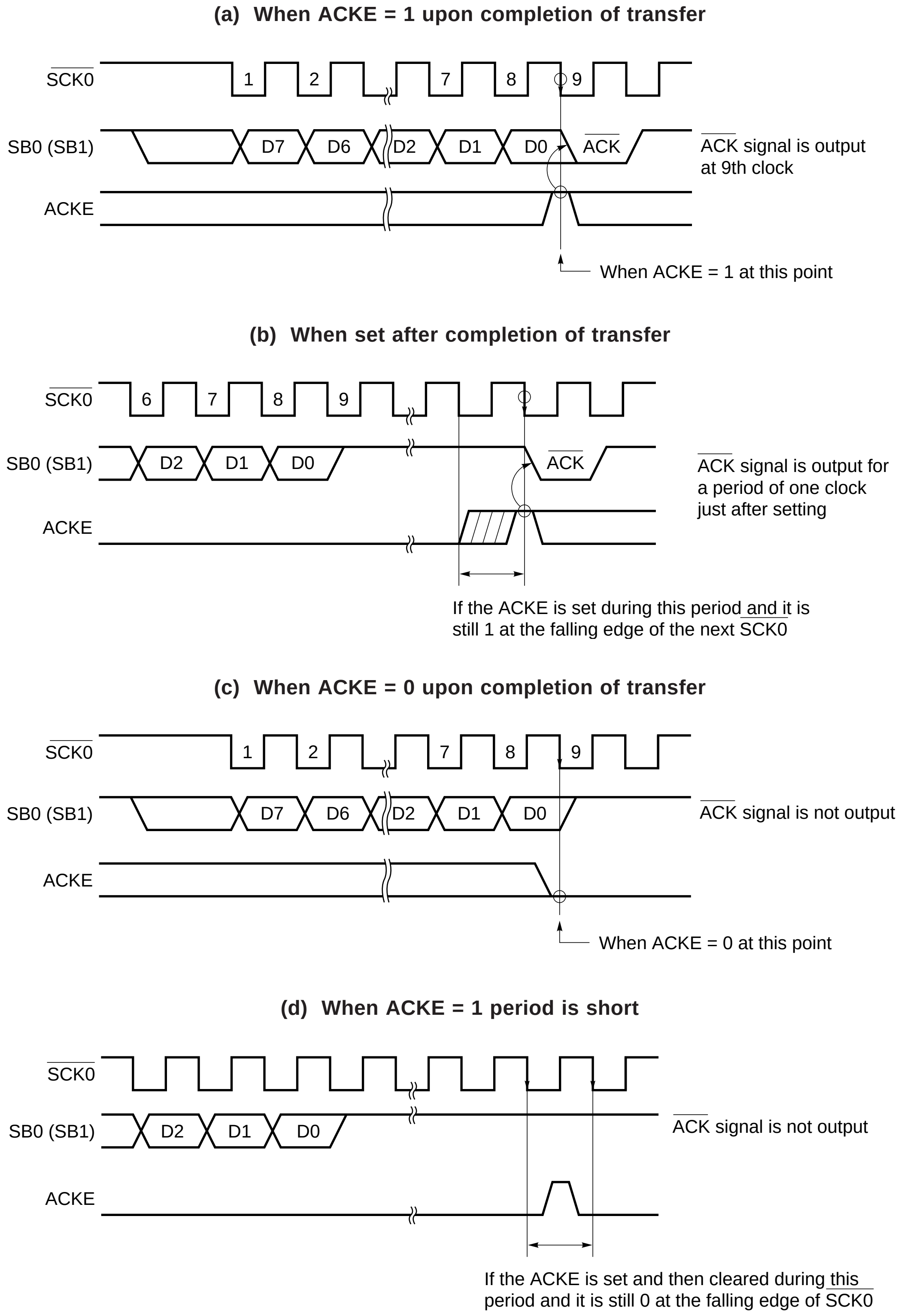


Figure 16-22. ACKT Operation


SCK0
6
7
8
9
SB0 (SB1)
D2
D1
D0
ACK
ACKT
ACK signal is output for a period of one clock just after setting
If the ACKT is set during this period

Caution Do not set ACKT before completion of transfer.

Figure 16-23. ACKE Operations



(a) When ACKE = 1 upon completion of transfer

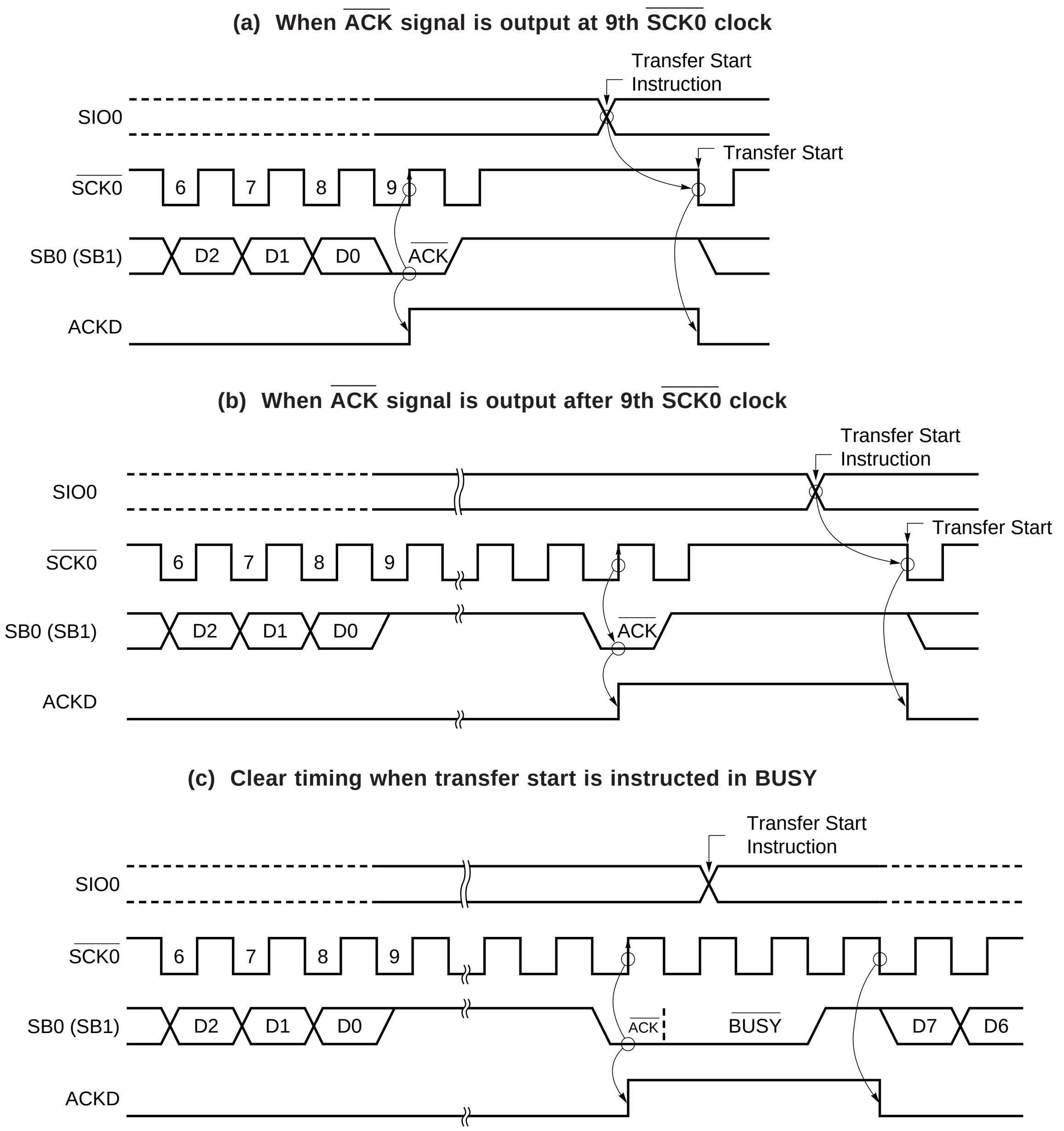
(b) When set after completion of transfer

(c) When ACKE = 0 upon completion of transfer

(d) When ACKE = 1 period is short

SCK0
1
2
7
8
9
SB0 (SB1)
D7
D6
D2
D1
D0
ACK
ACK signal is output at 9th clock
ACKE
When ACKE = 1 at this point
SCK0
6
7
8
9
SB0 (SB1)
D2
D1
D0
ACK
ACK signal is output for a period of one clock just after setting
ACKE
If the ACKE is set during this period and it is still 1 at the falling edge of the next SCK0
SCK0
1
2
7
8
9
SB0 (SB1)
D7
D6
D2
D1
D0
ACK signal is not output
ACKE
When ACKE = 0 at this point
SCK0
SB0 (SB1)
D2
D1
D0
ACK signal is not output
ACKE
If the ACKE is set and then cleared during this period and it is still 0 at the falling edge of SCK0

Figure 16-24. ACKD Operations

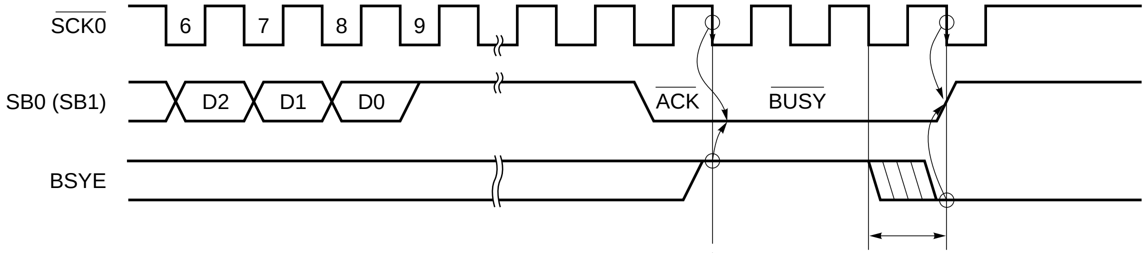


(a) When $\overline{\text{ACK}}$ signal is output at 9th $\overline{\text{SCK0}}$ clock

(b) When $\overline{\text{ACK}}$ signal is output after 9th $\overline{\text{SCK0}}$ clock

(c) Clear timing when transfer start is instructed in BUSY

Figure 16-25. BSYE Operation



When BSYE = 1 at this point

If the BSYE is reset during this period and it is still 0 at the falling edge of $\overline{\text{SCK0}}$

Table 16-3. Various Signals in SBI Mode (1/2)

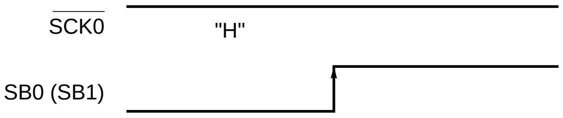
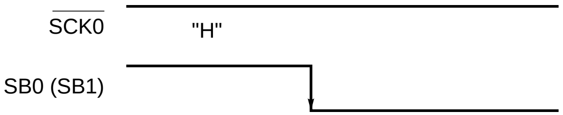
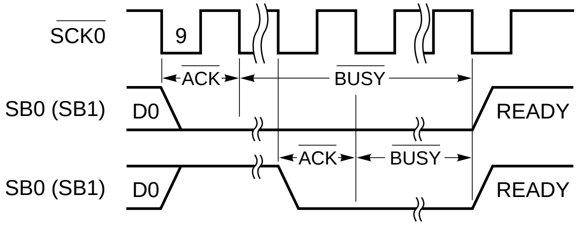
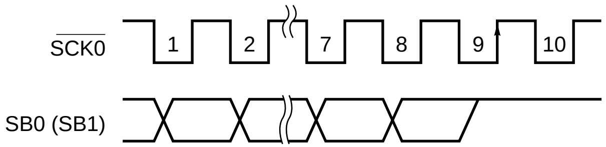
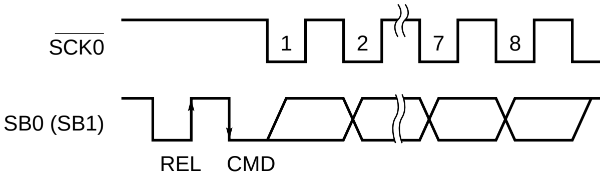
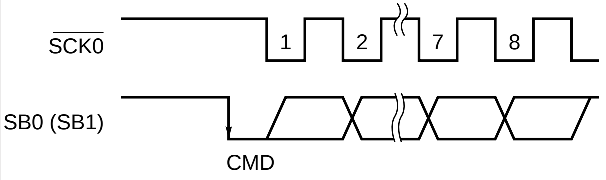
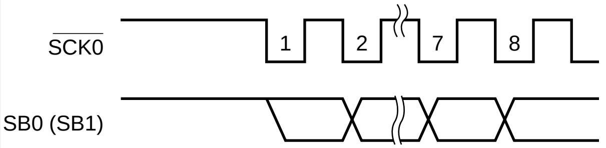
Signal Name	Output Device	Definition	Timing Chart	Output Condition	Effects on Flag	Meaning of Signal
Bus release signal (REL)	Master	SB0 (SB1) rising edge when $\overline{\text{SCK0}}$ = 1		• RELT set	• RELD set • CMDD clear	CMD signal is output to indicate that transmit data is an address.
Command signal (CMD)	Master	SB0 (SB1) falling edge when $\overline{\text{SCK0}}$ = 1		• CMDT set	• CMDD set	i) Transmit data is an address after REL signal output. ii) REL signal is not output and transmit data is an command.
Acknowledge signal ($\overline{\text{ACK}}$)	Master/ slave	Low-level signal to be output to SB0 (SB1) during one-clock period of $\overline{\text{SCK0}}$ after completion of serial reception	[Synchronous BUSY output]  $\overline{\text{SCK0}}$ 9 $\overline{\text{ACK}}$ BUSY SB0 (SB1) D0 READY $\overline{\text{ACK}}$ BUSY SB0 (SB1) D0 READY	①ACKE = 1 ②ACKT set	• ACKD set	Completion of reception
Busy signal ($\overline{\text{BUSY}}$)	Slave	[Synchronous BUSY signal] Low-level signal to be output to SB0 (SB1) following Acknowledge signal		• BSYE = 1	—	Serial receive disable because of processing
Ready signal (READY)	Slave	High-level signal to be output to SB0 (SB1) before serial transfer start and after completion of serial transfer		①BSYE = 0 ②Execution of instruction for data write to SIO0 (transfer start instruction)	—	Serial receive enable

Table 16-3. Various Signals in SBI Mode (2/2)

Signal Name	Output Device	Definition	Timing Chart	Output Condition	Effects on Flag	Meaning of Signal
Serial clock ($\overline{\text{SCK0}}$)	Master	Synchronous clock to output address/command/ data, $\overline{\text{ACK}}$ signal, synchronous $\overline{\text{BUSY}}$ signal, etc. Address/command/data are transferred with the first eight synchronous clocks.	 $\overline{\text{SCK0}}$ 1 2 7 8 9 10 SB0 (SB1)	When CSIE0 = 1, execution of instruction for data write to SIO0 (serial transfer start instruction) Note 2	CSIIF0 set (rising edge of 9th clock of $\overline{\text{SCK0}}$) Note 1	Timing of signal output to serial data bus
Address (A7 to A0)	Master	8-bit data to be transferred in synchronization with $\overline{\text{SCK0}}$ after output of REL and CMD signals	 $\overline{\text{SCK0}}$ 1 2 7 8 SB0 (SB1) REL CMD			Address value of slave device on the serial bus
Commands (C7 to C0)	Master	8-bit data to be transferred in synchronization with $\overline{\text{SCK0}}$ after output of only CMD signal without REL signal output	 $\overline{\text{SCK0}}$ 1 2 7 8 SB0 (SB1) CMD			Instructions and messages to the slave device
Data (D7 to D0)	Master/ slave	8-bit data to be transferred in synchronization with $\overline{\text{SCK0}}$ without output of REL and CMD signals				Numeric values to be processed with slave or master device

Notes 1. When WUP = 0, CSIIF0 is set at the rising edge of the 9th clock of $\overline{\text{SCK0}}$.
When WUP = 1, an address is received. Only when the address matches the slave address register (SVA) value, CSIIF0 is set (if the address does not coincide with the value of SVA, RELD is cleared).

2. In $\overline{\text{BUSY}}$ state, transfer starts after the READY state is set.

(5) Pin configuration

The serial clock pin $\overline{\text{SCK0}}$ and serial data bus pin SB0 (SB1) have the following configurations.

(a) $\overline{\text{SCK0}}$ .... Serial clock input/output pin

① Master ... CMOS and push-pull output

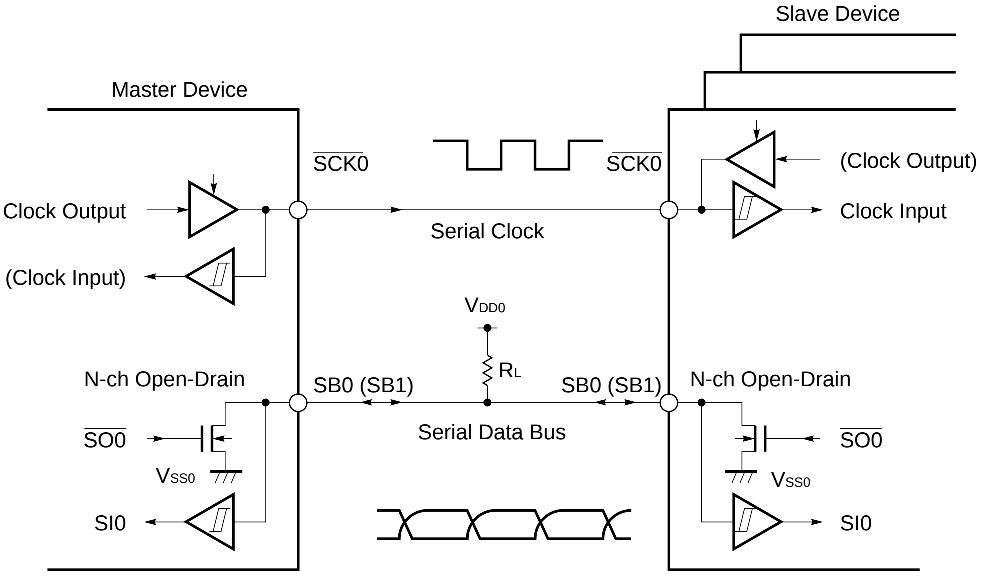
② Slave...... Schmitt input

(b) SB0 (SB1) .... Serial data input/output alternate function pin

Both master and slave devices have an N-ch open drain output and a Schmitt input.

Because the serial data bus line has an N-ch open-drain output, an external pull-up resistor is necessary.

Figure 16-26. Pin Configuration



Caution Because the N-ch open-drain output must made to go into a high-impedance state during data reception, write FFH to serial I/O shift register 0 (SIO0) in advance. The N-ch open-drain output can always go into a high-impedance state during transfer. However, when the wake-up function specify bit (WUP) = 1, the N-ch open-drain output always goes into a high-impedance state. Thus, it is not necessary to write FFH to SIO0 before reception.

(6) Address match detection method

In the SBI mode, the master transmits a slave address to select a specific slave device.

Coincidence of the addresses can be automatically detected by hardware. CSIIF0 is set only when the slave address transmitted by the master coincides with the address set to SVA when the wake-up function specify bit (WUP) = 1.

If the bit 5 (SIC) of the interrupt timing specify register (SINT) is set, the wake-up function cannot be used even if WUP is set (an interrupt request signal is generated when bus release is detected). To use the wake-up function, clear the SIC to 0.

Cautions 1. Slave selection/non-selection is detected by matching of the slave address received after bus release (RELD = 1).

For this match detection, match interrupt request (INTCSI0) of the address to be generated with WUP = 1 is normally used. Thus, execute selection/non-selection detection by slave address when WUP = 1.

2. When detecting selection/non-selection without the use of interrupt request with WUP = 0, do so by means of transmission/reception of the command preset by program instead of using the address match detection method.

(7) Error detection

In the SBI mode, the serial bus SB0 (SB1) status being transmitted is fetched into the destination device, that is, the serial I/O shift register 0 (SIO0). Thus, transmit errors can be detected in the following way.

(a) Method of comparing SIO0 data before transmission to that after transmission

In this case, if two data differ from each other, a transmit error is judged to have occurred.

(b) Method of using the slave address register (SVA)

Transmit data is set to both SIO0 and SVA and is transmitted. After termination of transmission, COI bit (match signal coming from the address comparator) of the serial operating mode register 0 (CSIM0) is tested. If “1”, normal transmission is judged to have been carried out. If “0”, a transmit error is judged to have occurred.

(8) Communication operation

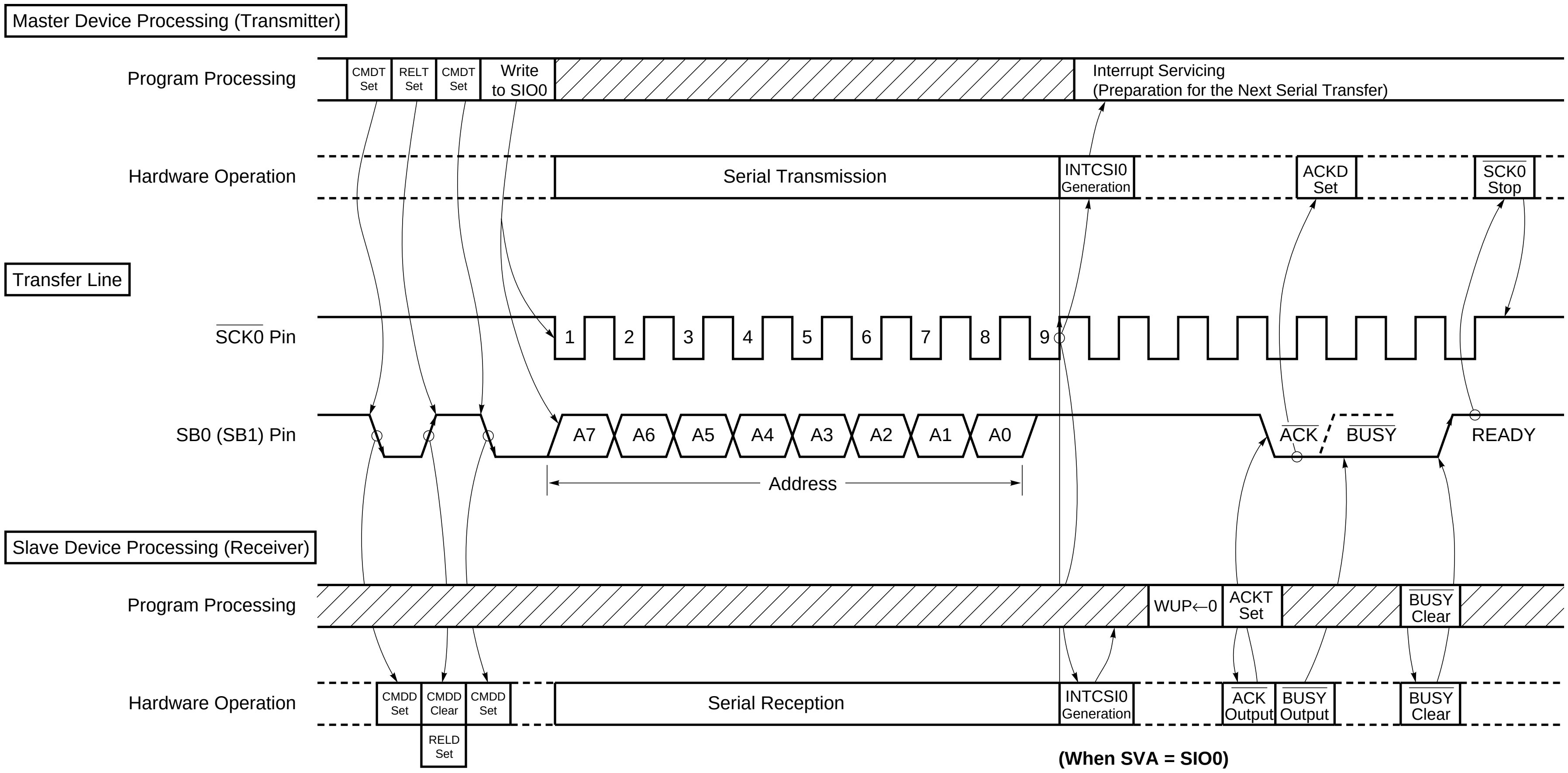
In the SBI mode, the master device selects normally one slave device as communication target from among two or more devices by outputting an “address” to the serial bus.

After the communication target device has been determined, commands and data are transmitted/received and serial communication is realized between the master and slave devices.

Figures 16-27 to 16-30 show data communication timing charts.

Shift operation of the serial I/O shift register 0 (SIO0) is carried out at the falling edge of serial clock ($\overline{\text{SCK0}}$). Transmit data is latched into the SO0 latch and is output with MSB set as the first bit from the SB0/P25 or SB1/P26 pin. Receive data input to the SB0 (or SB1) pin at the rising edge of $\overline{\text{SCK0}}$ is latched into the SIO0.

Figure 16-27. Address Transmission from Master Device to Slave Device (WUP = 1)



(When SVA = SIO0)

Figure 16-28. Command Transmission from Master Device to Slave Device

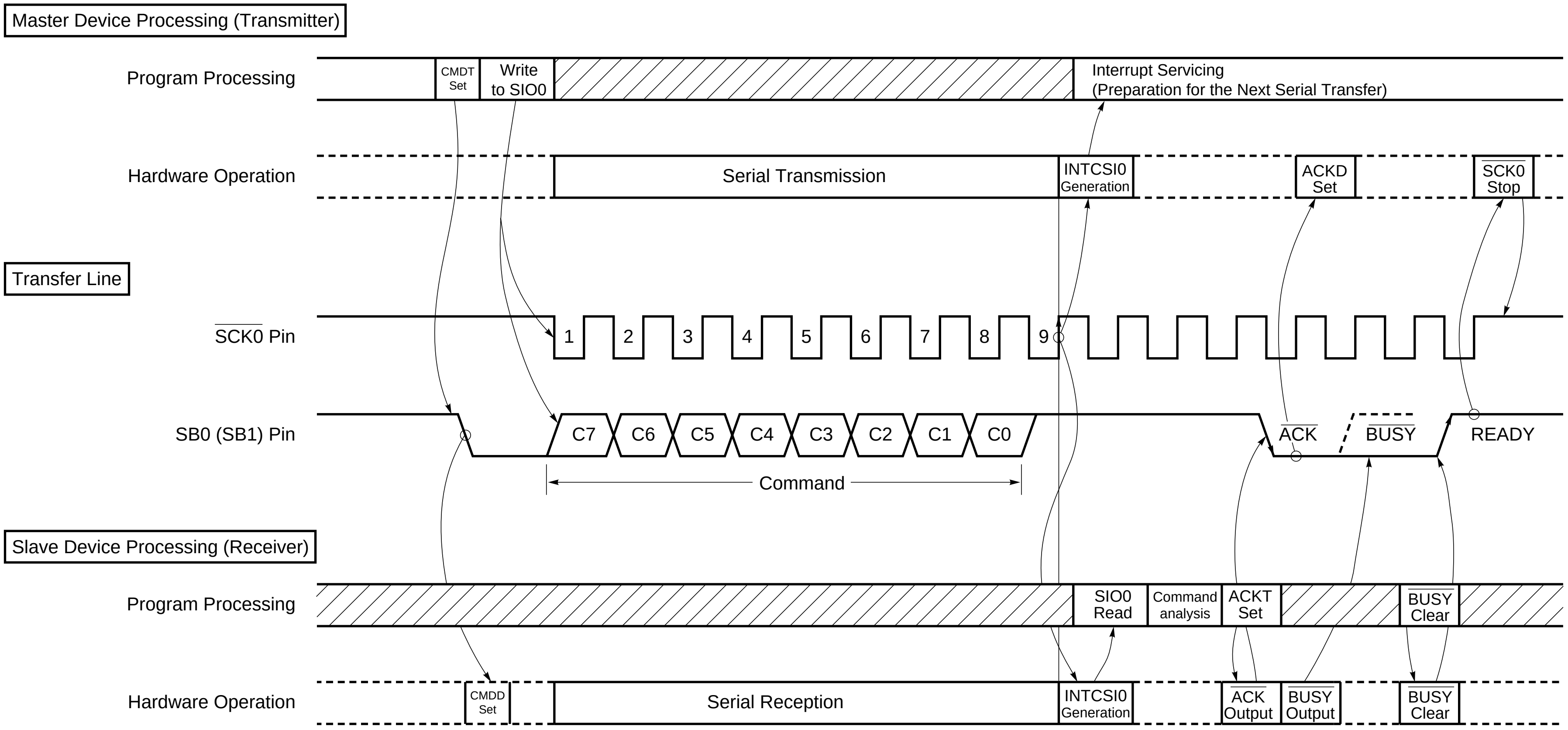


Figure 16-29. Data Transmission from Master Device to Slave Device

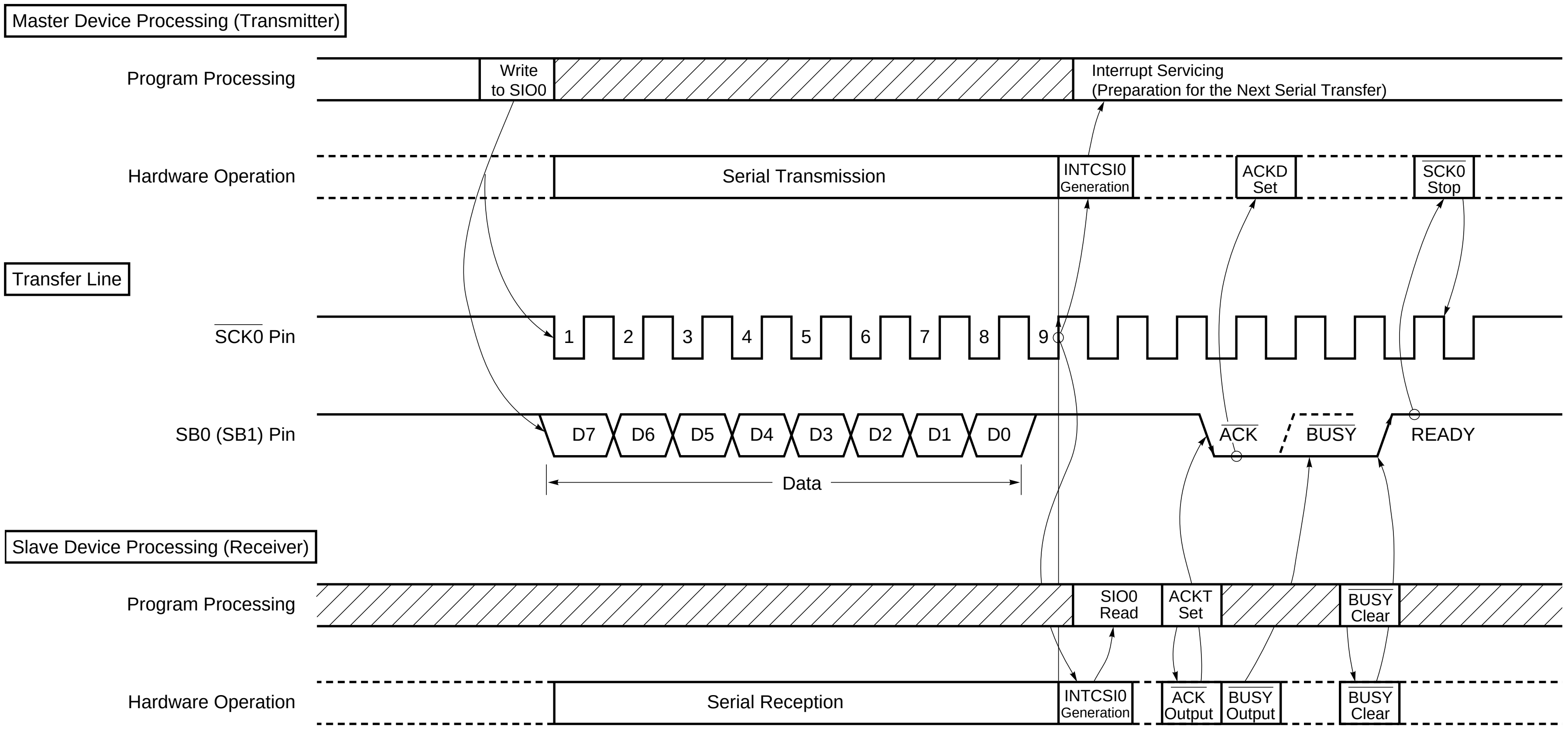
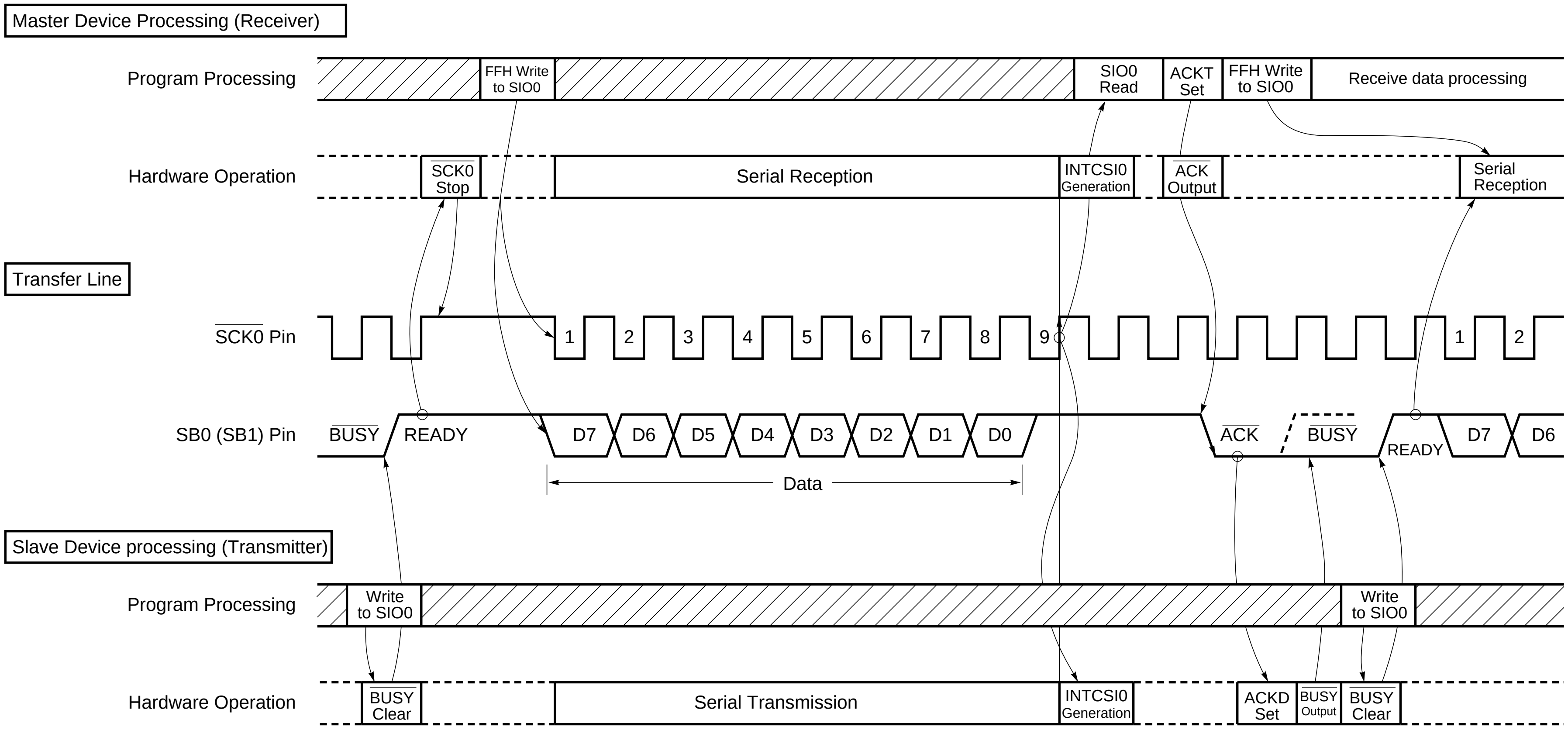


Figure 16-30. Data Transmission from Slave Device to Master Device



(9) Transfer start

Serial transfer is started by setting transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied.

- Serial interface channel 0 operation control bit (CSIE0) = 1
- Internal serial clock is stopped or $\overline{\text{SCK0}}$ is at high level after 8-bit serial transfer.

Cautions 1. If CSIE0 is set to “1” after data write to SIO0, transfer does not start.

2. Because the N-ch open-drain output must go into a high-impedance state during data reception, write FFH to SIO0 in advance.
However, when the make-up function specify bit (WUP) = 1, the N-ch open-drain output always goes into a high-impedance state. Thus, it is not necessary to write FFH to SIO0 before reception.

3. If data is written to SIO0 when the slave is busy, the data is not lost.
When the busy state is cleared and SB0 (or SB1) input is set to the high level (READY) state, transfer starts.

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (CSIIF0) is set.
For pins that are to be used for data input/output, be sure to carry out the following settings before serial transfer of the 1st byte after $\overline{\text{RESET}}$ input.

<1> Set the P25 and P26 output latches to 1.
<2> Set bit 0 (RELT) of the serial bus interface control register (SBIC) to 1.
<3> Reset the P25 and P26 output latches from 1 to 0.

(10) Judging busy state of slave

When the device is in the master mode, follow the procedure below to judge whether slave device is in the busy state or not.

<1> Detect acknowledge signal ($\overline{\text{ACK}}$) or interrupt request signal generation.
<2> Set the port mode register PM25 (or PM26) of the SB0/P25 (or SB1/P26) pin into the input mode.
<3> Read out the pin state (when the pin level is high, the READY state is set).

After the detection of the READY state, set the port mode register to 0 and return to the output mode.

(11) SBI mode precautions

(a) Slave selection/non-selection is detected by match detection of the slave address received after bus release (RELD = 1).
For this match detection, match interrupt request (INTCSI0) of the address to be generated with WUP = 1 is normally used. Thus, execute selection/non-selection detection by slave address when WUP = 1.

(b) When detecting selection/non-selection without the use of interrupt with WUP = 0, do so by means of transmission/reception of the command preset by program instead of using the address match detection method.

(c) In the SBI mode, the $\overline{\text{BUSY}}$ signal is output until the next serial clock falls after a command that resets the $\overline{\text{BUSY}}$ signal has been issued. If WUP is set to 1 during this period by mistake, the $\overline{\text{BUSY}}$ signal is not reset. Therefore, be sure to confirm that the SB0 (SB1) pin has gone high after resetting the $\overline{\text{BUSY}}$ signal, by setting WUP to 1.

(d) For pins that are to be used for data input/output, be sure to carry out the following settings before serial transfer of the 1st byte after $\overline{\text{RESET}}$ input.

<1> Set the P25 and P26 output latches to 1.
<2> Set bit 0 (RELT) of the serial bus interface control register (SBIC) to 1.
<3> Reset the P25 and P26 output latches from 1 to 0.

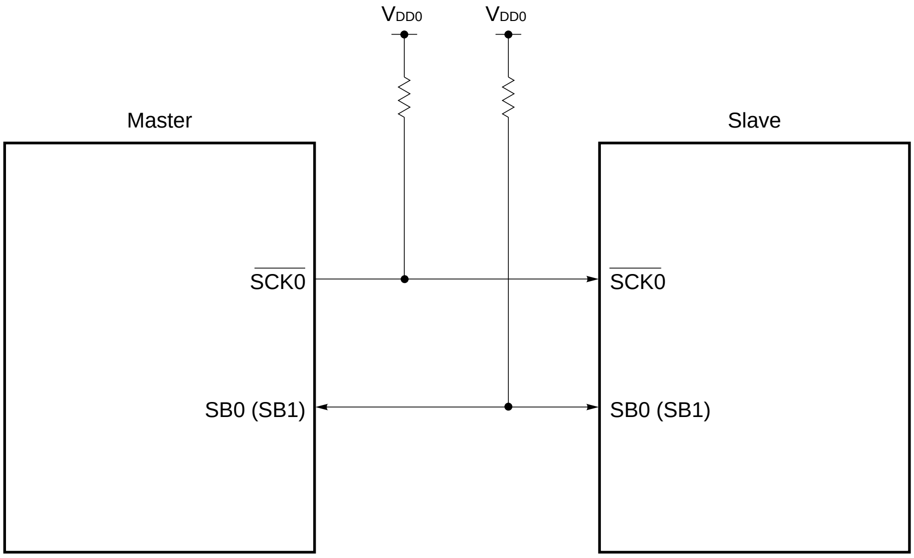
★ (e) The transition of the SB0 (SB1) line from low to high or from high to low when the $\overline{\text{SCK0}}$ line is high is recognized as a bus release signal or a command signal, respectively. If the transition timing of the bus is shifted due to the influence of board capacitance, transmitted data may be judged as a bus release signal (or a command signal). Exercise care in wiring so that noise is not superimposed on the signal lines.

16.4.4 2-wire serial I/O mode operation

The 2-wire serial I/O mode can cope with any communication format by program.

Communication is basically carried out with two lines of serial clock ($\overline{\text{SCK0}}$) and serial data input/output (SB0 or SB1).

Figure 16-31. Serial Bus Configuration Example Using 2-wire Serial I/O Mode



(1) Register setting

The 2-wire serial I/O mode is set with the serial operating mode register 0 (CSIM0), the serial bus interface control register (SBIC), and the interrupt timing specify register (SINT).

(a) Serial operating mode register 0 (CSIM0)

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM0 to 00H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W Note 1

R/W

CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection
0	×	Input clock to $\overline{\text{SCK0}}$ pin from off-chip
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operation Mode	Start Bit	SIO/SB0/P25 Pin Function	SO0/SB1/P26 Pin Function	$\overline{\text{SCK0}}$/P27 Pin Function
0	×	3-wire Serial I/O mode (see **16.4.2 3-wire serial I/O mode operation**)											
1	0	SBI mode (see **16.4.3 SBI mode operation**)											
1	1	0	Note 2 ×	Note 2 ×	0	0	0	1	2-wire serial I/O mode	MSB	P25 (CMOS input/output)	SB1 (N-ch open-drain input/output)	$\overline{\text{SCK0}}$ (N-ch open-drain input/output)
		1	0	0	Note 2 ×	Note 2 ×	0	1			SB0 (N-ch open-drain input/output)	P26 (CMOS input/output)	

R/W

WUP	Wake-up Function Control Note 3
0	Interrupt request signal generation with each serial transfer in any mode
1	Interrupt request signal generation when the address received after bus release (when CMDD = RELD = 1) matches the slave address register (SVA) data in SBI mode

R

COI	Slave Address Comparison Result Flag Note 4
0	Slave address register (SVA) not equal to serial I/O shift register 0 (SIO0) data
1	Slave address register (SVA) equal to serial I/O shift register 0 (SIO0) data

R/W

CSIE0	Serial Interface Channel 0 Operation Control
0	Operation stopped
1	Operation enabled

Notes **1.** Bit 6 (COI) is a read-only bit.

2. Can be used freely as port function.

3. Be sure to set WUP to 0 when the 2-wire serial I/O mode.

4. When CSIE0 = 0, COI becomes 0.

Remark		
×	:	don’t care
PM××	:	Port mode register
P××	:	Port output latch

(b) Serial bus interface control register (SBIC)

SBIC is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears SBIC to 00H.

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W

R/W		
	RELT	When RELT = 1, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W		
	CMDT	When CMDT = 1, SO0 latch is cleared to 0. After SO0 latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

CSIE0: Bit 7 of serial operating mode register 0 (CSIM0)

(c) Interrupt timing specify register (SINT)

SINT is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears SINT to 00H.

Symbol	7	⑥	⑤	④	3	2	1	0	Address	After Reset	R/W
SINT	0	CLD	SIC	SVAM	0	0	0	0	FF63H	00H	R/W Note 1

R/W

SIC	INTCSI0 Interrupt Factor Selection
0	CSIIF0 is set upon termination of serial interface channel 0 transfer
1	CSIIF0 is set upon bus release detection or termination of serial interface channel 0 transfer

R

CLD	$\overline{\text{SCK0}}$/P27 Pin Level Note 2
0	Low level
1	High level

Caution Be sure to set bits 0 to 3 to 0.

Notes 1. Bit 6 (CLD) is a read-only bit.

2. When CSIE0 = 0, CLD becomes 0.

Remark CSIIF0 : Interrupt request flag corresponding to INTCSI0

CSIE0 : Bit 7 of serial operating mode register 0 (CSIM0)

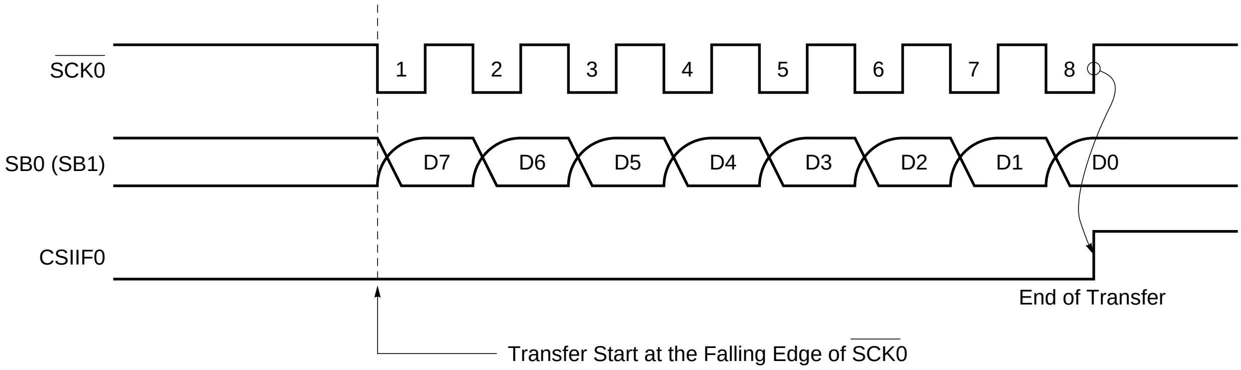
(2) Communication operation

The 2-wire serial I/O mode is used for data transmission/reception in 8-bit units. Data transmission/reception is carried out bit-wise in synchronization with the serial clock.

Shift operation of the serial I/O shift register 0 (SIO0) is carried out in synchronization with the falling edge of the serial clock ($\overline{\text{SCK0}}$). The transmit data is held in the SO0 latch and is output from the SB0/P25 (or SB1/P26) pin on an MSB-first basis. The receive data input from the SB0 (or SB1) pin is latched into the SIO0 at the rising edge of $\overline{\text{SCK0}}$.

Upon termination of 8-bit transfer, the SIO0 operation stops automatically and the interrupt request flag (CSIIF0) is set.

Figure 16-32. 2-wire Serial I/O Mode Timings



The SB0 (or SB1) pin specified for the serial data bus is an N-ch open-drain input/output and thus it must be externally connected to a pull-up resistor. Because N-ch open-drain output must go into a high-impedance state during data reception, write FFH to SIO0 in advance.

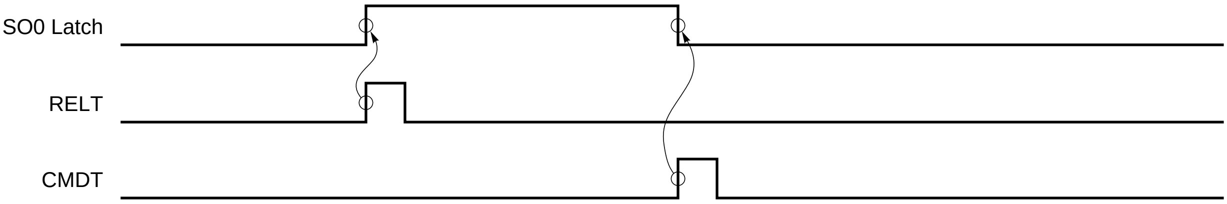
The SB0 (or SB1) pin generates the SO0 latch status and thus the SB0 (or SB1) pin output status can be manipulated by setting bit 0 (RELT) and bit 1 (CMDT) of serial bus interface control register (SBIC). However, do not carry out this manipulation during serial transfer.

Control the $\overline{\text{SCK0}}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **16.4.5 $\overline{\text{SCK0}}$/P27 pin output manipulation**).

(3) Other signals

Figure 16-33 shows RELT and CMDT operations.

Figure 16-33. RELT and CMDT Operations



(4) Transfer start

Serial transfer is started by setting transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied.

- Serial interface channel 0 operation control bit (CSIE0) = 1
- Internal serial clock is stopped or $\overline{\text{SCK0}}$ is at high level after 8-bit serial transfer.

Cautions 1. If CSIE0 is set to "1" after data write to SIO0, transfer does not start.

2. Because the N-ch open-drain output must go into a high-impedance state during data reception, write FFH to SIO0 in advance.

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (CSIIF0) is set.

(5) Error detection

In the 2-wire serial I/O mode, the serial bus SB0 (SB1) status being transmitted is fetched into the destination device, that is, serial I/O shift register 0 (SIO0). Thus, transmit error can be detected in the following way.

(a) Method of comparing SIO0 data before transmission to that after transmission

In this case, if two data differ from each other, a transmit error is judged to have occurred.

(b) Method of using the slave address register (SVA)

Transmit data is set to both SIO0 and SVA and is transmitted. After termination of transmission, COI bit (match signal coming from the address comparator) of the serial operating mode register 0 (CSIM0) is tested. If "1", normal transmission is judged to have been carried out. If "0", a transmit error is judged to have occurred.

16.4.5 SCK0/P27 pin output manipulation

Because the SCK0/P27 pin incorporates an output latch, static output is also possible by software in addition to normal serial clock output.

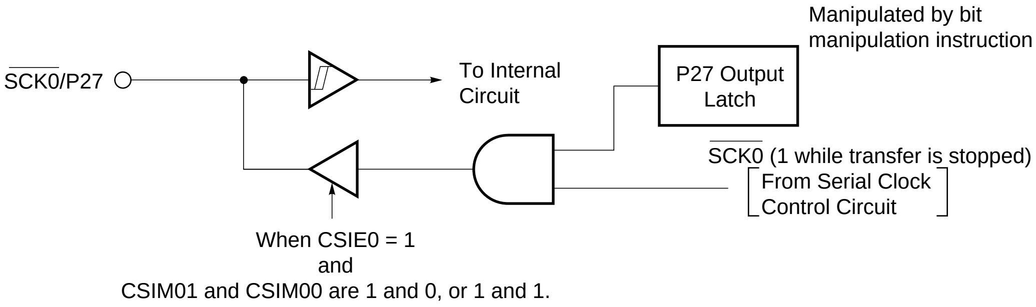
P27 output latch manipulation enables any value of SCK0 to be set by software. (SI0/SB0 and SO0/SB1 pin to be controlled with the bits 0 and 1 (RELT and CMDT) of serial bus interface control register (SBIC).)

SCK0/P27 pin output manipulating procedure is described below.

① Set the serial operating mode register 0 (CSIM0) (SCK0 pin enabled for serial operation in the output mode). SCK0 = 1 with serial transfer suspended.

② Manipulate the P27 output latch with a bit manipulation instruction.

Figure 16-34. SCK0/P27 Pin Configuration


Manipulated by bit manipulation instruction
SCK0/P27
To Internal Circuit
P27 Output Latch
SCK0 (1 while transfer is stopped)
From Serial Clock Control Circuit
When CSIE0 = 1
and
CSIM01 and CSIM00 are 1 and 0, or 1 and 1.

CHAPTER 17 SERIAL INTERFACE CHANNEL 0 (μPD780058Y Subseries)

The μPD780058Y Subseries incorporates three channels of serial interfaces. Differences between channels 0, 1, and 2 are as follows (Refer to **CHAPTER 18 SERIAL INTERFACE CHANNEL 1** for details of the serial interface channel 1. Refer to **CHAPTER 19 SERIAL INTERFACE CHANNEL 2** for details of the serial interface channel 2).

Table 17-1. Differences Between Channels 0, 1, and 2

Serial Transfer Mode		Channel 0	Channel 1	Channel 2
3-wire serial I/O	Clock selection	$f_{XX}/2$, $f_{XX}/2^2$, $f_{XX}/2^3$, $f_{XX}/2^4$, $f_{XX}/2^5$, $f_{XX}/2^6$, $f_{XX}/2^7$, $f_{XX}/2^8$, external clock, TO2 output	$f_{XX}/2$, $f_{XX}/2^2$, $f_{XX}/2^3$, $f_{XX}/2^4$, $f_{XX}/2^5$, $f_{XX}/2^6$, $f_{XX}/2^7$, $f_{XX}/2^8$, external clock, TO2 output	External clock, baud rate generator output
	Transfer method	MSB/LSB switchable as the start bit	MSB/LSB switchable as the start bit Automatic transmit/receive function	MSB/LSB switchable as the start bit
	Transfer end flag	Serial transfer end interrupt request flag (CSIIF0)	Serial transfer end interrupt request flag (CSIIF1)	Serial transfer end interrupt request flag (SRIF)
2-wire serial I/O		Use possible	None	None
I^{2}C bus (Inter IC Bus)				
UART (Asynchronous serial interface)		None		Use possible Timer-division transfer function

17.1 Serial Interface Channel 0 Functions

Serial interface channel 0 employs the following four modes.

- Operation stop mode
- 3-wire serial I/O mode
- 2-wire serial I/O mode
- I^{2}C (Inter IC) bus mode

Caution Do not change the operating mode (3-wire serial I/O, 2-wire serial I/O, or SBI) while serial interface channel 0 is enabled to operate. To change the operating mode, once stop the serial operation.

(1) Operation stop mode

This mode is used when serial transfer is not carried out. Power consumption can be reduced.

(2) 3-wire serial I/O mode (MSB-/LSB-first selectable)

This mode is used for 8-bit data transfer using three lines, one each for serial clock ($\overline{\text{SCK0}}$), serial output (SO0) and serial input (SI0). This mode enables simultaneous transmission/reception and therefore reduces the data transfer processing time.

The start bit of transferred 8-bit data is switchable between MSB and LSB, so that devices can be connected regardless of their start bit recognition.

This mode should be used when connecting with peripheral I/O devices or display controllers which incorporate a conventional synchronous clocked serial interface as is the case with the 75X/XL, 78K, and 17K Series.

(3) 2-wire serial I/O mode (MSB-first)

This mode is used for 8-bit data transfer using two lines of serial clock ($\overline{\text{SCK0}}$) and serial data bus (SB0 or SB1).

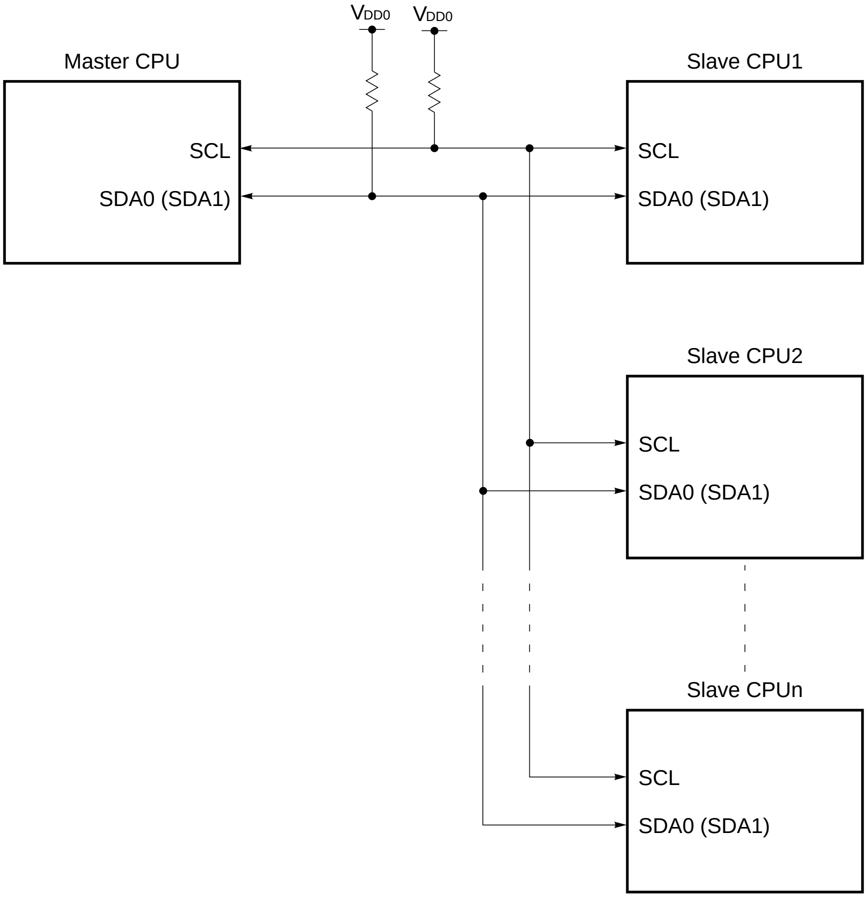
This mode enables to cope with any one of the possible data transfer formats by controlling the $\overline{\text{SCK0}}$ level and the SB0 or SB1 output level. Thus, the handshake line previously necessary for connection of two or more devices can be removed, resulting in the increased number of available input/output ports.

(4) I^{2}C (Inter IC) bus mode (MSB-first)

This mode is used for 8-bit data transfer with two or more devices using two lines of serial clock (SCL) and serial data bus (SDA0 or SDA1).

This mode is in compliance with the I^{2}C bus format. In this mode, the transmitter outputs three kinds of data onto the serial data bus: “start condition”, “data”, and “stop condition”, to be actually sent or received. The receiver automatically distinguishes the received data into “start condition”, “data”, or “stop condition”, by hardware.

Figure 17-1. Serial Bus Configuration Example Using I^{2}C Bus



17.2 Serial Interface Channel 0 Configuration

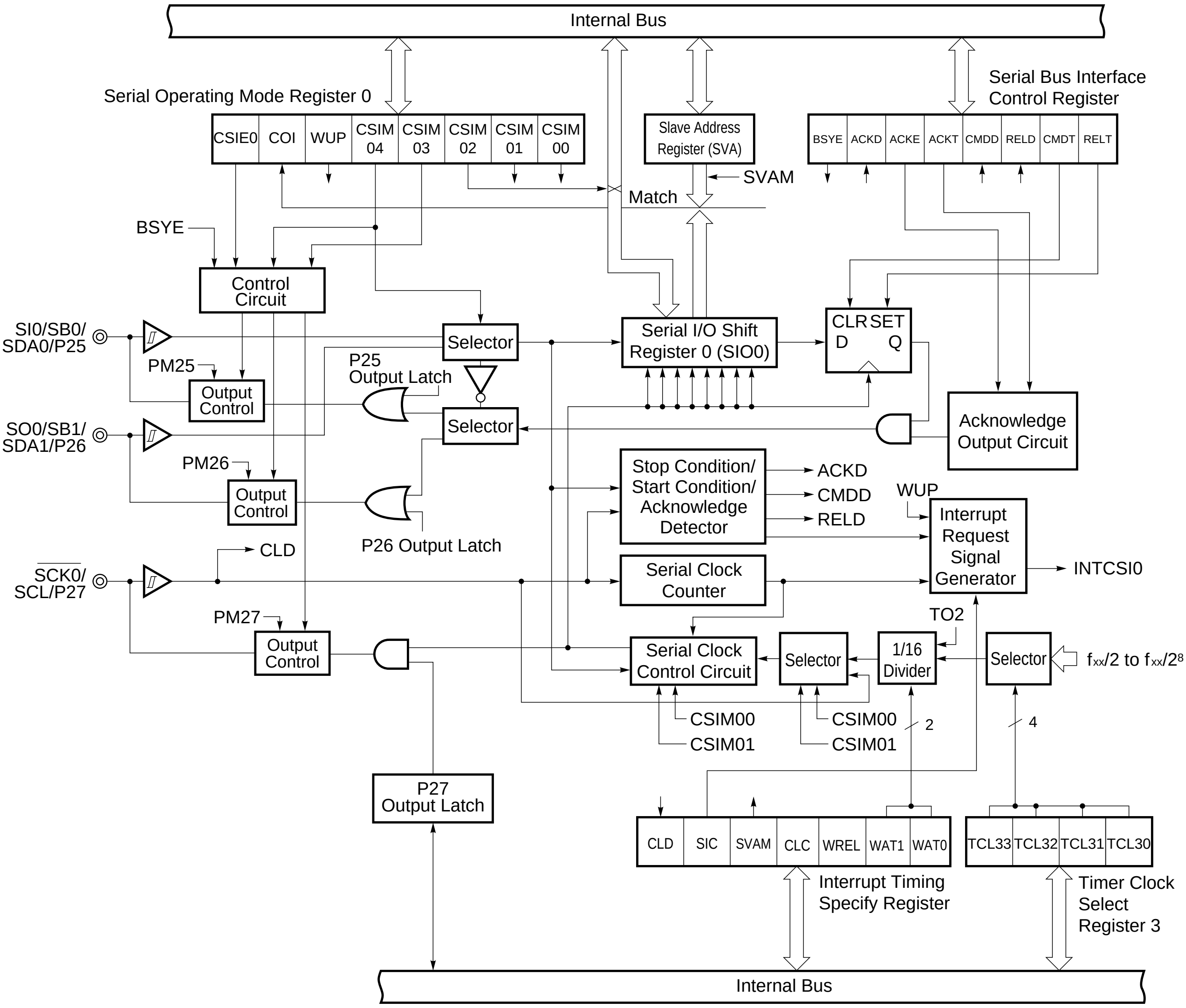
Serial interface channel 0 consists of the following hardware.

Table 17-2. Serial Interface Channel 0 Configuration

Item	Configuration
Register	Serial I/O shift register 0 (SIO0) Slave address register (SVA)
Control register	Timer clock select register 3 (TCL3) Serial operating mode register 0 (CSIM0) Serial bus interface control register (SBIC) Interrupt timing specify register (SINT) Port mode register 2 (PM2) **Note**

Note Refer to **Figure 6-7 P20, P21, and P23 to P26 Block Diagram** and **Figure 6-8 P22 and P27 Block Diagram**.

Figure 17-2. Serial Interface Channel 0 Block Diagram



Remark Output Control performs selection between CMOS output and N-ch open drain output.

(1) Serial I/O shift register 0 (SIO0)

This is an 8-bit register to carry out parallel-serial conversion and to carry out serial transmission/reception (shift operation) in synchronization with the serial clock.

SIO0 is set with an 8-bit memory manipulation instruction.

When bit 7 (CSIE0) of serial operating mode register 0 (CSIM0) is 1, writing data to SIO0 starts serial operation. In transmission, data written to SIO0 is output to the serial output (SO0) or serial data bus (SB0/SB1). In reception, data is read from the serial input (SI0) or SB0/SB1 to SIO0.

Note that, if a bus is driven in the I^2C bus mode or 2-wire serial I/O mode, the bus pin must serve for both input and output. Therefore, the transmission N-ch transistor of the device which will start reception of data must be turned off beforehand. Consequently, write FFH to SIO0 in advance.

In the I^2C bus mode, set SIO0 to FFH with bit 7 (BSYE) of the serial bus interface control register (SBIC) set to 0.

$\overline{\text{RESET}}$ input makes SIO0 undefined.

Caution Do not execute an instruction that writes SIO0 in the I^2C bus mode while WUP (bit 5 of the serial operating mode register 0 (CSIM0)) = 1. Even if such an instruction is not executed, data can be received when the wake-up function is used (WUP = 1). For the detail of the wake-up function, refer to 17.4.4 (1) (c) Wake-up function.

(2) Slave address register (SVA)

This is an 8-bit register to set the slave address value for connection of a slave device to the serial bus.

SVA is set with an 8-bit memory manipulation instruction. This register is not used in the 3-wire serial I/O mode.

The master device outputs a slave address for selection of a particular slave device to the connected slave device. These two data (the slave address output from the master device and the SVA value) are compared with an address comparator. If they match, the slave device has been selected. In that case, bit 6 (COI) of serial operating mode register 0 (CSIM0) becomes 1.

Address comparison can also be executed on the data of LSB-masked high-order 7 bits by setting bit 4 (SVAM) of the interrupt timing specify register (SINT) to 1.

If no matching is detected in address reception, bit 2 (RELD) of the serial bus interface control register (SBIC) is cleared to 0. In the I^2C bus mode, the wake-up function can be used by setting the bit 5 (WUP) of CSIM0 to 1. In this case, the interrupt request signal (INTCSI0) is generated when the slave address output by the master coincides with the value of SVA (the interrupt request signal is also generated when the stop condition is detected), and it can be learned by this interrupt request that the master requests for communication. To use the wake-up function, set SIC to 1.

Further, an error can be detected by using SVA when the device transmits data as master or slave device in I^2C bus mode or 2-wire serial I/O mode.

$\overline{\text{RESET}}$ input makes SVA undefined.

(3) SO0 latch

This latch holds SI0/SB0/SDA0/P25 and SO0/SB1/SDA1/P26 pin levels. It can be directly controlled by software.

(4) Serial clock counter

This counter counts the serial clocks to be output and input during transmission/reception and to check whether 8-bit data has been transmitted/received.

(5) Serial clock control circuit

This circuit controls serial clock supply to the serial I/O shift register 0 (SIO0). When the internal system clock is used, the circuit also controls clock output to the $\overline{\text{SCK0}}$/SCL/P27 pin.

(6) Interrupt signal generator

This circuit controls interrupt request signal generation. It generates interrupt request signals according to the settings of interrupt timing specification register (SINT) bits 0 and 1 (WAT0, WAT1) and serial operation mode register 0 (CSIM0) bit 5 (WUP), as shown in Table 17-3.

(7) Acknowledge output circuit and stop condition/start condition/acknowledge detector

These two circuits output and detect various control signals in the I^{2}C mode.

These do not operate in the 3-wire serial I/O mode and 2-wire serial I/O mode.

Table 17-3. Serial Interface Channel 0 Interrupt Request Signal Generation

Serial Transfer Mode	BSYE	WUP	WAT1	WAT0	ACKE	Description
3-wire or 2-wire serial I/O mode	0	0	0	0	0	An interrupt request signal is generated each time 8 serial clocks are counted.
	Other than above					Setting prohibited
I^{2}C bus mode (transmit)	0	0	1	0	0	An interrupt request signal is generated each time 8 serial clocks are counted (8-clock wait). Normally, during transmission the settings WAT21, WAT0 = 1, 0, are not used. They are used only when wanting to coordinate receive time and processing systematically using software. ACK information is generated by the receiving side, thus ACKE should be set to 0 (disable).
			1	1	0	An interrupt request signal is generated each time 9 serial clocks are counted (9-clock wait). ACK information is generated by the receiving side, thus ACKE should be set to 0 (disable).
	Other than above					Setting prohibited
I^{2}C bus mode (receive)	1	0	1	0	0	An interrupt request signal is generated each time 8 serial clocks are counted (8-clock wait). ACK information is output by manipulating ACKT by software after an interrupt request is generated.
			1	1	0/1	An interrupt request signal is generated each time 9 serial clocks are counted (9-clock wait). To automatically generate ACK information, preset ACKE to 1 before transfer start. However, in the case of the master, set ACKE to 0 (disable) before receiving the last data.
	1	1	1	1	1	After address is received, if the values of the serial I/O shift register 0 (SI00) and the slave address register (SVA) match, and if the stop condition is detected, an interrupt request signal is generated. To automatically generate ACK information, preset ACKE to 1 (enable) before transfer start.
	Other than above					Setting prohibited

Remark BSYE: Bit 7 of serial bus interface control register (SBIC)
ACKE: Bit 5 of serial bus interface control register (SBIC)

17.3 Serial Interface Channel 0 Control Registers

The following four types of registers are used to control serial interface channel 0.

- Timer clock select register 3 (TCL3)
- Serial operating mode register 0 (CSIM0)
- Serial bus interface control register (SBIC)
- Interrupt timing specify register (SINT)

(1) Timer clock select register 3 (TCL3)

This register sets the serial clock of serial interface channel 0.
TCL3 is set with an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets TCL3 to 88H.

Figure 17-3. Timer Clock Select Register 3 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33	TCL32	TCL31	TCL30	Serial Interface Channel 0 Serial Clock Selection					
				Serial Clock in I^2C Bus Mode			Serial Clock in 2-Wire or 3-Wire Serial I/O Mode		
					MCS = 1	MCS = 0		MCS = 1	MCS = 0
0	1	1	0	$f_{XX}/2^5$	Setting prohibited	$f_X/2^6$ (78.1 kHz)	$f_{XX}/2$	Setting prohibited	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.77 kHz)	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^9$	$f_X/2^9$ (9.77 kHz)	$f_X/2^{10}$ (4.88 kHz)	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^{10}$	$f_X/2^{10}$ (4.88 kHz)	$f_X/2^{11}$ (2.44 kHz)	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.44 kHz)	$f_X/2^{12}$ (1.22 kHz)	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_{XX}/2^{12}$	$f_X/2^{12}$ (1.22 kHz)	$f_X/2^{13}$ (0.61 kHz)	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
Other than above				Setting prohibited					

TCL37	TCL36	TCL35	TCL34	Serial Interface Channel 1 Serial Clock Selection		
					MCS = 1	MCS = 0
0	1	1	0	$f_{XX}/2$	Setting prohibited	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
Other than above				Setting prohibited		

Caution When rewriting TCL3 to other data, stop the serial transfer operation beforehand.

Remarks 1. f_{XX} : Main system clock frequency (f_X or $f_X/2$)

2. f_X : Main system clock oscillation frequency

3. MCS : Bit 0 of oscillation mode selection register (OSMS)

4. Figures in parentheses apply to operation with f_X = 5.0 MHz.

(2) Serial operating mode register 0 (CSIM0)

This register sets serial interface channel 0 serial clock, operating mode, operation enable/stop wake-up function and displays the address comparator match signal.

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM0 to 00H.

Caution Do not change the operating mode (3-wire serial I/O, 2-wire serial I/O, or SBI) while serial interface channel 0 is enabled to operate. To change the operating mode, once stop the serial operation.

Figure 17-4. Serial Operating Mode Register 0 Format

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W Note 1

R/W

CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection
0	×	Input clock to $\overline{\text{SCK0}}$/SCL pin from off-chip
1	0	8-bit timer register 2 (TM2) output Note 2
1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operation Mode	Start Bit	SI0/SB0/SDA0/ P25 Pin Function	SO0/SB1/SDA1/ P26 Pin Function	$\overline{\text{SCK0}}$/SCL/P27 Pin Function
0	×	0	Note 3 1	Note 3 ×	0	0	0	1	3-wire serial I/O mode	MSB	SI0 Note 3 (Input)	SO0 (CMOS output)	$\overline{\text{SCK0}}$ (CMOS input/output)
		1								LSB			
1	1	0	Note 4 ×	Note 4 ×	0	0	0	1	2-wire serial I/O mode or I^2C bus mode	MSB	P25 (CMOS input/output)	SB1/SDA1 (N-ch open-drain input/output)	$\overline{\text{SCK0}}$/SCL (N-ch open-drain input/ output)
		1	0	0	Note 4 ×	Note 4 ×	0	1			SB0/SDA0 (N-ch open-drain input/output)	P26 (CMOS input/output)	

R/W

WUP	Wake-up Function Control Note 5
0	Interrupt request signal generation with each serial transfer in any mode
1	Interrupt request signal generation when the address received after detecting start condition (when CMDD = 1) matches the slave address register (SVA) data in I^2C bus mode

R

COI	Slave Address Comparison Result Flag Note 6
0	Slave address register (SVA) not equal to serial I/O shift register 0 (SIO0) data
1	Slave address register (SVA) equal to serial I/O shift register 0 (SIO0) data

R/W

CSIE0	Serial Interface Channel 0 Operation Control
0	Operation stopped
1	Operation enabled

Notes
1. Bit 6 (COI) is a read-only bit.
2. I^2C bus mode, the clock frequency becomes 1/16 of that output from TO2.
3. Can be used as P25 (CMOS input/output) when used only for transmission.
4. Can be used freely as port function.
5. To use the wake-up function (WUP = 1), set the bit 5 (SIC) of the interrupt timing specify register (SINT) to 1. Do not execute an instruction that writes the serial I/O shift register 0 (SIO0) while WUP = 1.
6. When CSIE0 = 0, COI becomes 0.

Remark × : don't care

PM×× : Port mode register

P×× : Port output latch

(3) Serial bus interface control register (SBIC)

This register sets serial bus interface operation and displays statuses.

SBIC is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears SBIC to 00H.

Figure 17-5. Serial Bus Interface Control Register Format (1/2)

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W Note

R/W	RELT	Used for stop condition signal output. When RELT = 1, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W	CMDT	Used for start condition signal output. When CMDT = 1, SO0 latch is cleared to 0. After SO0 latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R

RELD	Stop Condition Detection
Clear Conditions (RELD = 0)	Set Conditions (RELD = 1)
• When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied	• When stop condition signal is detected

R

CMDD	Start Condition Detection
Clear Conditions (CMDD = 0)	Set Conditions (CMDD = 1)
• When transfer start instruction is executed • When stop condition signal is detected • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied	• When start condition signal is detected

R/W	ACKT	Used to generate the $\overline{\text{ACK}}$ signal by software when 8-clock wait mode is selected. Keeps SDA0 (SDA1) low from set instruction (ACKT = 1) execution to the next falling edge of SCL. Also cleared to 0 upon start of serial interface transfer or when CSIE0 = 0.

Note Bits 2, 3, and 6 (RELD, CMDD, and ACKD) are read-only bits.

Remark CSIE0: Bit 7 of serial operating mode register 0 (CSIM0)

Figure 17-5. Serial Bus Interface Control Register Format (2/2)

R/W

ACKE	Acknowledge Signal Output Control Note 1
0	Disables acknowledge signal automatic output. (However, output with ACKT is enabled) Used for reception when 8-clock wait mode is selected or for transmission. Note 2
1	Enables acknowledge signal automatic output. Outputs acknowledge signal in synchronization with the falling edge of the 9th SCL clock cycle (automatically output when ACKE = 1). However, not automatically cleared to 0 after acknowledge signal output. Used in reception with 9-clock wait mode selected.

R

ACKD	Acknowledge Detection
Clear Conditions (ACKD = 0)	Set Conditions (ACKD = 1)
• While executing the transfer start instruction • When CSIE0 = 0 • When $\overline{\text{RESET}}$input is applied	• When acknowledge signal ($\overline{\text{ACK}}$) is detected at the rising edge of SCL clock after completion of transfer

R/W

Note 3 BSYE	Control of N-ch Open-Drain Output for Transmission in I^2C Bus Mode Note 4
0	Output enabled (transmission)
1	Output disabled (reception)

Notes **1.** Setting should be performed before transfer.

2. If 8-clock wait mode is selected, the acknowledge signal at reception time must be output using ACKT.

3. The busy mode can be canceled by start of serial interface transfer or reception of address signal. However, the BSYE flag is not cleared to 0.

4. When using the wake-up function, be sure to set the BSYE to 1.

Remark CSIE0: Bit 7 of serial operating mode register 0 (CSIM0)

(4) Interrupt timing specify register (SINT)

This register sets the bus release interrupt and address mask functions and displays the $\overline{\text{SCK0}}$/SCL pin level status. SINT is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears SINT to 00H.

Figure 17-6. Interrupt Timing Specify Register Format (1/2)

Symbol	7	⑥	⑤	④	③	②	1	0	Address	After Reset	R/W
SINT	0	CLD	SIC	SVAM	CLC	WREL	WAT1	WAT0	FF63H	00H	R/W Note 1

R/W

WAT1	WAT0	Wait and Interrupt Control
0	0	Generates interrupt service request at rising edge of 8th $\overline{\text{SCK0}}$ clock cycle. (keeping clock output in high impedance)
0	1	Setting prohibited
1	0	Used in I^2C bus mode. (8-clock wait) Generates interrupt service request at rising edge of 8th SCK0 clock cycle. (In the case of master device, makes SCL output low to enter wait state after 8 clock pulses are output. In the case of slave device, makes SCL output low to request wait state after 8 clock pulses are input.)
1	1	Used in I^2C bus mode. (9-clock wait) Generates interrupt service request at rising edge of 9th SCK0 clock cycle. (In the case of master device, makes SCL output low to enter wait state after 9 clock pulses are output. In the case of slave device, makes SCL output low to request wait state after 9 clock pulses are input.)

R/W

WREL	Wait Sate Cancellation Control
0	Wait state has been cancelled.
1	Cancels wait state. Automatically cleared to 0 when the state is cancelled. (Used to cancel wait state by means of WAT0 and WAT1.)

R/W

CLC	Clock Level Control Note 2
0	Used in I^2C bus mode. Make output level of SCL pin low unless serial transfer is being performed.
1	Used in I^2C bus mode. Make SCL pin enter high-impedance state unless serial transfer is being performed. (except for clock line which is kept high) Used to enable master device to generate start condition and stop condition signals.

Notes 1. Bit 6 (CLD) is a read-only bit.

2. When not using the I^2C mode, set CLC to 0.

Figure 17-6. Interrupt Timing Specify Register Format (2/2)

R/W

SVAM	SVA Bit to be Used as Slave Address
0	Bits 0 to 7
1	Bits 1 to 7

R/W

SIC	INTCSI0 Interrupt Sourse Selection[Note 1]
0	CSIIF0 is set to 1 upon termination of serial interface channel 0 transfer
1	CSIIF0 is set to 1 upon stop condition detection or termination of serial interface channel 0 transfer

R

CLD	$\overline{\text{SCK0}}$/SCL Pin Level[Note 2]
0	Low level
1	High level

Notes 1. When using wake-up function in the I^{2}C mode, set SIC to 0.

2. When CSIE0 = 0, CLD becomes 0.

Remark SVA : Slave address register

CSIIF0 : Interrupt request flag corresponding to INTCSI0

CSIE0 : Bit 7 of serial operating mode register 0 (CSIM0)

17.4 Serial Interface Channel 0 Operations

The following four operating modes are available to the serial interface channel 0.

- Operation stop mode
- 3-wire serial I/O mode
- 2-wire serial I/O mode
- I^{2}C (Inter IC) bus mode

17.4.1 Operation stop mode

Serial transfer is not carried out in the operation stop mode. Thus, power consumption can be reduced. The serial I/O shift register 0 (SIO0) does not carry out shift operation either and thus it can be used as ordinary 8-bit register.

In the operation stop mode, the P25/SI0/SB0/SDA0, P26/SO0/SB1/SDA1, and P27/$\overline{\text{SCK0}}$/SCL pins can be used as general input/output ports.

(1) Register setting

The operation stop mode is set with the serial operating mode register 0 (CSIM0).

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM0 to 00H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W

R/W	CSIE0	Serial Interface Channel 0 Operation Control
	0	Operation stopped
	1	Operation enabled

17.4.2 3-wire serial I/O mode operation

The 3-wire serial I/O mode is valid for connection of peripheral I/O units and display controllers which incorporate a conventional synchronous clocked serial interface as is the case with the 75X/XL, 78K, and 17K Series.

Communication is carried out with three lines of serial clock ($\overline{\text{SCK0}}$), serial output (SO0), and serial input (SI0).

(1) Register setting

The 3-wire serial I/O mode is set with the serial operating mode register 0 (CSIM0) and serial bus interface control register (SBIC).

(a) Serial operating mode register 0 (CSIM0)

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM0 to 00H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W [Note 1]

R/W

CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection
0	×	Input clock to $\overline{\text{SCK0}}$ pin from off-chip
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operation Mode	Start Bit	SIO/SB0/SDA0 /P25 Pin Function	SO0/SB1/SDA1 /P26 Pin Function	$\overline{\text{SCK0}}$/SCL/P27 Pin Function
0	×	0	[Note 2] 1	[Note 2] ×	0	0	0	1	3-wire serial I/O mode	MSB	SI0 [Note 2] (Input)	SO0 (CMOS output)	$\overline{\text{SCK0}}$ (CMOS input/output)
		1								LSB			
1	1	2-wire serial I/O mode (see **17.4.3 2-wire serial I/O mode operation**.) or I^{2}C bus mode (see **17.4.4 I^{2}C bus mode operation**.)											

R/W

WUP	Wake-up Function Control [Note 3]
0	Interrupt request signal generation with each serial transfer in any mode
1	Interrupt request signal generation when the address received after detecting start condition (when CMDD = 1) matches the slave address register (SVA) data in I^{2}C bus mode

R/W

CSIE0	Serial Interface Channel 0 Operation Control
0	Operation stopped
1	Operation enabled

Notes 1. Bit 6 (COI) is a read-only bit.

2. Can be used as P25 (CMOS input/output) when used only for transmission.

3. Be sure to set WUP to 0 when the 3-wire serial I/O mode is selected.

Remark		
×	:	don’t care
PM××	:	Port mode register
P××	:	Port output latch

(b) Serial bus interface control register (SBIC)

SBIC is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears SBIC to 00H.

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W

R/W	RELT	When RELT = 1, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W	CMDT	When CMDT = 1, SO0 latch is cleared to 0. After SO0 latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

CSIE0: Bit 7 of serial operating mode register 0 (CSIM0)

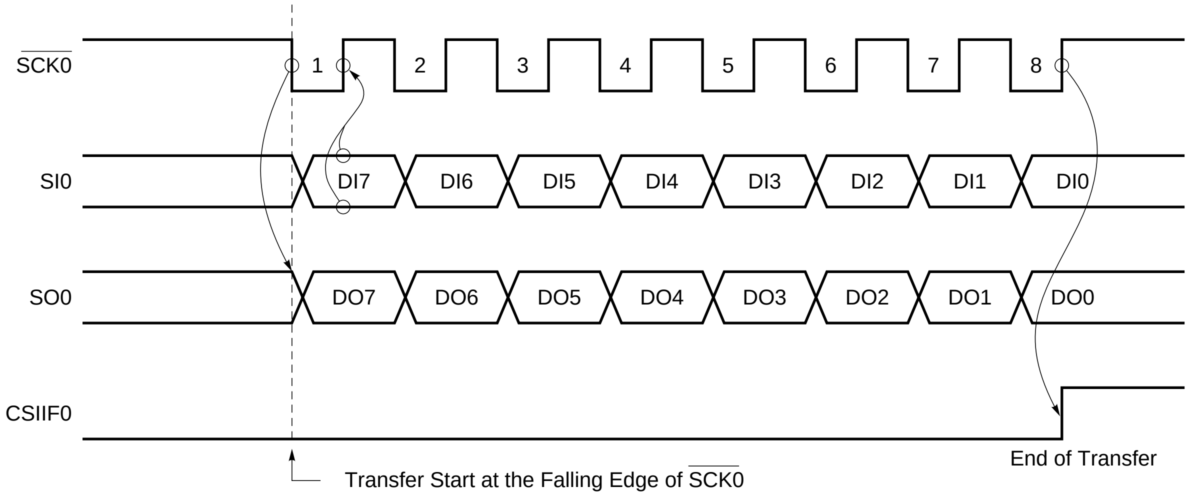
(2) Communication operation

The 3-wire serial I/O mode is used for data transmission/reception in 8-bit units. Bit-wise data transmission/reception is carried out in synchronization with the serial clock.

Shift operation of the serial I/O shift register 0 (SIO0) is carried out at the falling edge of the serial clock ($\overline{\text{SCK0}}$). The transmitted data is held in the SO0 latch and is output from the SO0 pin. The received data input to the SI0 pin is latched in SIO0 at the rising edge of $\overline{\text{SCK0}}$.

Upon termination of 8-bit transfer, SIO0 operation stops automatically and the interrupt request flag (CSIIF0) is set.

Figure 17-7. 3-wire Serial I/O Mode Timings



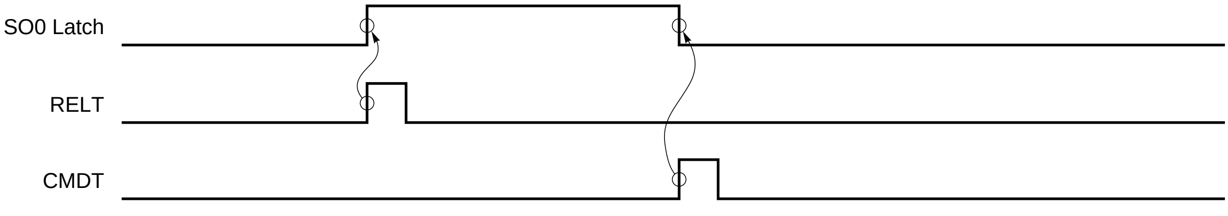
The SO0 pin is a CMOS output pin and outputs current SO0 latch statuses. Thus, the SO0 pin output status can be manipulated by setting bit 0 (RELT) and bit 1 (CMDT) of serial bus interface control register (SBIC). However, do not carry out this manipulation during serial transfer.

Control the $\overline{\text{SCK0}}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (see **17.4.7 $\overline{\text{SCK0}}$/SCL/P27 pin output manipulation**).

(3) Other signals

Figure 17-8 shows RELT and CMDT operations.

Figure 17-8. RELT and CMDT Operations



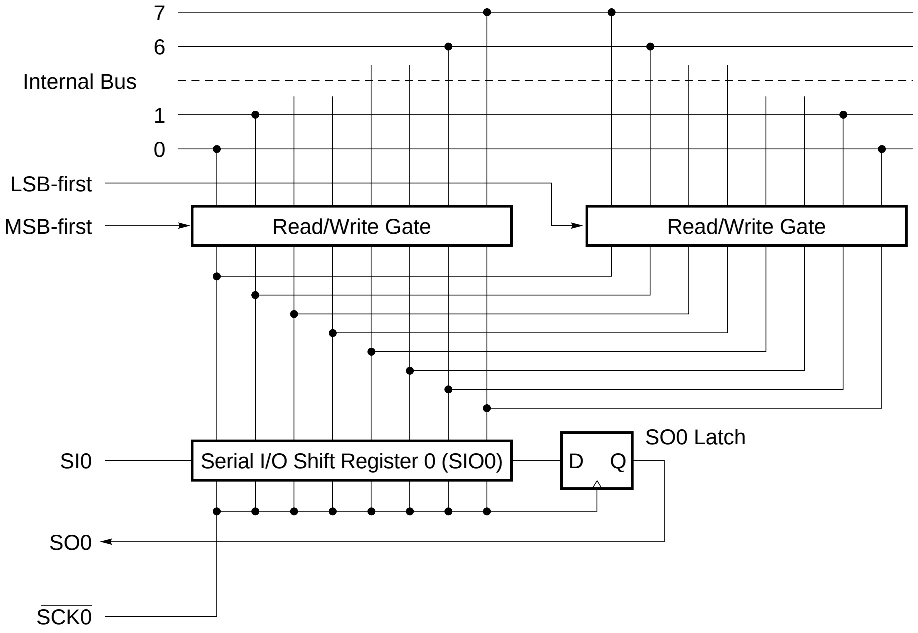
(4) MSB/LSB switching as the start bit

The 3-wire serial I/O mode enables to select transfer to start from MSB or LSB.

Figure 17-9 shows the configuration of the serial I/O shift register 0 (SIO0) and internal bus. As shown in the figure, MSB/LSB can be read/written in reverse form.

MSB/LSB switching as the start bit can be specified with bit 2 (CSIM02) of the serial operating mode register 0 (CSIM0).

Figure 17-9. Circuit of Switching in Transfer Bit Order



Start bit switching is realized by switching the bit order for data write to SIO0. The SIO0 shift order remains unchanged.

Thus, switching between MSB-first and LSB-first must be performed before writing data to SIO0.

(5) Transfer start

Serial transfer is started by setting transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied.

- Serial interface channel 0 operation control bit (CSIE0) = 1.
- Internal serial clock is stopped or $\overline{\text{SCK0}}$ is a high level after 8-bit serial transfer.

Caution If CSIE0 is set to “1” after data write to SIO0, transfer does not start.

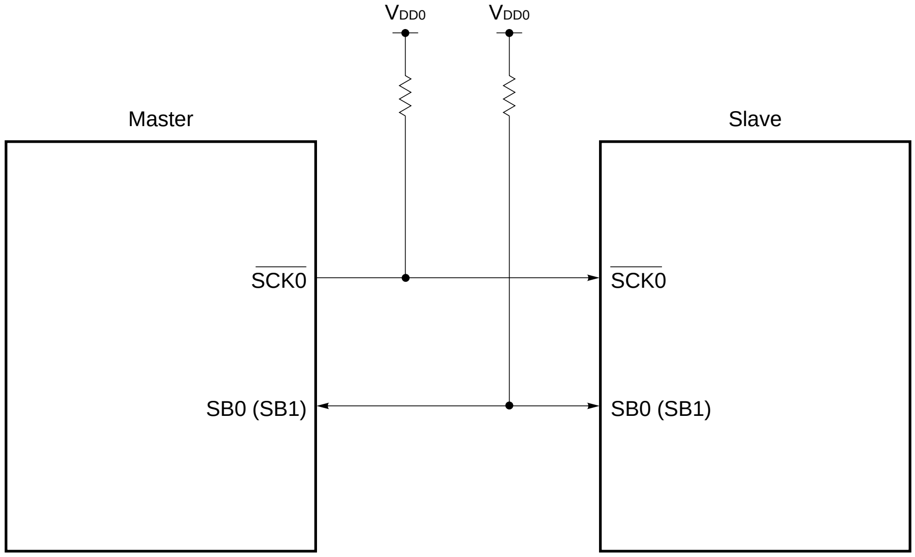
Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (CSIIF0) is set.

17.4.3 2-wire serial I/O mode operation

The 2-wire serial I/O mode can cope with any communication format by program.

Communication is basically carried out with two lines of serial clock ($\overline{\text{SCK0}}$) and serial data input/output (SB0 or SB1).

Figure 17-10. Serial Bus Configuration Example Using 2-wire Serial I/O Mode



(1) Register setting

The 2-wire serial I/O mode is set with the serial operating mode register 0 (CSIM0), the serial bus interface control register (SBIC), and the interrupt timing specify register (SINT).

(a) Serial operating mode register 0 (CSIM0)

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM0 to 00H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W Note 1

R/W

CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection
0	×	Input clock to $\overline{\text{SCK0}}$ pin from off-chip
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operation Mode	Start Bit	SI0/SB0/SDA0 /P25 Pin Function	SO0/SB1/SDA1 /P26 Pin Function	$\overline{\text{SCK0}}$/SCL/P27 Pin Function
0	×	3-wire Serial I/O mode (see **17.4.2 3-wire serial I/O mode operation**)											
1	1	0	Note 2 ×	Note 2 ×	0	0	0	1	2-wire serial I/O mode or I^{2}C bus mode	MSB	P25 (CMOS input/output)	SB1/SDA1 (N-ch open-drain input/output)	$\overline{\text{SCK0}}$/SCL (N-ch open-drain input/output)
		1	0	0	Note 2 ×	Note 2 ×	0	1			SB0/SDA0 (N-ch open-drain input/output)	P26 (CMOS input/output)	

R/W

WUP	Wake-up Function Control Note 3
0	Interrupt request signal generation with each serial transfer in any mode
1	Interrupt request signal generation when the address received after detecting start condition (when CMDD = 1) matches the slave address register (SVA) data in I^{2}C bus mode

R

COI	Slave Address Comparison Result Flag Note 4
0	Slave address register (SVA) not equal to serial I/O shift register 0 (SIO0) data
1	Slave address register (SVA) equal to serial I/O shift register 0 (SIO0) data

R/W

CSIE0	Serial Interface Channel 0 Operation Control
0	Operation stopped
1	Operation enabled

Notes 1. Bit 6 (COI) is a read-only bit.

2. Can be used freely as port function.

3. Be sure to set WUP to 0 when the 2-wire serial I/O mode.

4. When CSIE0 = 0, COI becomes 0.

Remark		
×	:	don't care
PM××	:	Port mode register
P××	:	Port output latch

(b) Serial bus interface control register (SBIC)

SBIC is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears SBIC to 00H.

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W

R/W	RELT	When RELT = 1, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W	CMDT	When CMDT = 1, SO0 latch is cleared to 0. After SO0 latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

CSIE0: Bit 7 of serial operating mode register 0 (CSIM0)

(c) Interrupt timing specify register (SINT)

SINT is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears SINT to 00H.

Symbol	7	⑥	⑤	④	③	②	1	0	Address	After Reset	R/W
SINT	0	CLD	SIC	SVAM	CLC	WREL	WAT1	WAT0	FF63H	00H	R/W Note 1

R/W	SIC	INTCSI0 Interrupt Source Selection
	0	CSIIF0 is set to 1 upon termination of serial interface channel 0 transfer
	1	CSIIF0 is set to 1 upon bus release detection or termination of serial interface channel 0 transfer

R	CLD	$\overline{\text{SCK0}}$ Pin Level Note 2
	0	Low level
	1	High level

Notes 1. Bit 6 (CLD) is a read-only bit.
2. When CSIE0 = 0, CLD becomes 0.

Caution Be sure to set bits 0 to 3 to 0 in the 2-wire serial I/O mode is used.

Remark CSIIF0 : Interrupt request flag corresponding to INTCSI0
CSIE0 : Bit 7 of serial operating mode register 0 (CSIM0)

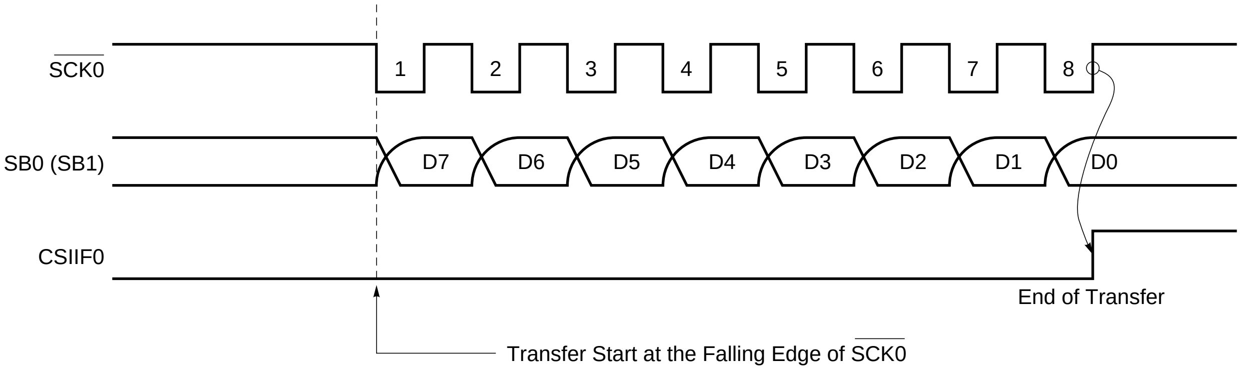
(2) Communication operation

The 2-wire serial I/O mode is used for data transmission/reception in 8-bit units. Data transmission/reception is carried out bit-wise in synchronization with the serial clock.

Shift operation of the serial I/O shift register 0 (SIO0) is carried out in synchronization with the falling edge of the serial clock ($\overline{\text{SCK0}}$). The transmit data is held in the SO0 latch and is output from the SB0/SDA0/P25 (or SB1/SDA1/P26) pin on an MSB-first basis. The receive data input from the SB0 (or SB1) pin is latched into the SIO0 at the rising edge of $\overline{\text{SCK0}}$.

Upon termination of 8-bit transfer, the SIO0 operation stops automatically and the interrupt request flag (CSIIF0) is set.

Figure 17-11. 2-wire Serial I/O Mode Timings



The SB0 (or SB1) pin specified for the serial data bus is an N-ch open-drain input/output and thus it must be externally connected to a pull-up resistor. Because N-ch open-drain output must go into a high-impedance state during data reception, write FFH to SIO0 in advance.

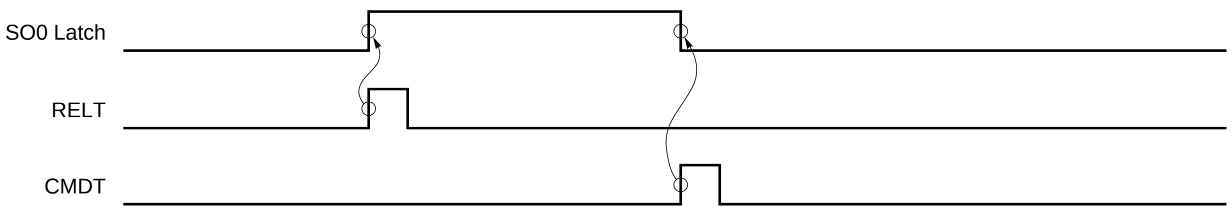
The SB0 (or SB1) pin generates the SO0 latch status and thus the SB0 (or SB1) pin output status can be manipulated by setting bit 0 (RELT) and bit 1 (CMDT) of serial bus interface control register (SBIC). However, do not carry out this manipulation during serial transfer.

Control the $\overline{\text{SCK0}}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **17.4.7 $\overline{\text{SCK0}}$/SCL/P27 pin output manipulation**).

(3) Other signals

Figure 17-12 shows RELT and CMDT operations.

Figure 17-12. RELT and CMDT Operations



(4) Transfer start

Serial transfer is started by setting transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied.

- Serial interface channel 0 operation control bit (CSIE0) = 1
- Internal serial clock is stopped or $\overline{\text{SCK0}}$ is at high level after 8-bit serial transfer.

Cautions 1. If CSIE0 is set to "1" after data write to SIO0, transfer does not start.

2. Because the N-ch open-drain output must go into a high-impedance state during data reception, write FFH to SIO0 in advance.

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (CSIIF0) is set.

(5) Error detection

In the 2-wire serial I/O mode, the serial bus SB0 (SB1) status being transmitted is fetched into the destination device, that is, serial I/O shift register 0 (SIO0). Thus, transmit error can be detected in the following way.

(a) Method of comparing SIO0 data before transmission to that after transmission

In this case, if two data differ from each other, a transmit error is judged to have occurred.

(b) Method of using the slave address register (SVA)

Transmit data is set to both SIO0 and SVA and is transmitted. After termination of transmission, COI bit (match signal coming from the address comparator) of the serial operating mode register 0 (CSIM0) is tested. If "1", normal transmission is judged to have been carried out. If "0", a transmit error is judged to have occurred.

17.4.4 I²C bus mode operation

The I²C bus mode is provided for when communication operations are performed between a single master device and multiple slave devices. This mode configures a serial bus that includes only a single master device, and is based on the clocked serial I/O format with the addition of bus configuration functions, which allows the master device to communicate with a number of (slave) devices using only two lines: serial clock (SCL) line and serial data bus (SDA0 or SDA1) line. Consequently, when the user plans to configure a serial bus which includes multiple microcontrollers and peripheral devices, using this configuration results in reduction of the required number of port pins and on-board wires.

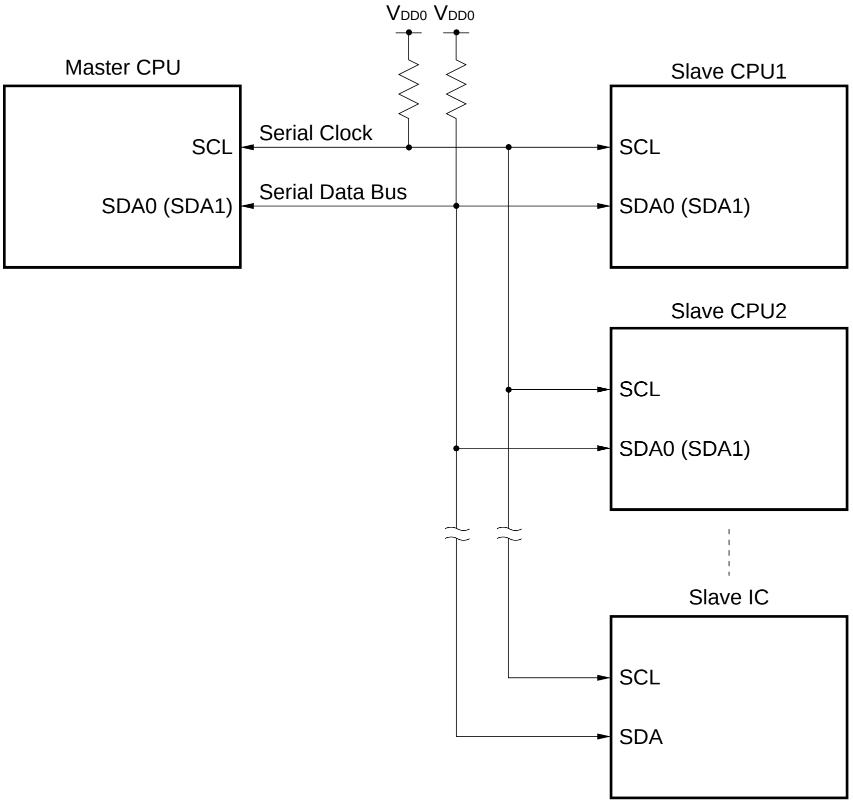
In the I²C bus specification, the master sends start condition, data, and stop condition signals to slave devices through the serial data bus, while slave devices automatically detect and distinguish the type of signals due to the signal detection function incorporated as hardware. An application program that controls I²C bus can be simplified by using this function.

An example of a serial bus configuration is shown in Figure 17-13. This system below is composed of CPUs and peripheral ICs having serial interface hardware that complies with the I²C bus specification.

Note that pull-up resistors are required to connect to both serial clock line and serial data bus line, because open-drain buffers are used for the serial clock pin (SCL) and the serial data bus pin (SDA0 or SDA1) on the I²C bus.

The signals used in the I²C bus mode are described in Table 17-4.

Figure 17-13. Example of Serial Bus Configuration Using I²C Bus



(1) I^{2}C bus mode functions

In the I^{2}C bus mode, the following functions are available.

(a) Automatic identification of serial data

Slave devices automatically detect and identifies start condition, data, and stop condition signals sent in series through the serial data bus.

(b) Chip selection by specifying device addresses

The master device can select a specific slave device connected to the I^{2}C bus and communicate with it by sending in advance the address data corresponding to the destination device.

(c) Wake-up function

When address data is sent from the master device, slave devices compare it with the value registered in their internal slave address registers. If the values in one of the slave devices match, the slave device internally generates an interrupt request signal to terminate the current processing and communicates with the master device (the interrupt request also occurs when the stop condition is detected). Therefore, CPUs other than the selected slave device on the I^{2}C bus can perform independent operations during the serial communication.

(d) Acknowledge signal ($\overline{\text{ACK}}$) control function

The master device and a slave device send and receive acknowledge signals to confirm that the serial communication has been executed normally.

(e) Wait signal ($\overline{\text{WAIT}}$) control function

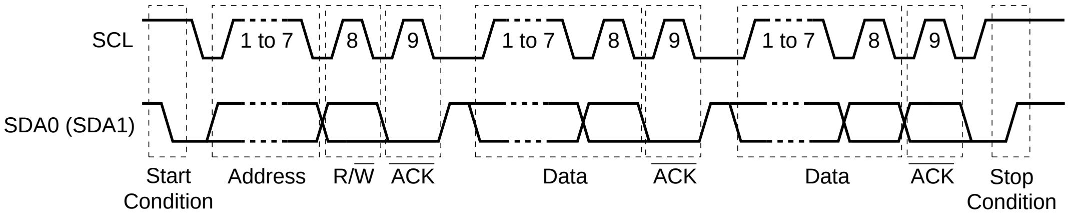
When a slave device is preparing for data transmission or reception and requires more waiting time, the slave device outputs a wait signal on the bus to inform the master device of the wait status.

(2) I^{2}C bus definition

This section describes the format of serial data communications and functions of the signals used in the I^{2}C bus mode.

First, the transfer timings of the “start condition”, “data”, and “stop condition” signals, which are output onto the signal data bus of the I^{2}C bus, are shown in Figure 17-14.

Figure 17-14. I^{2}C Bus Serial Data Transfer Timing



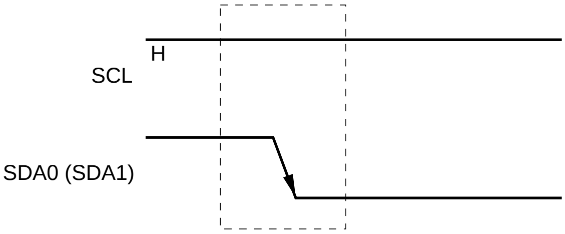
The start condition, slave address, and stop condition signals are output by the master. The acknowledge signal ($\overline{\text{ACK}}$) is output by either the master or the slave device (normally by the device which has received the 8-bit data that was sent). A serial clock (SCL) is continuously supplied from the master device.

(a) Start condition

When the SDA0 (SDA1) pin level is changed from high to low while the SCL pin is high, this transition is recognized as the start condition signal. This start condition signal, which is created using the SCL and SDA0 (or SDA1) pins, is output from the master device to slave devices to initiate a serial transfer. See **17.4.5 Cautions on Use of I^{2}C Bus Mode**, for details of the start condition output.

The start condition signal is detected by hardware incorporated in slave devices.

Figure 17-15. Start Condition

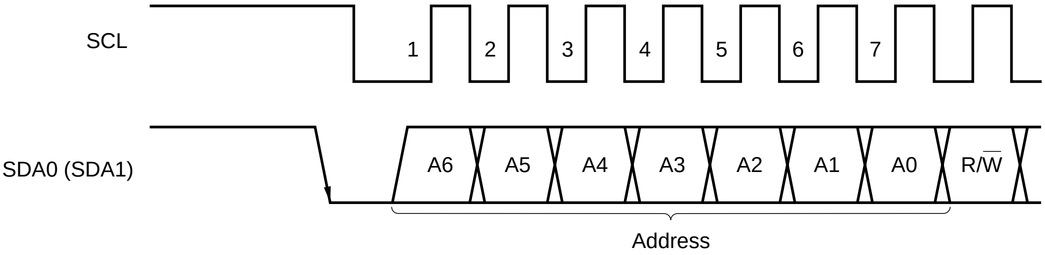


(b) Address

The 7 bits following the start condition signal are defined as an address.

The 7-bit address data is output by the master device to specify a specific slave from among those connected to the bus line. Each slave device on the bus line must therefore have a different address. Therefore, after a slave device detects the start condition, it compares the 7-bit address data received and the data of the slave address register (SVA). After the comparison, only the slave device in which the data are a match becomes the communication partner, and subsequently performs communication with the master device until the master device sends a start condition or stop condition signal.

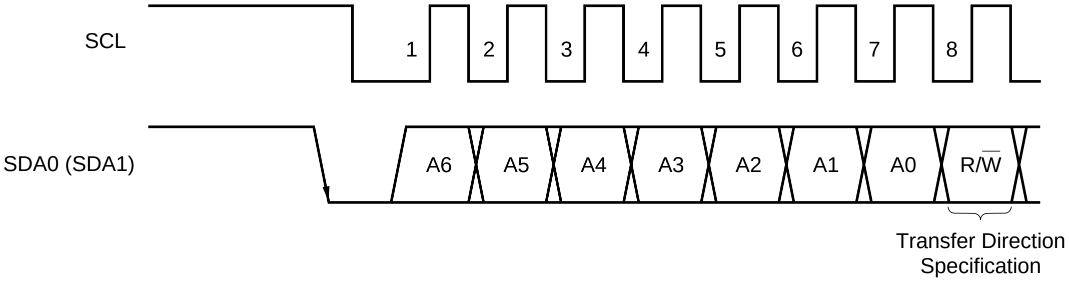
Figure 17-16. Address



(c) Transfer direction specification

The 1 bit that follows the 7-bit address data will be sent from the master device, and it is defined as the transfer direction specification bit. If this bit is 0, it is the master device which will send data to the slave. If it is 1, it is the slave device which will send data to the master.

Figure 17-17. Transfer Direction Specification

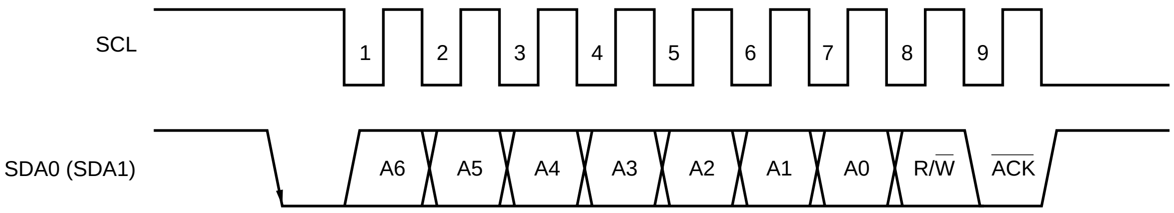


(d) Acknowledge signal ($\overline{\text{ACK}}$)

The acknowledge signal indicates that the transferred serial data has definitely been received. This signal is used between the sending side and receiving side devices for confirmation of correct data transfer. In principle, the receiving side device returns an acknowledge signal to the sending device each time it receives 8-bit data. The only exception is when the receiving side is the master device and the 8-bit data is the last transfer data; the master device outputs no acknowledge signal in this case.

The sending side that has transferred 8-bit data waits for the acknowledge signal which will be sent from the receiving side. If the sending side device receives the acknowledge signal, which means a successful data transfer, it proceeds to the next processing. If this signal is not sent back from the slave device, this means that the data sent has not been received by the slave device, and therefore the master device outputs a stop condition signal to terminate subsequent transmissions.

Figure 17-18. Acknowledge Signal



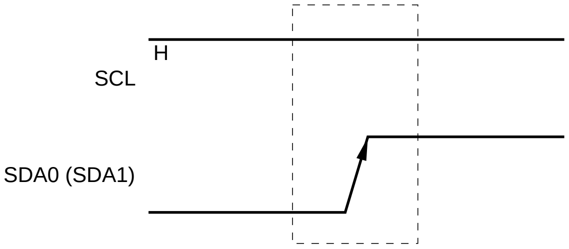
(e) Stop condition

If the SDA0 (SDA1) pin level changes from low to high while the SCL pin is high, this transition is defined as a stop condition signal.

The stop condition signal is output from the master to the slave device to terminate a serial transfer.

The stop condition signal is detected by hardware incorporated in the slave device.

Figure 17-19. Stop Condition



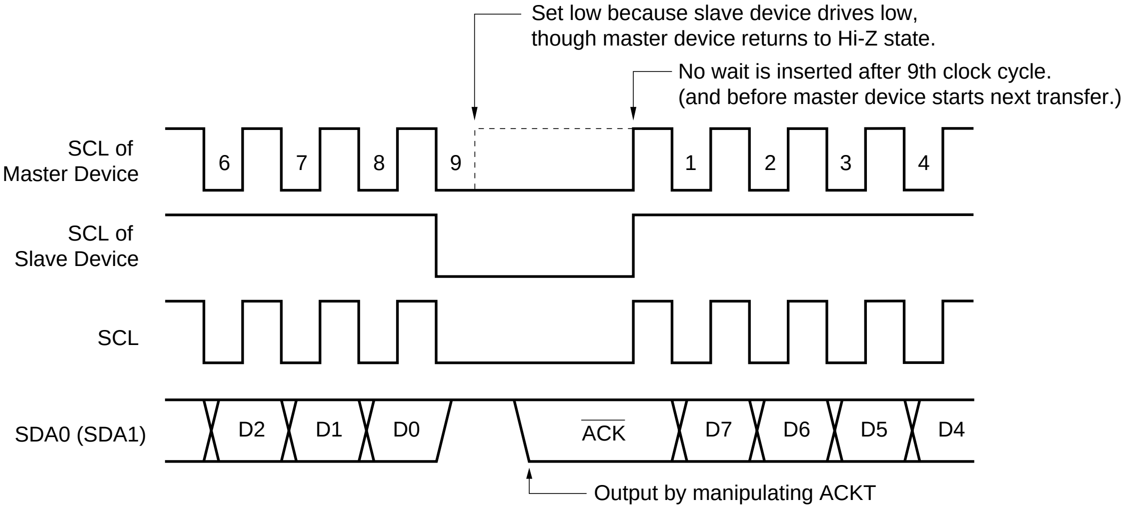
(f) Wait signal ($\overline{\text{WAIT}}$)

The wait signal is output by a slave device to inform the master device that the slave device is in wait state due to preparing for transmitting or receiving data.

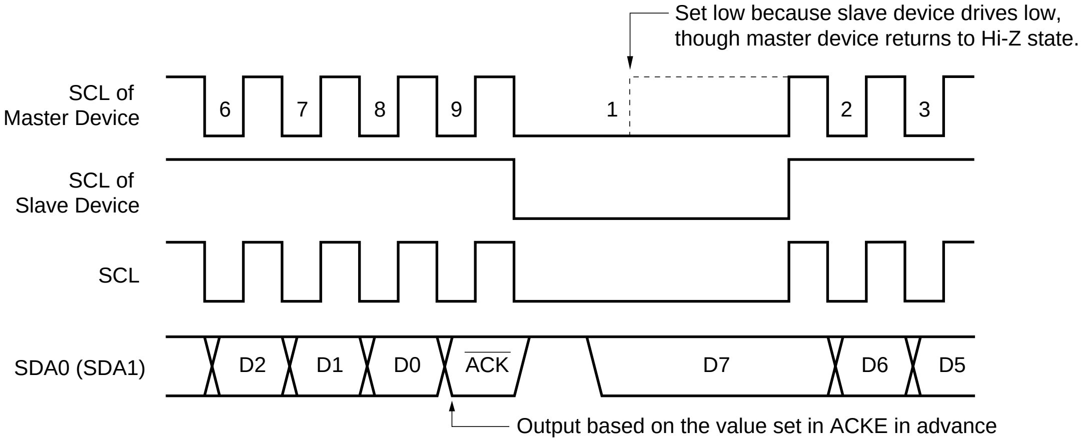
During the wait state, the slave device continues to output the wait signal by keeping the SCL pin low to delay subsequent transfers. When the wait state is released, the master device can start the next transfer. For the releasing operation of slave devices, see **17.4.5 Cautions on Use of I^{2}C Bus Mode**.

Figure 17-20. Wait Signal

(a) Wait of 8 Clock Cycles


Set low because slave device drives low,
though master device returns to Hi-Z state.
No wait is inserted after 9th clock cycle.
(and before master device starts next transfer.)
SCL of Master Device
6
7
8
9
1
2
3
4
SCL of Slave Device
SCL
SDA0 (SDA1)
D2
D1
D0
ACK
D7
D6
D5
D4
Output by manipulating ACKT

(b) Wait of 9 Clock Cycles


Set low because slave device drives low,
though master device returns to Hi-Z state.
SCL of Master Device
6
7
8
9
1
2
3
SCL of Slave Device
SCL
SDA0 (SDA1)
D2
D1
D0
ACK
D7
D6
D5
Output based on the value set in ACKE in advance

(3) Register setting

The I^2C mode setting is performed by the serial operating mode register 0 (CSIM0), the serial bus interface control register (SBIC), and the interrupt timing specify register (SINT).

(a) Serial operating mode register 0 (CSIM0)

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM0 to 00H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W Note 1

R/W

CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection
0	×	Input clock from off-chip to SCL pin
1	0	8-bit timer register 2 (TM2) output Note 2
1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operation mode	Start bit	SI0/SB0/SDA0/ P25 pin function	SO0/SB1/SDA1/ P26 pin function	$\overline{\text{SCK0}}$/SCL/P27 pin function
0	×	3-wire serial I/O mode (see **17.4.2 Operation in 3-wire serial I/O mode**)											
1	1	0	× Note 3	× Note 3	0	0	0	1	2-wire serial I/O or I^2C bus mode	MSB	P25 (CMOS I/O)	SB1/SDA1 N-ch open-drain I/O	$\overline{\text{SCK0}}$/SCL N-ch open-drain I/O
1	1	1	0	0	× Note 3	× Note 3	0	1	2-wire serial I/O or I^2C bus mode	MSB	SB0/SDA0 N-ch open-drain I/O	P26 (CMOS I/O)	$\overline{\text{SCK0}}$/SCL N-ch open-drain I/O

R/W

WUP	Wake-up Function Control Note 4
0	Interrupt request signal generation with each serial transfer in any mode
1	In I^2C bus mode, interrupt request signal is generated when the address data received after start condition detection (when CMDD = 1) matches data in slave address register (SVA).

R

COI	Slave Address Comparison Result Flag Note 5
0	Slave address register (SVA) not equal to data in serial I/O shift register 0 (SIO0)
1	Slave address register (SVA) equal to data in serial I/O shift register 0 (SIO0)

R/W

CSIE0	Serial Interface Channel 0 Operation Control
0	Stops operation.
1	Enables operation.

Notes 1. Bit 6 (COI) is a read-only bit.

2. In the I^2C bus mode, the clock frequency is 1/16 of the clock frequency output by TO2.

3. Can be used freely as a port.

4. To use the wake-up function (WUP = 1), set the bit 5 (SIC) of the interrupt timing specify register (SINT) to 1. Do not execute an instruction that writes the serial I/O shift register 0 (SIO0) while WUP = 1.

5. When CSIE0 = 0, COI is 0.

Remark × : don't care

PM×× : Port mode register

P×× : Port output latch

(b) Serial bus interface control register (SBIC)

SBIC is set with a 1-bit or 8-bit memory manipulation instruction.

RESET input clears SBIC to 00H.

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W Note 1

R/W	RELT	Use for stop condition output. When RELT = 1, SO0 latch is set to 1. After SO0 latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W	CMDT	Use for start condition output. When CMDT = 1, SO0 latch is cleared to 0. After clearing SO0 latch, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R	RELD	Stop Condition Detection
	0	Clear Conditions • When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception • When CSIE0 = 0 • When RESET input is applied
	1	Setting Condition • When stop condition is detected

R	CMDD	Start Condition Detection
	0	Clear Conditions • When transfer start instruction is executed • When stop condition is detected • When CSIE0 = 0 • When RESET input is applied
	1	Setting Condition • When start condition is detected

R/W	ACKT	SDA0 (SDA1) is set to low after the Set instruction execution (ACKT = 1) before the next SCL falling edge. Used for generating an ACK signal by software if the 8-clock wait mode is selected. Cleared to 0 if CSIE = 0 when a transfer by the serial interface is started.

R/W	ACKE	Acknowledge Signal Automatic Output Control Note 2
	0	Disabled (with ACKT enabled). Used when receiving data in the 8-clock wait mode or when transmitting data. Note 3
	1	Enabled. After completion of transfer, acknowledge signal is output in synchronization with the 9th falling edge of SCL clock (automatically output when ACKE = 1). However, not automatically cleared to 0 after acknowledge signal output. Used for reception when the 9-clock wait mode is selected.

R	ACKD	Acknowledge Detection
	0	Clear Conditions • When transfer start instruction is executed • When CSIE0 = 0 • When RESET input is applied
	1	Set Conditions • When acknowledge signal is detected at the rising edge of SCL clock after completion of transfer

R/W	Note 4 BSYE	Control of N-ch Open-Drain Output for Transmission in I^2C Bus Mode Note 5
	0	Output enabled (transmission)
	1	Output disabled (reception)

Notes **1.** Bits 2, 3, and 6 (RELD, CMDD, ACKD) are read-only bits.

2. This setting must be performed prior to transfer start.

3. In the 8-clock wait mode, use ACKT for output of the acknowledge signal after normal data reception.

4. The busy mode can be released by the start of a serial interface transfer or reception of an address signal. However, the BSYE flag is not cleared.

5. When using the wake-up function, be sure to set BSYE to 1.

Remark Bit 7 of serial operating mode register 0 (CSIM0)

(c) Interrupt timing specification register (SINT)

SINT is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears SINT to 00H.

Symbol	7	⑥	⑤	④	③	②	1	0	Address	After Reset	R/W
SINT	0	CLD	SIC	SVAM	CLC	WREL	WAT1	WAT0	FF63H	00H	R/W Note 1

R/W

WAT1	WAT0	Interrupt Control by Wait Note 2
0	0	Interrupt service request is generated on rise of 8th $\overline{\text{SCK0}}$ clock cycle (clock output is high impedance).
0	1	Setting prohibited
1	0	Used in I^{2}C bus mode (8-clock wait) Generates an interrupt service request on rise of 8th SCL clock cycle. (In case of master device, SCL pin is driven low after output of 8 clock cycles, to enter the wait state. In case of slave device, SCL pin is driven low after input of 8 clock cycles, to require the wait state.)
1	1	Used in I^{2}C bus mode (9-clock wait) Generates an interrupt service request on rise of 9th SCL clock cycle. (In case of master device, SCL pin is driven low after output of 9 clock cycles, to enter the wait state. In case of slave device, SCL pin is driven low after input of 9 clock cycles, to require the wait state.)

R/W

WREL	Wait Release Control
0	Indicates that the wait state has been released.
1	Releases the wait state. Automatically cleared to 0 after releasing the wait state. This bit is used to release the wait state set by means of WAT0 and WAT1.

R/W

CLC	Clock Level Control
0	Used in I^{2}C bus mode. In cases other than serial transfer, SCL pin output is driven low.
1	Used in I^{2}C bus mode. In cases other than serial transfer, SCL pin output is set to high impedance. (Clock line is held high.) Used by master device to generate the start condition and stop condition signals.

R/W

SVAM	SVA Bits Used as Slave Address
0	Bits 0 to 7
1	Bits 1 to 7

R/W

SIC	INTCSI0 Interrupt Source Selection Note 3
0	CSIIF0 is set to 1 after end of serial interface channel 0 transfer.
1	CSIIF0 is set to 1 after end of serial interface channel 0 transfer or when stop condition is detected.

R

CLD	SCL Pin Level Note 4
0	Low level
1	High level

Notes 1. Bit 6 (CLD) is read-only.

2. When the I^{2}C bus mode is used, be sure to set 1 and 0, or 1 and 1 in WAT0 and WAT1, respectively.

3. When using the wake-up function in I^{2}C mode, be sure to set SIC to 1.

4. When CSIE0 = 0, CLD is 0.

Remark SVA : Slave address register

CSIIF0 : Interrupt request flag corresponding to INTCSI0

CSIE0 : Bit 7 of serial operating mode register 0 (CSIM0)

(4) Various signals

A list of signals in the I^{2}C bus mode is given in Table 17-4.

Table 17-4. Signals in I^{2}C Bus Mode

Signal name	Description	
Start condition	Definition :	SDA0 (SDA1) falling edge when SCL is high **Note 1**
	Function :	Indicates that serial communication starts and subsequent data are address data.
	Signaled by :	Master
	Signaled when :	CMDT is set.
	Affected flag(s) :	CMDD (is set.)
Stop condition	Definition :	SDA0 (SDA1) rising edge when SCL is high **Note 1**
	Function :	Indicates end of serial transmission.
	Signaled by :	Master
	Signaled when :	RELT is set.
	Affected flag(s) :	RELD (is set) and CMDD (is cleared)
Acknowledge signal ($\overline{\text{ACK}}$)	Definition :	Low level of SDA0 (SDA1) pin during one SCL clock cycle after serial reception
	Function :	Indicates completion of reception of 1 byte.
	Signaled by :	Master or slave
	Signaled when :	ACKT is set with ACKE = 1.
	Affected flag(s) :	ACKD (is set.)
Wait ($\overline{\text{WAIT}}$)	Definition :	Low-level signal output to SCL
	Function :	Indicates state in which serial reception is not possible.
	Signaled by :	Slave
	Signaled when :	WAT1, WAT0 = 1x.
	Affected flag(s) :	None
Serial clock (SCL)	Definition :	Synchronization clock for output of various signals
	Function :	Serial communication synchronization signal.
	Signaled by :	Master
	Signaled when :	See **Note 2** below.
	Affected flag(s) :	CSIIF0. Also see **Note 3** below.
Address (A6 to A0)	Definition :	7-bit data synchronized with SCL immediately after start condition signal
	Function :	Indicates address value for specification of slave on serial bus.
	Signaled by :	Master
	Signaled when :	See **Note 2** below.
	Affected flag(s) :	CSIIF0. Also see **Note 3** below.
Transfer direction (R/$\overline{\text{W}}$)	Definition :	1-bit data output in synchronization with SCL after address output
	Function :	Indicates whether data transmission or reception is to be performed.
	Signaled by :	Master
	Signaled when :	See **Note 2** below.
	Affected flag(s) :	CSIIF0. Also see **Note 3** below.
Data (D7 to D0)	Definition :	8-bit data synchronized with SCL, not immediately after start condition
	Function :	Contains data actually to be sent.
	Signaled by :	Master or slave
	Signaled when :	See **Note 2** below.
	Affected flag(s) :	CSIIF0. Also see **Note 3** below.

Notes 1. The level of the serial clock can be controlled by bit 3 (CLC) of interrupt timing specify register (SINT).

2. Execution of instruction to write data to SIO0 when CSIE0 = 1 (serial transfer start directive). In the wait state, the serial transfer operation will be started after the wait state is released.

3. If the 8-clock wait is selected when WUP = 0, CSIIF0 is set at the rising edge of the 8th clock cycle of SCL. If the 9-clock wait is selected when WUP = 0, CSIIF0 is set at the rising edge of the 9th clock cycle of SCL. CSIIF0 is set if an address is received and that address coincides with the value of the slave address register (SVA) when WUP = 1, or if the stop condition is detected.

(5) Pin configurations

The configurations of the serial clock pin SCL and the serial data bus pins SDA0 (SDA1) are shown below.

(a) SCL

Pin for serial clock input/output alternate function pin.

<1> Master ..... N-ch alternate drain output

<2> Slave .... Schmitt input

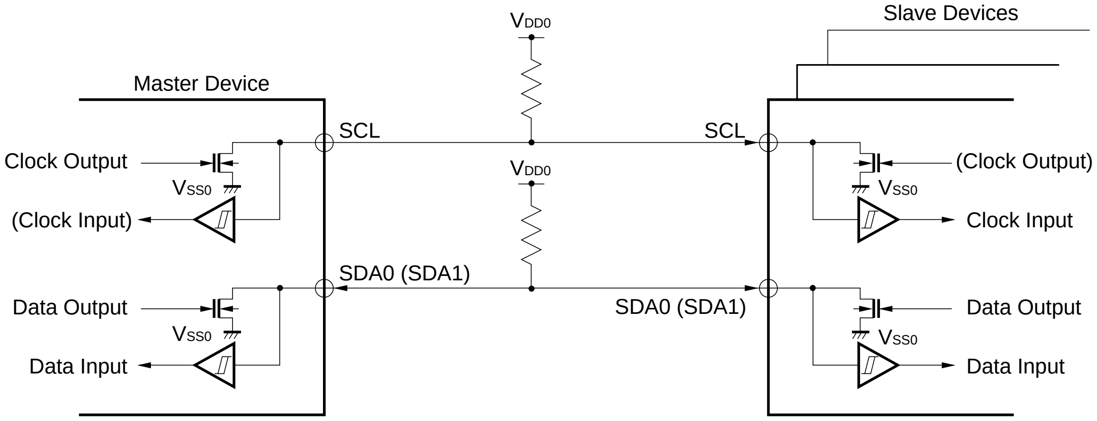
(b) SDA0 (SDA1)

Serial data input/output alternate function pin.

Uses N-ch open-drain output and Schmitt-input buffers for both master and slave devices.

Note that pull-up resistors are required to connect to both serial clock line and serial data bus line, because open-drain buffers are used for the serial clock pin (SCL) and the serial data bus pin (SDA0 or SDA1) on the I^{2}C bus.

Figure 17-21. Pin Configuration



Caution To receive data, the N-ch open-drain output must made to go into a high-impedance state. Therefore, set the bit 7 (BSYE) of the serial bus interface control register (SBIC) to 1 in advance, and write FFH to the serial I/O shift register 0 (SIO0).

When the wake-up function is used (by setting the bit 5 (WUP) of the serial operating mode register 0 (CSIM0)), however, do not write FFH to SIO0 before reception. Even if FFH is not written to SIO0, the N-ch open-drain output always goes into a high-impedance state.

(6) Address match detection method

In the I^{2}C mode, the master can select a specific slave device by sending slave address data.

Coincidence of the addresses can be automatically detected by hardware. CSIIF0 is set if the slave address transmitted by the master coincides with the value set to the slave address register (SVA) when a slave device address has a slave register (SVA), and the wake-up function specify bit (WUP) = 1 (CSIIF0 is also set when the stop condition is detected).

When using the wake-up function, set SIC to 1.

Caution Slave selection/non-selection is detected by matching of the data (address) received after the start condition.

For this match detection, match interrupt request (INTCSI0) of the address to be generated with WUP = 1 is normally used. Thus, execute selection/non-selection detection by slave address when WUP = 1.

(7) Error detection

In the I^{2}C bus mode, transmission error detection can be performed by the following methods because the serial bus SDA0 (SDA1) status during transmission is also taken into the serial I/O shift register 0 (SIO0) register of the transmitting device.

(a) Comparison of SIO0 data before and after transmission

In this case, a transmission error is judged to have occurred if the two data values are different.

(b) Using the slave address register (SVA)

Transmit data is set in SIO0 and SVA before transmission is performed. After transmission, the COI bit (match signal from the address comparator) of serial operating mode register 0 (CSIM0) is tested: "1" indicates normal transmission, and "0" indicates a transmission error.

(8) Communication operation

In the I^{2}C bus mode, the master selects the slave device to be communicated with from among multiple devices by outputting address data onto the serial bus.

After the slave address data, the master sends the R/W bit which indicates the data transfer direction, and starts serial communication with the selected slave device.

Data communication timing charts are shown in **Figures 17-22** and **17-23**.

In the transmitting device, the serial I/O shift register 0 (SIO0) shifts transmission data to the SO latch in synchronization with the falling edge of the serial clock (SCL), the SO0 latch outputs the data on an MSB-first basis from the SDA0 or SDA1 pin to the receiving device.

In the receiving device, the data input from the SDA0 or SDA1 pin is taken into the SIO0 in synchronization with the rising edge of SCL.

Figure 17-22. Data Transmission from Master to Slave (Both Master and Slave Selected 9-clock Wait) (1/3)

(a) Start Condition to Address

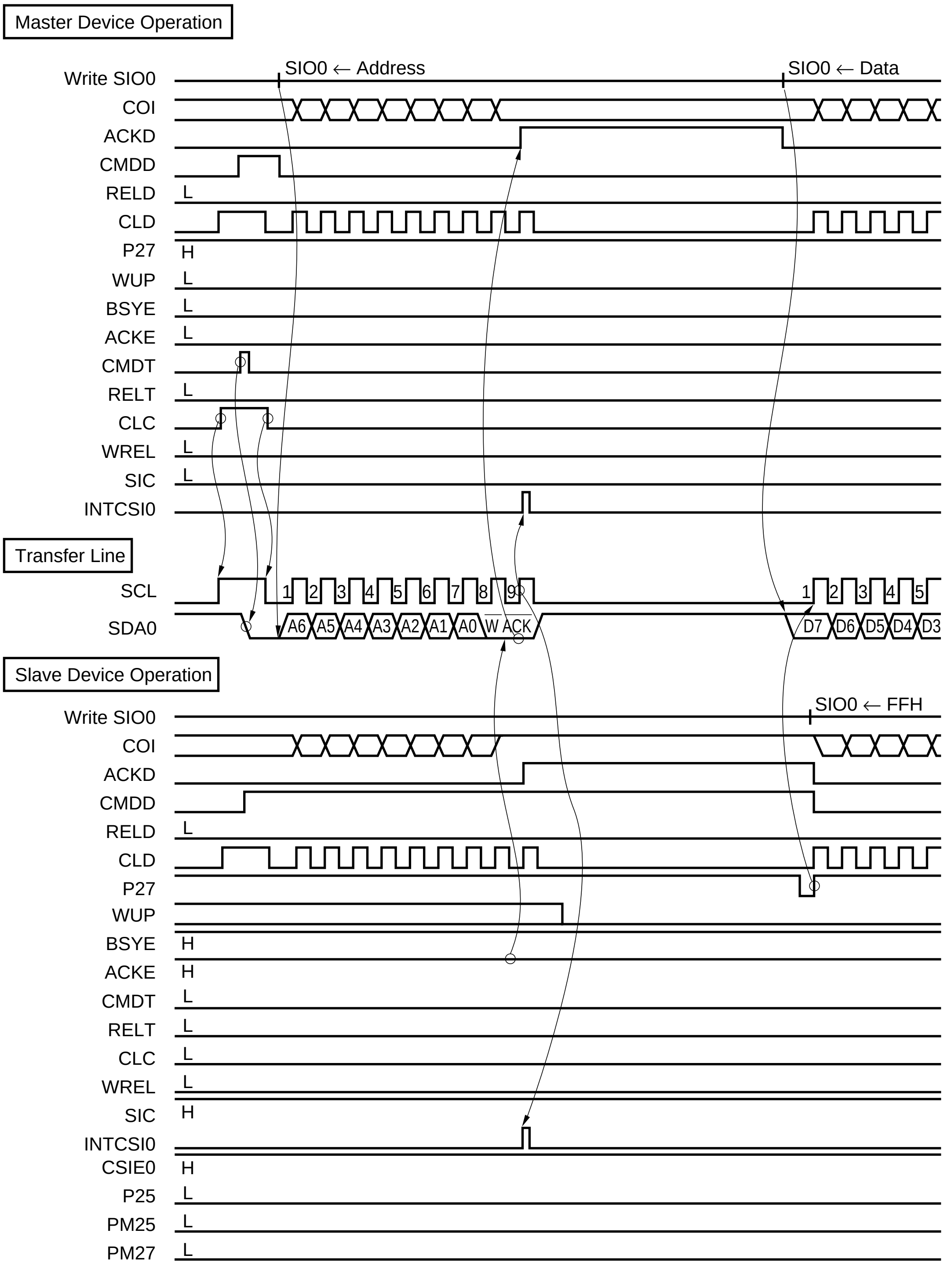


Figure 17-22. Data Transmission from Master to Slave (Both Master and Slave Selected 9-clock Wait) (2/3)

(b) Data

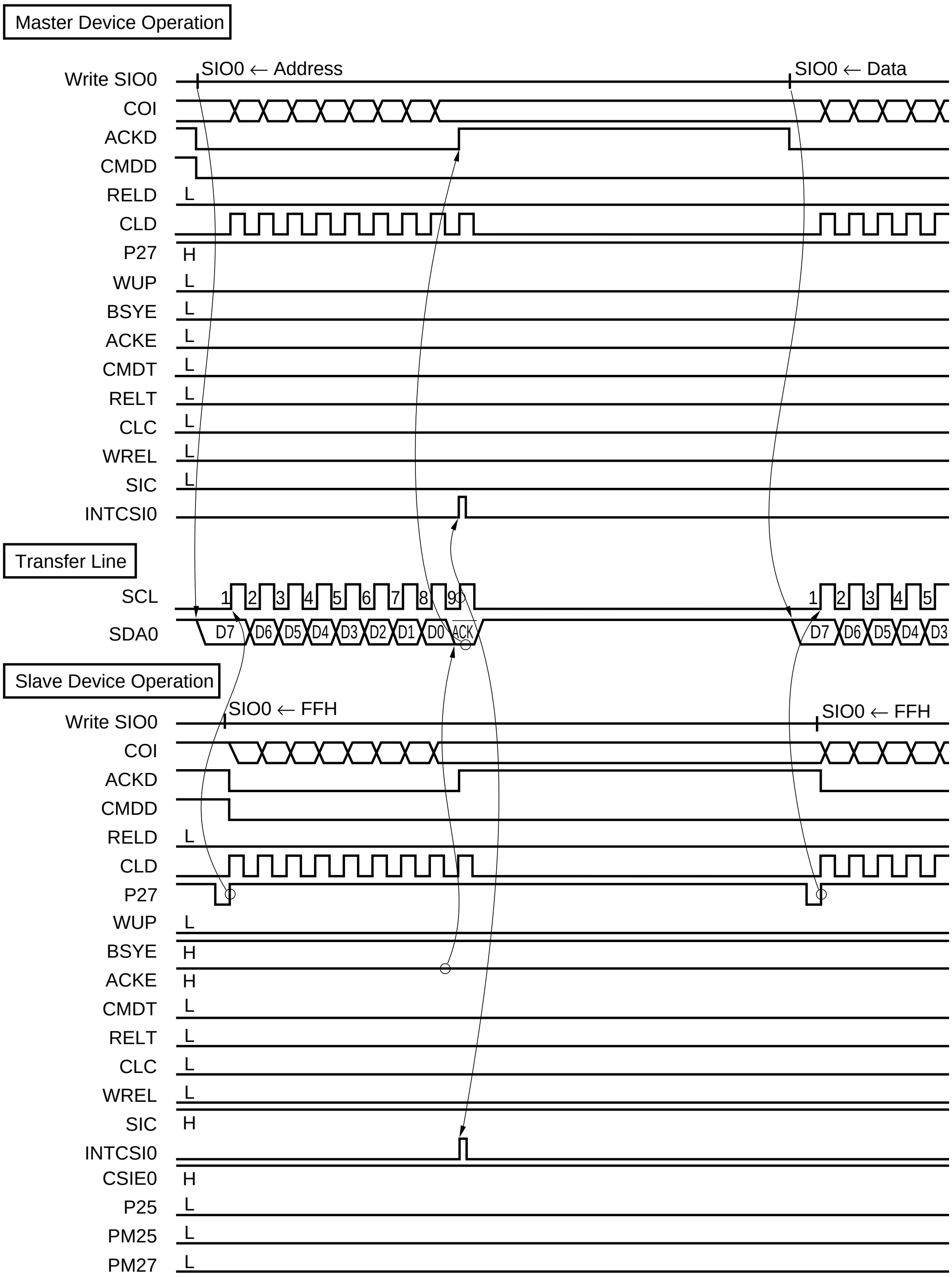


Figure 17-22. Data Transmission from Master to Slave (Both Master and Slave Selected 9-clock Wait) (3/3)

(c) Stop Condition

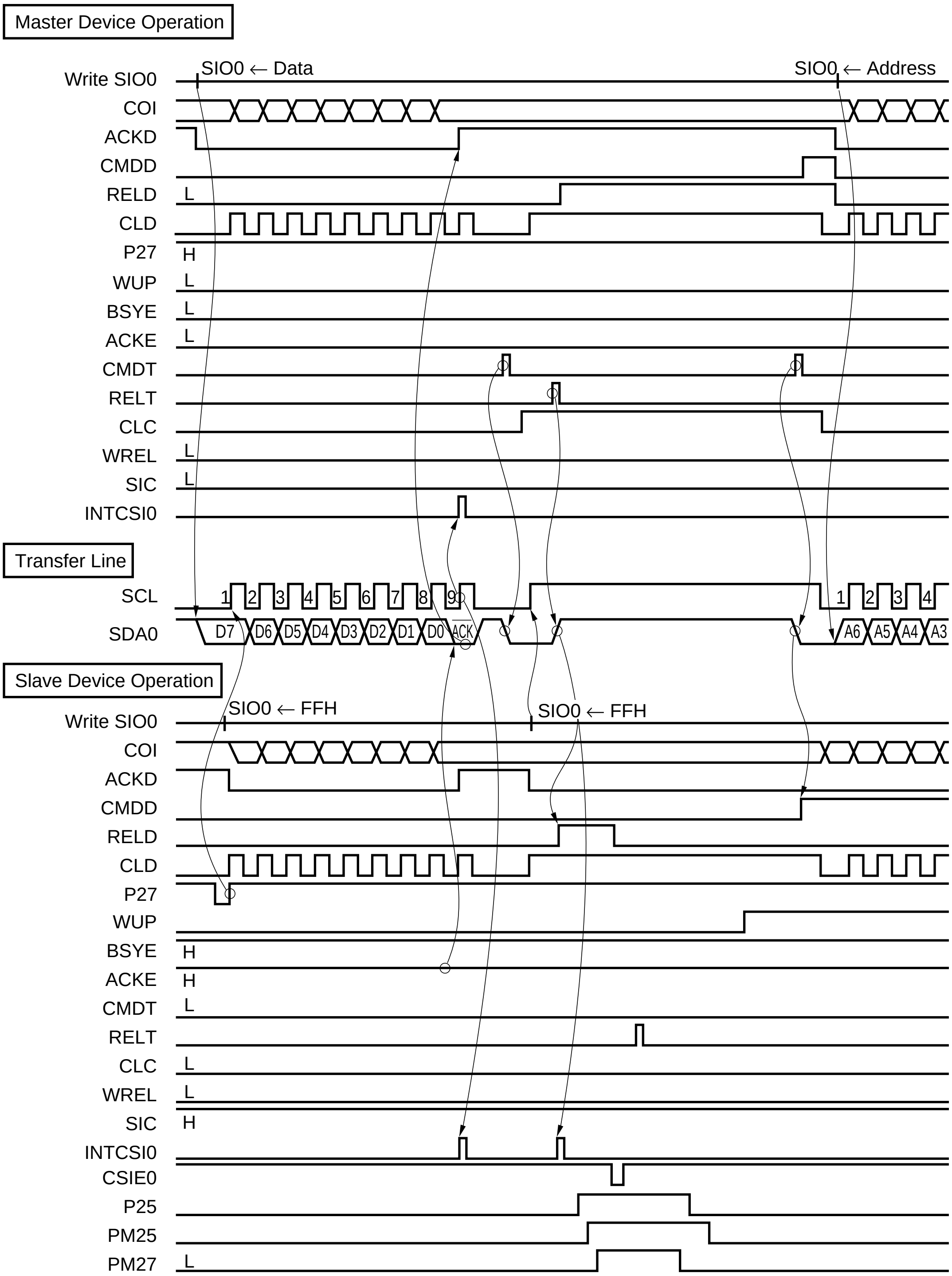


Figure 17-23. Data Transmission from Slave to Master (Both Master and Slave Selected 9-clock Wait) (1/3)

(a) Start Condition to Address

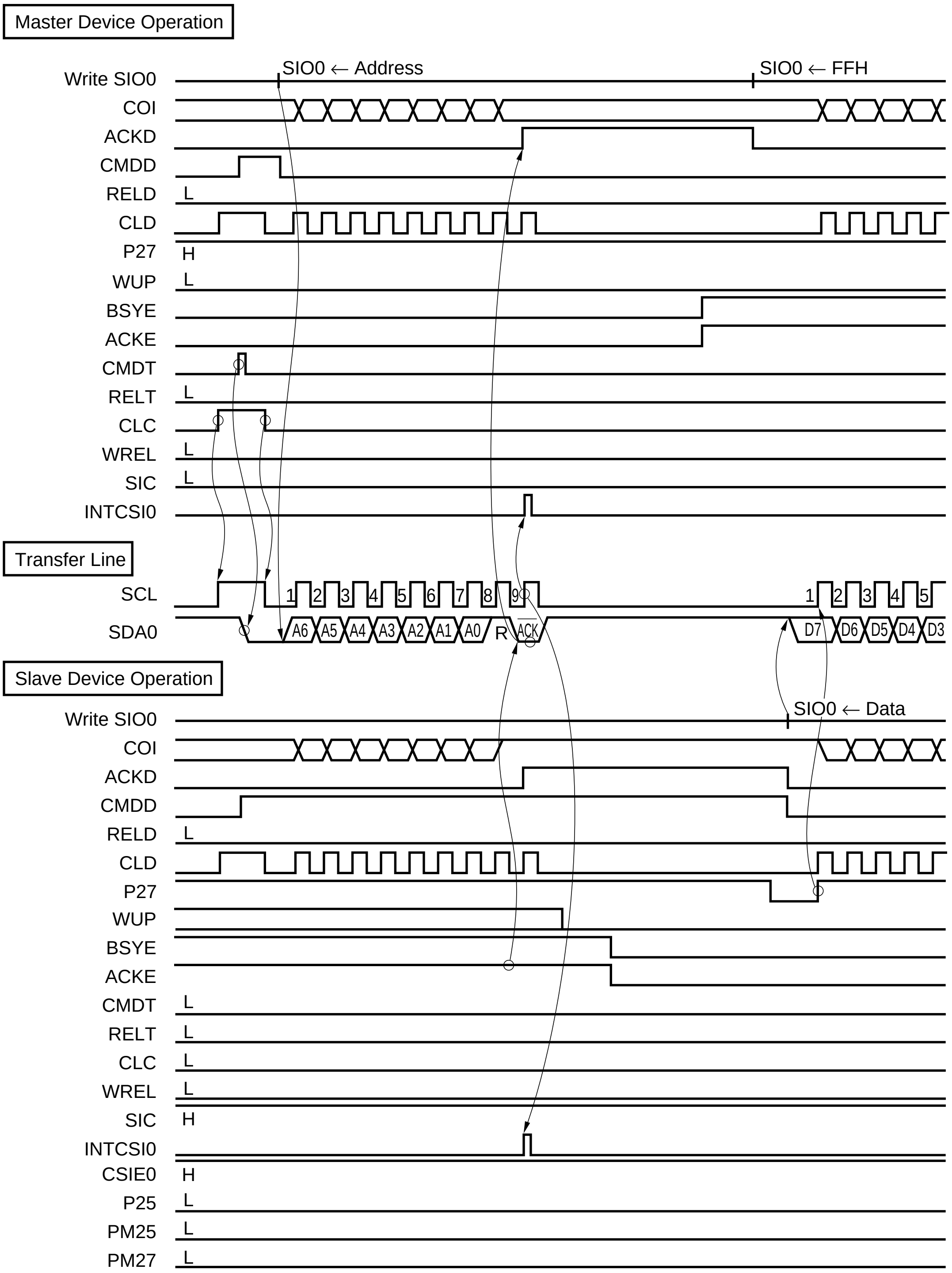


Figure 17-23. Data Transmission from Slave to Master (Both Master and Slave Selected 9-clock Wait) (2/3)

(b) Data

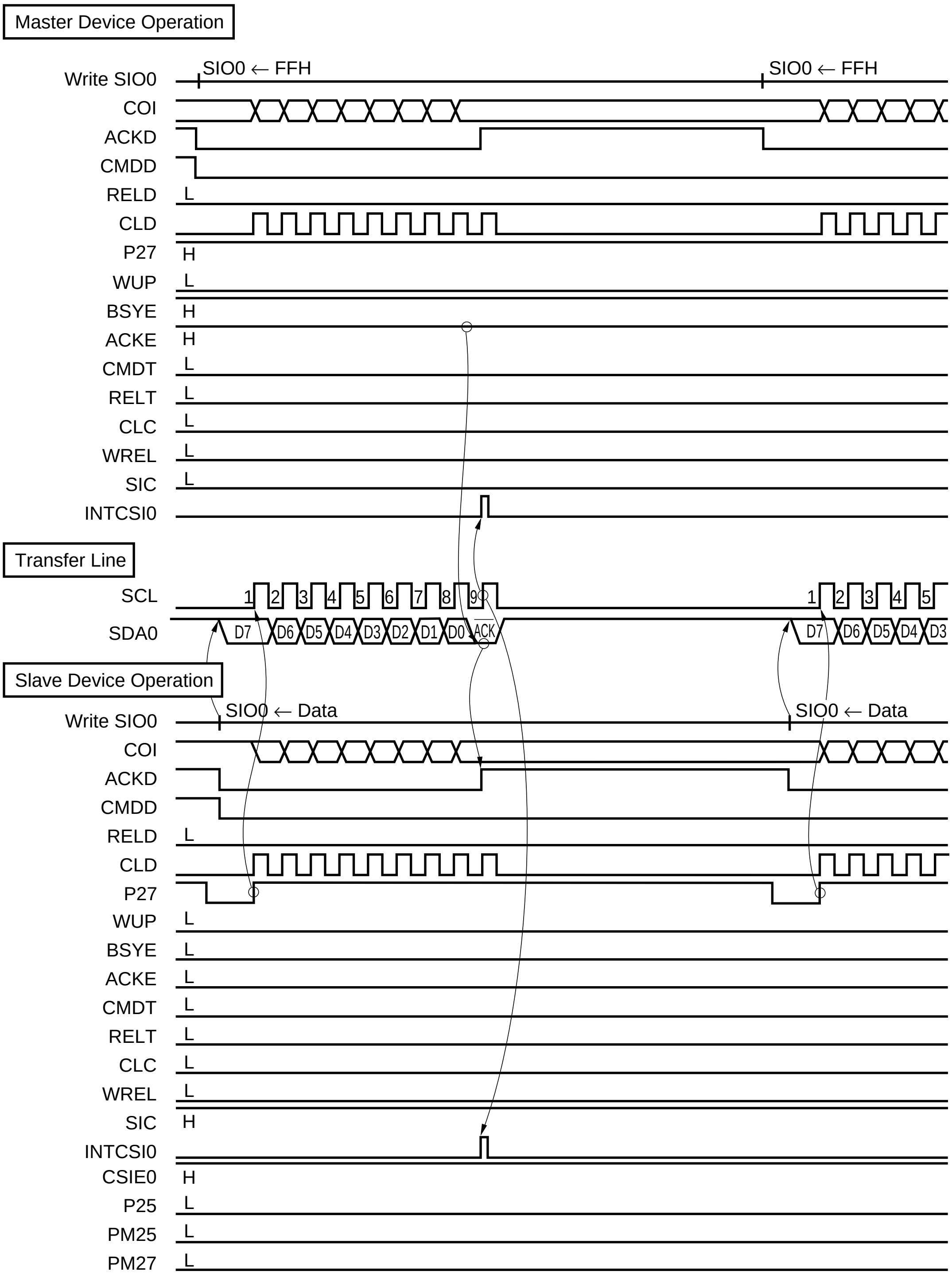
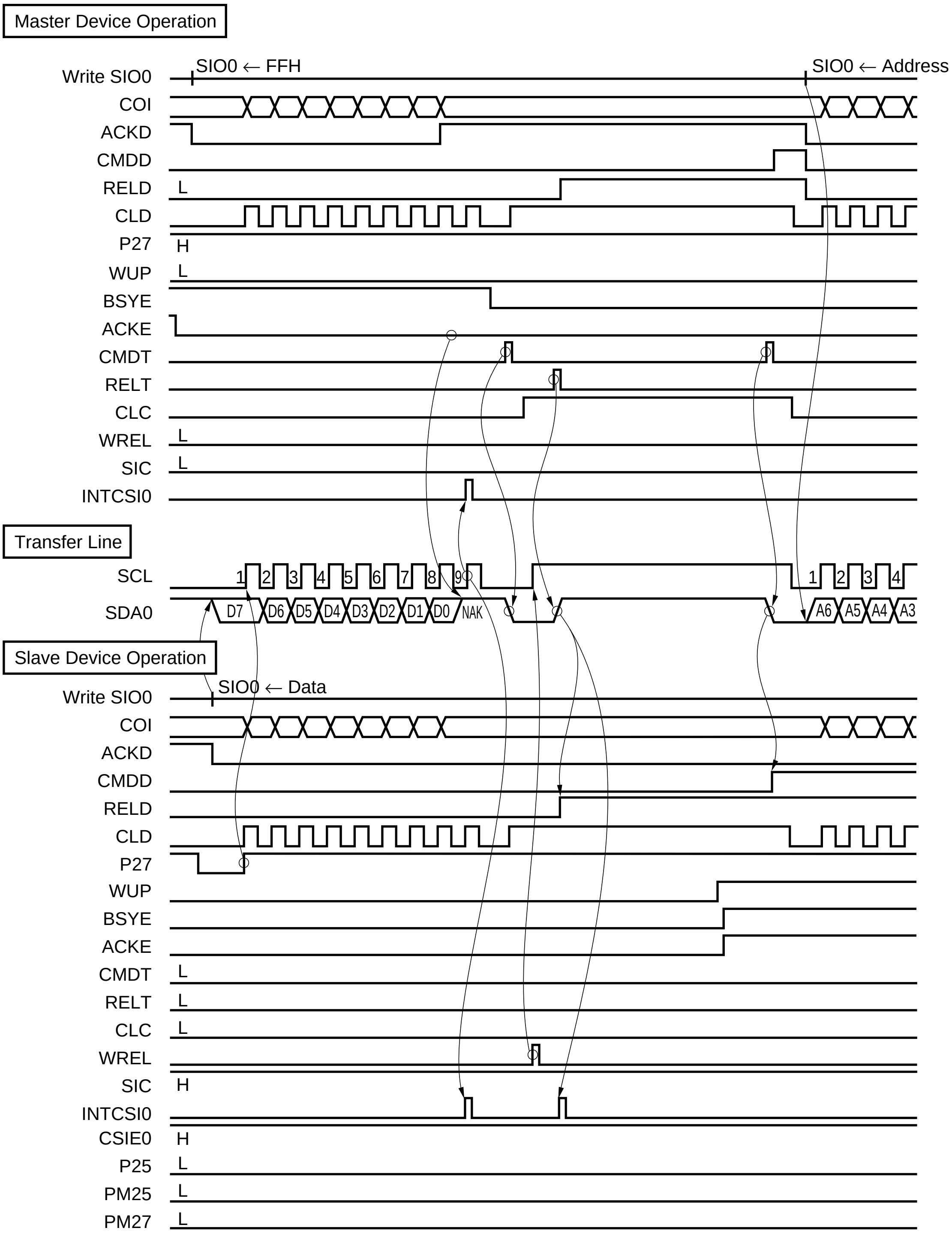


Figure 17-23. Data Transmission from Slave to Master (Both Master and Slave Selected 9-clock Wait) (3/3)

(c) Stop Condition



(9) Transfer start

A serial transfer is started by setting transfer data in serial I/O shift register 0 (SIO0) if the following two conditions have been satisfied:

- The serial interface channel 0 operation control bit (CSIE0) = 1.
- After an 8-bit serial transfer, the internal serial clock is stopped or SCL is low.

Cautions 1. Be sure to set CSIE0 to 1 before writing data in SIO0. Setting CSIE0 to 1 after writing data in SIO0 does not initiate transfer operation.

2. Because the N-ch open-drain output must made to go into a high-impedance state during data reception, set bit 7 (BSYE) of serial bus interface control register (SBIC) to 1 before writing FFH to SIO0.

Do not write FFH to SIO0 before reception when the wake-up function is used (by setting the bit 5 (WUP) of the serial operating mode register 0 (CSIM0)). Even if FFH is not written to SIO0, the N-ch open-drain output always goes into a high-impedance state.

3. If data is written to SIO0 while the slave is in the wait state, that data is held. The transfer is started when SCL is output after the wait state is cleared.

When an 8-bit data transfer ends, serial transfer is stopped automatically and the interrupt request flag (CSIIF0) is set.

17.4.5 Cautions on use of I^{2}C bus mode

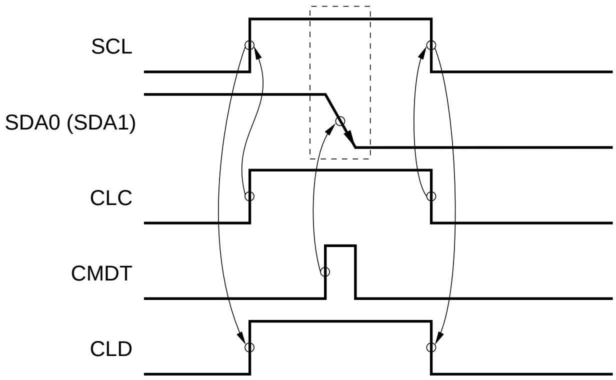
(1) Start condition output (master)

The SCL pin normally outputs a low-level signal when no serial clock is output. It is necessary to change the SCL pin to high in order to output a start condition signal. Set 1 in CLC of interrupt timing specify register (SINT) to drive the SCL pin high.

After setting CLC, clear CLC to 0 and return the SCL pin to low. If CLC remains 1, no serial clock is output.

If it is the master device which outputs the start condition and stop condition signals, confirm that CLD is set to 1 after setting CLC to 1; a slave device may have set SCL to low (wait state).

Figure 17-24. Start Condition Output



(2) Slave wait release (slave transmission)

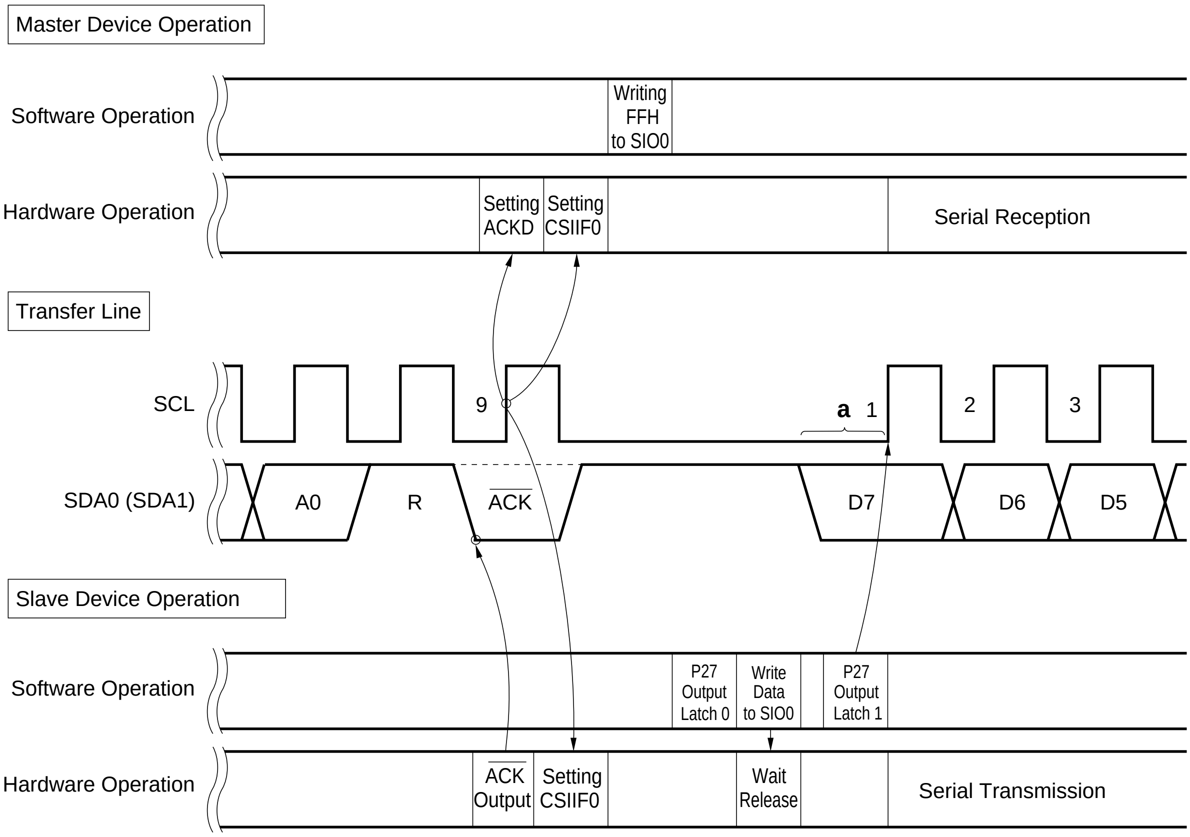
Slave wait status is released by WREL flag (bit 2 of interrupt timing specify register (SINT)) setting or execution of an serial I/O shift register 0 (SIO0) write instruction.

If the slave sends data, the wait is immediately released by execution of an SIO0 write instruction and the clock rises without the start transmission bit being output in the data line. Therefore, as shown in Figure 17-25, data should be transmitted by manipulating the P27 output latch through the program. At this time, control the low-level width ("**a**" in **Figure 17-25**) of the first serial clock at the timing used for setting the P27 output latch to 1 after execution of an SIO0 write instruction.

In addition, if the acknowledge signal from the master is not output (if data transmission from the slave is completed), set 1 in the WREL flag of SINT and release the wait.

For these timings, see **Figure 17-23**.

Figure 17-25. Slave Wait Release (Transmission)



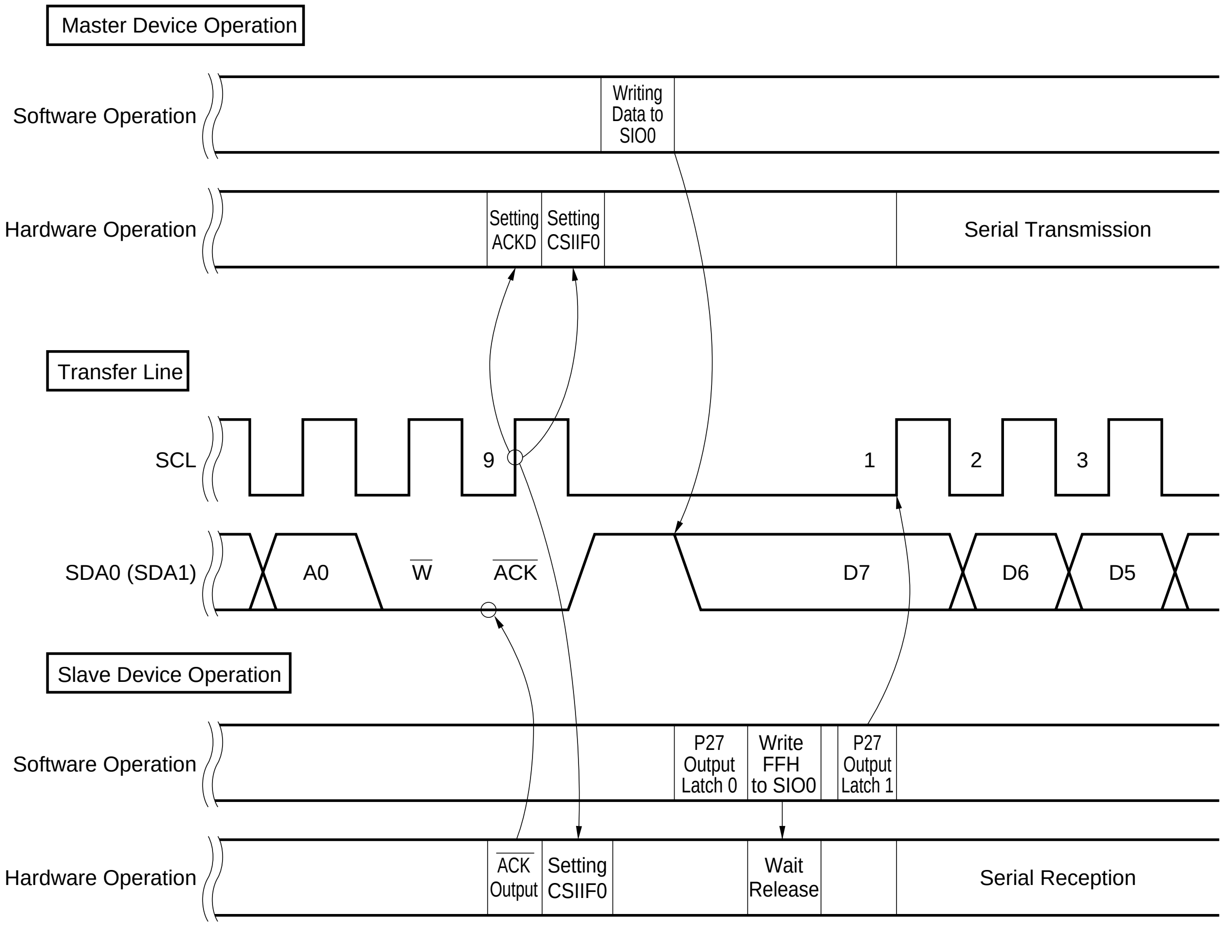
(3) Slave wait release (slave reception)

The slave is released from the wait status when the WREL flag (bit 2 of the interrupt timing specify register (SINT)) is set or when an instruction that writes data to the serial I/O shift register 0 (SIO0) is executed. When the slave receives data, the first bit of the data sent from the master may not be received if the SCL line immediately goes into a high-impedance state after an instruction that writes data to SIO has been executed.

This is because SIO0 does not start operating if the SCL line is in the high-impedance state while the instruction that writes data to SIO0 is executed (until the next instruction is executed).

Therefore, receive the data by manipulating the output latch of P27 by program, as shown in Figure 17-26. For this timing, refer to **Figure 17-22**.

Figure 17-26. Slave Wait Release (Reception)



(4) Reception completion of salve

In the reception completion processing of the slave, check the bit 3 (CMDD) of the serial bus interface control register (SBIC) and bit 6 (COI) of the serial operation mode register 0 (CSIM0) (when CMDD = 1). This is to avoid the situation where the slave cannot judge which of the start condition and data comes first and therefore, the wake-up condition cannot be used when the slave receives the undefined number of data from the master.

17.4.6 Restrictions in I^{2}C bus mode

The following restrictions are applied to the μPD780058Y Subseries.

- **Restrictions when used as slave device in I^{2}C bus mode**

Subject: μPD780053Y, 780054Y, 780055Y, 780056Y, 780058Y, 78F0058Y, IE-780308-R-EM, IE-780308-NS-EM1

Description: If the wake-up function is executed (by setting the bit 5 of the serial operating mode register 0 (CSIM0) to 1) in the serial transfer status **Note**, the μPD780058Y Subseries checks the address of the data between the other slave and master. If that data happens to coincide with the slave address of the μPD780058Y Subseries, the μPD780058Y Subseries takes part in communication, destroying the communication data.

Note The serial transfer status is the status since data has been written to the serial I/O shift register 0 (SIO0) until the interrupt request flag (CSIIF0) is set to 1 by completion of the serial transfer.

Preventive measure: The above phenomenon can be avoided by modifying the program.

Before executing the wake-up function, execute the following program that clears the serial transfer status. When executing the wake-up function, do not execute an instruction that writes data to SIO0. Even if such an instruction is not executed, data can be received while the wake-up function is executed.

This program releases the serial transfer status. To release the serial transfer status, the serial interface channel 0 must be once disabled (by clearing the CSIE0 flag (bit 7 of the serial operating mode register (CSIM0) to 0). If the serial interface channel 0 is disabled in the I^{2}C bus mode, however, the SCL pin outputs a high level, and SDA0 (SDA1) pin outputs a low level, affecting communication of the I^{2}C bus. Therefore, this program makes the SCL and SDA0 (SDA1) pins go into a high-impedance state to prevent the I^{2}C bus from being affected.

In this example, the SDA0 (/P25) pin is used as a serial data input/output pin. When the SDA1 (/P26) is used, take P2.5 and PM2.5 in the program example below as P2.6 and PM2.6.

For the timing of each signal when this program is executed, refer to **Figure 17-22**.

- **Example of program releasing serial transfer status**

```
SET1   P2.5;      <1>
SET1   PM2.5;     <2>
SET1   PM2.7;     <3>
CLR1   CSIE0;     <4>
SET1   CSIE0;     <5>
SET1   RELT;      <6>
CLR1   PM2.7;     <7>
CLR1   P2.5;      <8>
CLR1   PM2.5;     <9>
```

<1> This instruction prevents the SDA0 pin from outputting a low level when the I^{2}C bus mode is restored by instruction **<5>**. The output of the SDA0 pin goes into a high-impedance state.

<2> This instruction sets the P25 (/SDA0) pin in the input mode to protect the SDA0 line from adverse influence when the port mode is set by instruction **<4>**. The P25 pin is set in the input mode when instruction **<2>** is executed.

<3> This instruction sets the P27 (/SCL) pin in the input mode to protect the SCL line from adverse influence when the port mode is set by instruction **<4>**. The P27 pin is set in the input mode when instruction **<3>** is executed.

<4> This instruction changes the mode from I^{2}C bus mode to port mode.

<5> This instruction restores the I^{2}C bus mode from the port mode.

<6> This instruction prevents the SDA0 pin from outputting a low level when instruction **<8>** is executed.

<7> This instruction sets the P27 pin in the output mode because the P27 pin must be in the output mode in the I^{2}C bus mode.

<8> This instruction clears the output latch of the P25 pin to 0 because the output latch of the P25 pin must be set to 0 in the I^{2}C bus mode.

<9> This instruction sets the P25 pin in the output mode because the P25 pin must be in the output mode in the I^{2}C bus mode.

Remark RELT: Bit 0 of serial bus interface control register (SBIC)

17.4.7 $\overline{\text{SCK0}}$/SCL/P27 pin output manipulation

The $\overline{\text{SCK0}}$/SCL/P27 pin can execute static output via software, in addition to outputting the normal serial clock.

The value of serial clocks can also be arbitrarily set by software (the SI0/SB0/SDA0 and SO0/SB1/SDA1 pins are controlled with the bit 0 (RELT) and bit 1 (CMDT) of serial bus interface control register (SBIC)).

The $\overline{\text{SCK0}}$/SCL/P27 pin output should be manipulated as described below.

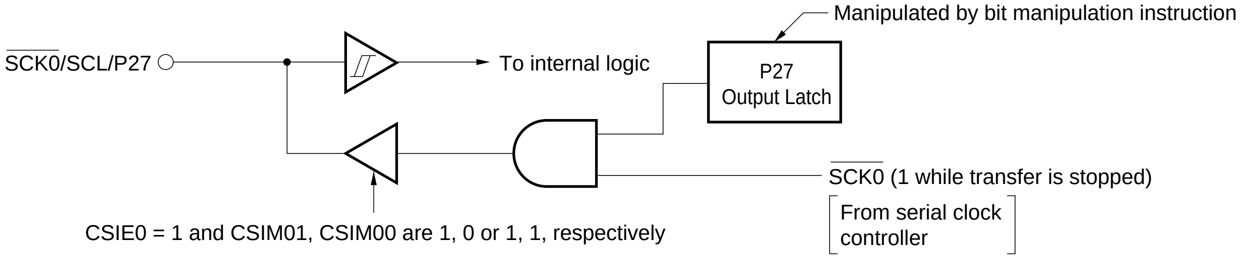
(1) In 3-wire serial I/O mode and 2-wire serial I/O mode

The output level of the $\overline{\text{SCK0}}$/SCL/P27 pin is manipulated by the P27 output latch.

<1> Set the serial operating mode register 0 (CSIM0) ($\overline{\text{SCK0}}$ pin is set in the output mode and serial operation is enabled). $\overline{\text{SCK0}}$ = 1 while serial transfer is stopped.

<2> Manipulate the content of the P27 output latch by executing the bit manipulation instruction.

Figure 17-27. $\overline{\text{SCK0}}$/SCL/P27 Pin Configuration


Manipulated by bit manipulation instruction
$\overline{\text{SCK0}}$/SCL/P27
To internal logic
P27
Output Latch
$\overline{\text{SCK0}}$ (1 while transfer is stopped)
From serial clock controller
CSIE0 = 1 and CSIM01, CSIM00 are 1, 0 or 1, 1, respectively

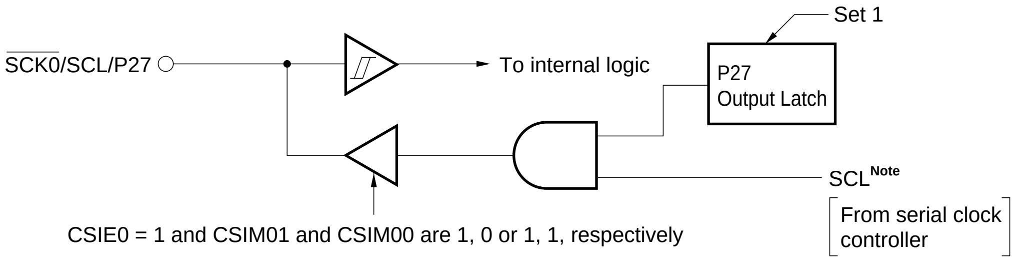
(2) In I^{2}C bus mode

The output level of the $\overline{\text{SCK0}}$/SCL/P27 pin is manipulated by the CLC bit of the interrupt timing specify register (SINT).

<1> Set the serial operating mode register 0 (CSIM0) (SCL pin is set in the output mode and serial operation is enabled). Set 1 to the P27 output latch. SCL = 0 while serial transfer is stopped.

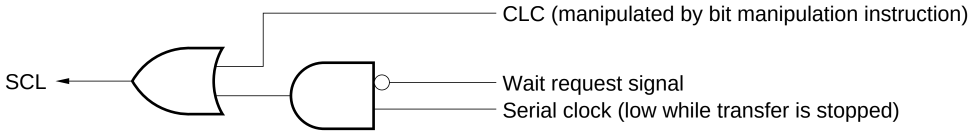
<2> Manipulate the CLC bit of SINT by executing the bit manipulation instruction.

Figure 17-28. $\overline{\text{SCK0}}$/SCL/P27 Pin Configuration


Set 1
$\overline{\text{SCK0}}$/SCL/P27
To internal logic
P27
Output Latch
SCL Note
From serial clock controller
CSIE0 = 1 and CSIM01 and CSIM00 are 1, 0 or 1, 1, respectively

Note The level of the SCL signal is in accordance with the contents of the logic circuits shown in Figure 17-29.

Figure 17-29. Logic Circuit of SCL Signal


CLC (manipulated by bit manipulation instruction)
SCL
Wait request signal
Serial clock (low while transfer is stopped)

Remarks 1. This figure indicates the relation of the signals and does not indicate the internal circuit.

2. CLC: Bit 3 of interrupt timing specify register (SINT)

CHAPTER 18 SERIAL INTERFACE CHANNEL 1

18.1 Serial Interface Channel 1 Functions

Serial interface channel 1 employs the following three modes.

- Operation stop mode
- 3-wire serial I/O mode
- 3-wire serial I/O mode with automatic transmit/receive function

(1) Operation stop mode

This mode is used when serial transfer is not carried out to reduce power consumption.

(2) 3-wire serial I/O mode (MSB-/LSB-first switchable)

This mode is used for 8-bit data transfer using three lines, each for serial clock ($\overline{\text{SCK1}}$), serial output (SO1), and serial input (SI1).

The 3-wire serial I/O mode enables simultaneous transmission/reception and so decreases the data transfer processing time.

Since the start bit of 8-bit data to undergo serial transfer is switchable between MSB and LSB, connection is enabled with either start bit device.

The 3-wire serial I/O mode is valid for connection of peripheral I/O units and display controllers which incorporate a conventional synchronous serial interface such as the 75X/XL, 78K, and 17K Series.

(3) 3-wire serial I/O mode with automatic transmit/receive function (MSB-/LSB-first switchable)

This mode is equivalent to the 3-wire serial I/O mode with the addition of an automatic transmit/receive function.

The automatic transmit/receive function is used to transmit/receive data with a maximum of 32 bytes. This function enables the hardware to transmit/receive data to/from the OSD (On Screen Display) device and a device with built-in display controller/driver independently of the CPU, thus the software load can be alleviated.

Caution When using the P23/STB/TxD1 and P24/BUSY/RxD1 pins in the asynchronous serial interface (UART) mode of serial interface channel 2, the busy control option and busy & strobe control option are invalid.

18.2 Serial Interface Channel 1 Configuration

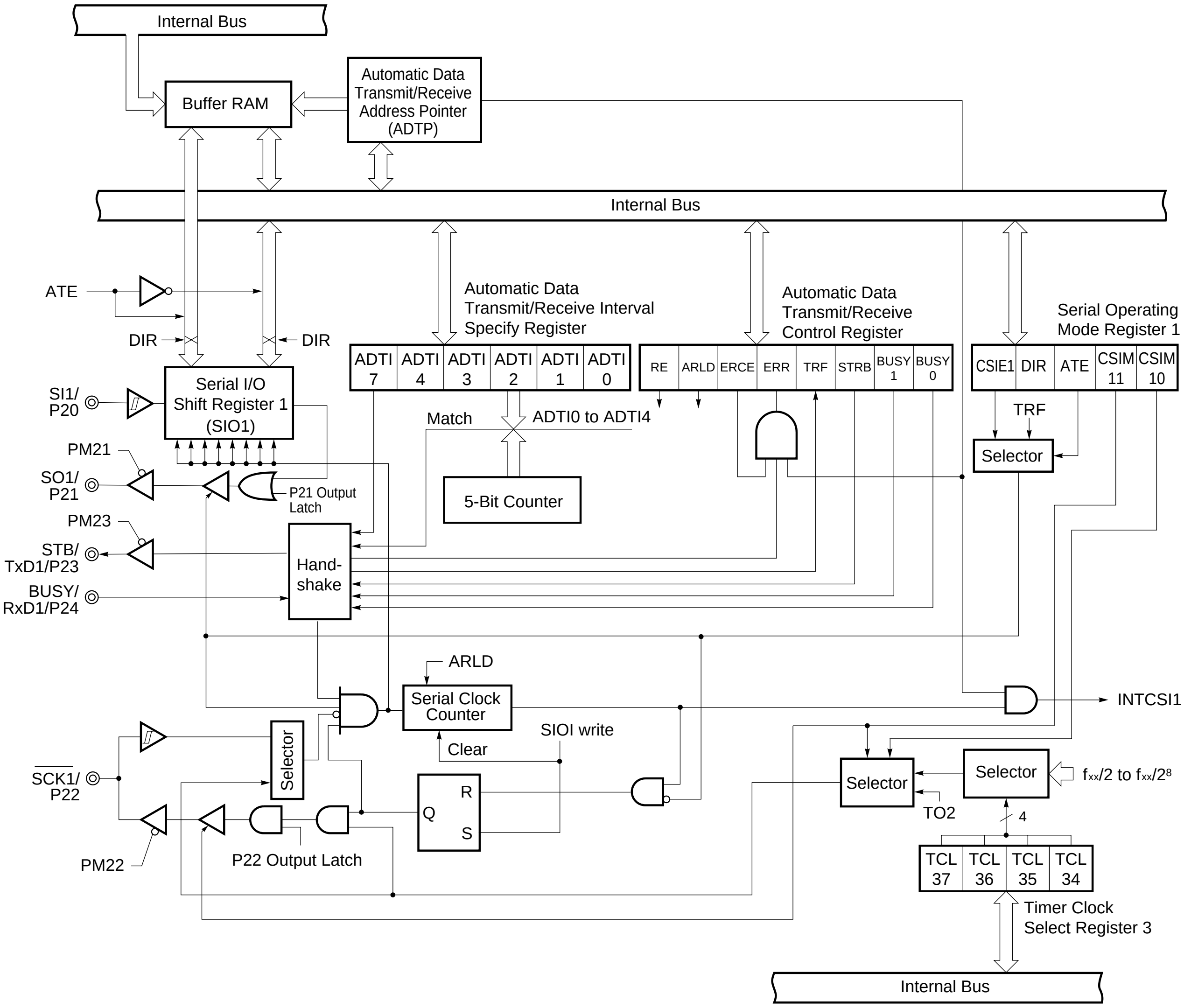
Serial interface channel 1 consists of the following hardware.

Table 18-1. Serial Interface Channel 1 Configuration

Item	Configuration
Register	Serial I/O shift register 1 (SIO1) Automatic data transmit/receive address pointer (ADTP)
Control register	Timer clock select register 3 (TCL3) Serial operating mode register 1 (CSIM1) Automatic data transmit/receive control register (ADTC) Automatic data transmit/receive interval specify register (ADTI) Port mode register 2 (PM2) [Note]

Note Refer to **Figures 6-5** and **6-7 P20, P21, and P23 to P26 Block Diagram** and **Figures 6-6** and **6-8 P22 and P27 Block Diagram**.

Figure 18-1. Serial Interface Channel 1 Block Diagram



(1) Serial I/O shift register 1 (SIO1)

This is an 8-bit register to carry out parallel/serial conversion and to carry out serial transmission/reception (shift operation) in synchronization with the serial clock.

The SIO1 is set with an 8-bit memory manipulation instruction.

When the value in bit 7 (CSIE1) of serial operating mode register 1 (CSIM1) is 1, writing data to SIO1 starts serial operation.

In transmission, data written to SIO1 is output to the serial output (SO1). In reception, data is read from the serial input (SI1) to the SIO1.

$\overline{\text{RESET}}$ input makes SIO1 undefined.

Caution Do not write data to SIO1 while the automatic transmit/receive function is activated.

(2) Automatic data transmit/receive address pointer (ADTP)

This register stores value of (the number of transmit data bytes – 1) while the automatic transmit/receive function is activated. As data is transferred/received, it is automatically decremented.

ADTP is set with an 8-bit memory manipulation instruction. The high-order 3 bits must be set to 0.

$\overline{\text{RESET}}$ input clears ADTP to 00H.

Caution Do not write data to ADTP while the automatic transmit/receive function is activated.

(3) Serial clock counter

This counter counts the serial clocks to be output and input during transmission/reception to check whether 8-bit data has been transmitted/received.

18.3 Serial Interface Channel 1 Control Registers

The following four types of registers are used to control serial interface channel 1.

- Timer clock select register 3 (TCL3)
- Serial operating mode register 1 (CSIM1)
- Automatic data transmit/receive control register (ADTC)
- Automatic data transmit/receive interval specify register (ADTI)

(1) Timer clock select register 3 (TCL3)

This register sets the serial clock of serial interface channel 1.
TCL3 is set with an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets TCL3 to 88H.

Remark Besides setting the serial clock of serial interface channel 1, TCL3 sets the serial clock of serial interface channel 0.

Figure 18-2. Timer Clock Select Register 3 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL37	TCL36	TCL35	TCL34	Serial Interface Channel 1 Serial Clock Selection		
					MCS = 1	MCS = 0
0	1	1	0	$f_{XX}/2$	Setting prohibited	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
Other than above				Setting prohibited		

Caution When rewriting other data to TCL3 , stop the serial transfer operation beforehand.

Remarks 1. f_{XX} : Main system clock frequency (f_X or $f_X/2$)

2. f_X : Main system clock oscillation frequency

3. MCS : Bit 0 of oscillation mode selection register (OSMS)

4. Figures in parentheses apply to operation with $f_X = 5.0$ MHz

(2) Serial operating mode register 1 (CSIM1)

This register sets serial interface channel 1 serial clock, operating mode, operation enable/stop and automatic transmit/receive operation enable/stop.

CSIM1 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM1 to 00H.

Figure 18-3. Serial Operation Mode Register 1 Format

Symbol	⑦	6	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM11	CSIM10	FF68H	00H	R/W

CSIM11	CSIM10	Serial Interface Channel 1 Clock Selection
0	×	External clock input to $\overline{\text{SCK1}}$ pin Note 1
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 4 to 7 of timer clock select register 3 (TCL3)

ATE	Serial Interface Channel 1 Operating Mode Selection
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

DIR	Start Bit	SI1 Pin Function	SO1 Pin Function
0	MSB	SI1/P20 (Input)	SO1 (CMOS output)
1	LSB		

CSIE1	CSIM11	PM20	P20	PM21	P21	PM22	P22	Shift Register 1 Operation	Serial Clock Counter Operation Control	SI1/P20 Pin Function	SO1/P21 Pin Function	$\overline{\text{SCK1}}$/P22 Pin Function
0	×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Operation stop	Clear	P20 (CMOS input/output)	P21 (CMOS input/output)	P22 (CMOS input/output)
1	0	Note 3 1	Note 3 ×	0	0	1	×	Operation enable	Count operation	SI1 Note 3 (input)	SO1 (CMOS output)	$\overline{\text{SCK1}}$ (Input)
	1					0	1					$\overline{\text{SCK1}}$ (CMOS output)

Notes 1. If the external clock input has been selected with CSIM11 set to 0, set bit 1 (BUSY1) and bit 2 (STRB) of the automatic data transmit/receive control register (ADTC) to 0, 0.

2. Can be used freely as port function.

3. Can be used as P20 (CMOS input/output) when only transmission is performed (clear bit 7 (RE) of ADTC to 0).

Remark		
	×	: don’t care
	PM××	: Port mode register
	P××	: Port output latch

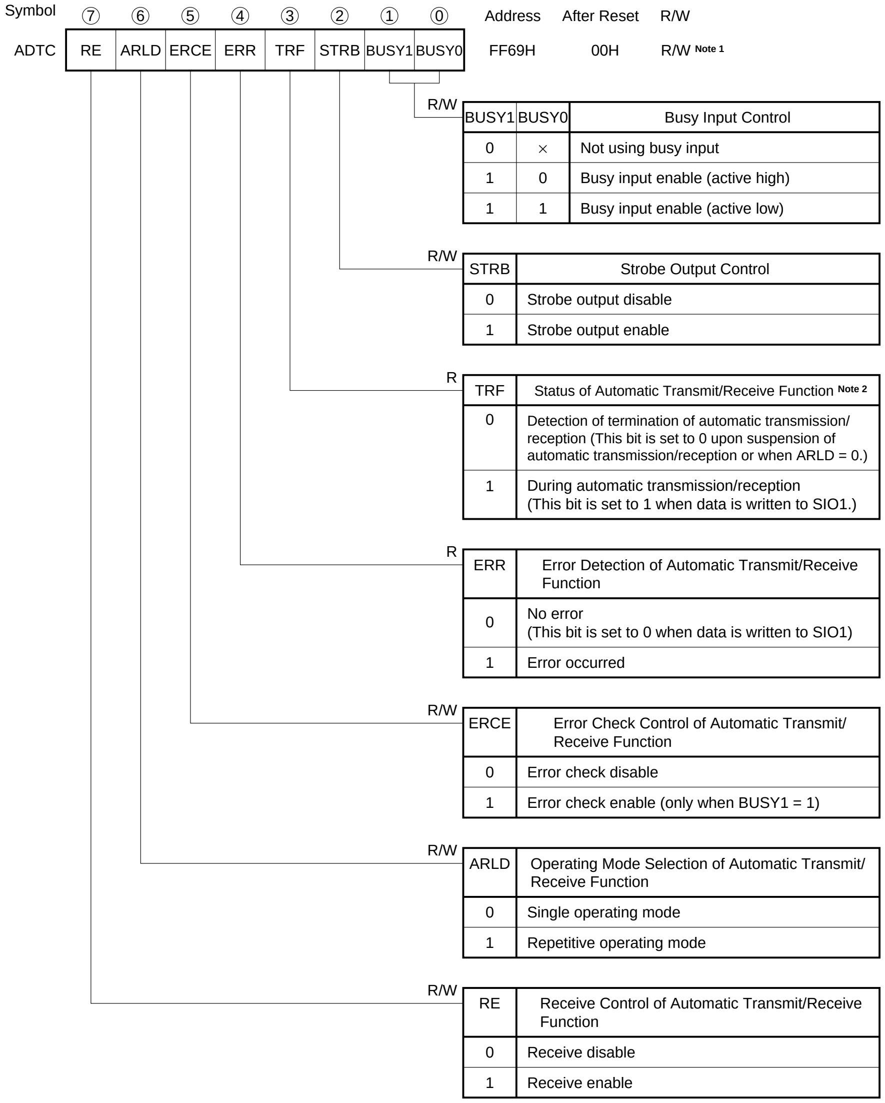
(3) Automatic data transmit/receive control register (ADTC)

This register sets automatic receive enable/disable, the operating mode, strobe output enable/disable, busy input enable/disable, error check enable/disable and displays automatic transmit/receive execution and error detection.

ADTC is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ADTC to 00H.

Figure 18-4. Automatic Data Transmit/Receive Control Register Format



Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
ADTC	RE	ARLD	ERCE	ERR	TRF	STRB	BUSY1	BUSY0	FF69H	00H	R/W Note 1

R/W

BUSY1	BUSY0	Busy Input Control
0	×	Not using busy input
1	0	Busy input enable (active high)
1	1	Busy input enable (active low)

R/W

STRB	Strobe Output Control
0	Strobe output disable
1	Strobe output enable

R

TRF	Status of Automatic Transmit/Receive Function Note 2
0	Detection of termination of automatic transmission/ reception (This bit is set to 0 upon suspension of automatic transmission/reception or when ARLD = 0.)
1	During automatic transmission/reception (This bit is set to 1 when data is written to SIO1.)

R

ERR	Error Detection of Automatic Transmit/Receive Function
0	No error (This bit is set to 0 when data is written to SIO1)
1	Error occurred

R/W

ERCE	Error Check Control of Automatic Transmit/ Receive Function
0	Error check disable
1	Error check enable (only when BUSY1 = 1)

R/W

ARLD	Operating Mode Selection of Automatic Transmit/ Receive Function
0	Single operating mode
1	Repetitive operating mode

R/W

RE	Receive Control of Automatic Transmit/Receive Function
0	Receive disable
1	Receive enable

Notes 1. Bits 3 and 4 (TRF and ERR) are read-only bits.

2. The termination of automatic transmission/reception should be judged by using TRF, not CSIIF1 (interrupt request flag).

Cautions 1. When an external clock input is selected by setting bit 1 (CSIM11) of the serial operating mode register 1 (CSIM1) to 0, set STRB and BUSY1 of ADTC to 0, 0.

2. When using the P23/STB/TxD1 and P24/BUSY/RxD1 pins in the asynchronous serial interface (UART) mode of serial interface channel 2, the busy control option and busy & strobe control option are invalid.

Remark ×: don't care

(4) Automatic data transmit/receive interval specify register (ADTI)

This register sets the automatic data transmit/receive function data transfer interval.

ADTI is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ADTI to 00H.

Figure 18-5. Automatic Data Transmit/Receive Interval Specify Register Format (1/4)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI7	Data Transfer Interval Control
0	No control of interval by ADTI[Note 1]
1	Control of interval by ADTI (ADTI0 to ADTI4)

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data Transfer Interval Specification (f_{XX} = 5.0 MHz Operation)	
					Minimum[Note 2]	Maximum[Note 2]
0	0	0	0	0	18.4 μs + 0.5/f_{SCK}	20.0 μs + 1.5/f_{SCK}
0	0	0	0	1	31.2 μs + 0.5/f_{SCK}	32.8 μs + 1.5/f_{SCK}
0	0	0	1	0	44.0 μs + 0.5/f_{SCK}	45.6 μs + 1.5/f_{SCK}
0	0	0	1	1	56.8 μs + 0.5/f_{SCK}	58.4 μs + 1.5/f_{SCK}
0	0	1	0	0	69.6 μs + 0.5/f_{SCK}	71.2 μs + 1.5/f_{SCK}
0	0	1	0	1	82.4 μs + 0.5/f_{SCK}	84.0 μs + 1.5/f_{SCK}
0	0	1	1	0	95.2 μs + 0.5/f_{SCK}	96.8 μs + 1.5/f_{SCK}
0	0	1	1	1	108.0 μs + 0.5/f_{SCK}	109.6 μs + 1.5/f_{SCK}
0	1	0	0	0	120.8 μs + 0.5/f_{SCK}	122.4 μs + 1.5/f_{SCK}
0	1	0	0	1	133.6 μs + 0.5/f_{SCK}	135.2 μs + 1.5/f_{SCK}
0	1	0	1	0	146.4 μs + 0.5/f_{SCK}	148.0 μs + 1.5/f_{SCK}
0	1	0	1	1	159.2 μs + 0.5/f_{SCK}	160.8 μs + 1.5/f_{SCK}
0	1	1	0	0	172.0 μs + 0.5/f_{SCK}	173.6 μs + 1.5/f_{SCK}
0	1	1	0	1	184.8 μs + 0.5/f_{SCK}	186.4 μs + 1.5/f_{SCK}
0	1	1	1	0	197.6 μs + 0.5/f_{SCK}	199.2 μs + 1.5/f_{SCK}
0	1	1	1	1	210.4 μs + 0.5/f_{SCK}	212.0 μs + 1.5/f_{SCK}

Notes 1. The interval is dependent only on CPU processing.

2. The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than $2/f_{SCK}$, the minimum interval time is $2/f_{SCK}$.

$$\text{Minimum} = (n + 1) \times \frac{2^6}{f_{XX}} + \frac{28}{f_{XX}} + \frac{0.5}{f_{SCK}}, \ \text{Maximum} = (n + 1) \times \frac{2^6}{f_{XX}} + \frac{36}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI during operation of automatic data transmit/receive function.

2. Bits 5 and 6 must be set to zero.

3. To control the data transfer interval by means of automatic transmission/reception with ADTI, busy control (refer to 18.4.3 (4) (a) Busy control option) is invalid.

Remarks 1. f_{XX} : Main system clock frequency (f_X or $f_X/2$)

2. f_X : Main system clock oscillation frequency

3. f_{SCK} : Serial clock frequency

Figure 18-5. Automatic Data Transmit/Receive Interval Specify Register Format (2/4)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data Transfer Interval Specification (f_{XX} = 5.0 MHz Operation)	
					Minimum Note	Maximum Note
1	0	0	0	0	$223.2\ \mu\text{s} + 0.5/f_{SCK}$	$224.8\ \mu\text{s} + 1.5/f_{SCK}$
1	0	0	0	1	$236.0\ \mu\text{s} + 0.5/f_{SCK}$	$237.6\ \mu\text{s} + 1.5/f_{SCK}$
1	0	0	1	0	$248.8\ \mu\text{s} + 0.5/f_{SCK}$	$250.4\ \mu\text{s} + 1.5/f_{SCK}$
1	0	0	1	1	$261.6\ \mu\text{s} + 0.5/f_{SCK}$	$263.2\ \mu\text{s} + 1.5/f_{SCK}$
1	0	1	0	0	$274.4\ \mu\text{s} + 0.5/f_{SCK}$	$276.0\ \mu\text{s} + 1.5/f_{SCK}$
1	0	1	0	1	$287.2\ \mu\text{s} + 0.5/f_{SCK}$	$288.8\ \mu\text{s} + 1.5/f_{SCK}$
1	0	1	1	0	$300.0\ \mu\text{s} + 0.5/f_{SCK}$	$301.6\ \mu\text{s} + 1.5/f_{SCK}$
1	0	1	1	1	$312.8\ \mu\text{s} + 0.5/f_{SCK}$	$314.4\ \mu\text{s} + 1.5/f_{SCK}$
1	1	0	0	0	$325.6\ \mu\text{s} + 0.5/f_{SCK}$	$327.2\ \mu\text{s} + 1.5/f_{SCK}$
1	1	0	0	1	$338.4\ \mu\text{s} + 0.5/f_{SCK}$	$340.0\ \mu\text{s} + 1.5/f_{SCK}$
1	1	0	1	0	$351.2\ \mu\text{s} + 0.5/f_{SCK}$	$352.8\ \mu\text{s} + 1.5/f_{SCK}$
1	1	0	1	1	$364.0\ \mu\text{s} + 0.5/f_{SCK}$	$365.6\ \mu\text{s} + 1.5/f_{SCK}$
1	1	1	0	0	$376.8\ \mu\text{s} + 0.5/f_{SCK}$	$378.4\ \mu\text{s} + 1.5/f_{SCK}$
1	1	1	0	1	$389.6\ \mu\text{s} + 0.5/f_{SCK}$	$391.2\ \mu\text{s} + 1.5/f_{SCK}$
1	1	1	1	0	$402.4\ \mu\text{s} + 0.5/f_{SCK}$	$404.0\ \mu\text{s} + 1.5/f_{SCK}$
1	1	1	1	1	$415.2\ \mu\text{s} + 0.5/f_{SCK}$	$416.8\ \mu\text{s} + 1.5/f_{SCK}$

Note The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than $2/f_{SCK}$, the minimum interval time is $2/f_{SCK}$.

$$\text{Minimum} = (n + 1) \times \frac{2^6}{f_{XX}} + \frac{28}{f_{XX}} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^6}{f_{XX}} + \frac{36}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI during operation of automatic data transmit/receive function.

2. Zero must be set in bits 5 and 6.

3. To control the data transfer interval by means of automatic transmission/reception with ADTI, busy control (refer to 18.4.3 (4) (a) Busy control option) is invalid.

Remarks 1. f_{XX} : Main system clock frequency (f_X or $f_X/2$)

2. f_X : Main system clock oscillation frequency

3. f_{SCK} : Serial clock frequency

Figure 18-5. Automatic Data Transmit/Receive Interval Specify Register Format (3/4)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI7	Data Transfer Interval Control
0	No control of interval by ADTI [Note 1]
1	Control of interval by ADTI (ADTI0 to ADTI4)

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data Transfer Interval Specification (f_{XX} = 2.5 MHz Operation)	
					Minimum [Note 2]	Maximum [Note 2]
0	0	0	0	0	36.8 μs + 0.5/f_{SCK}	40.0 μs + 1.5/f_{SCK}
0	0	0	0	1	62.4 μs + 0.5/f_{SCK}	65.6 μs + 1.5/f_{SCK}
0	0	0	1	0	88.0 μs + 0.5/f_{SCK}	91.2 μs + 1.5/f_{SCK}
0	0	0	1	1	113.6 μs + 0.5/f_{SCK}	116.8 μs + 1.5/f_{SCK}
0	0	1	0	0	139.2 μs + 0.5/f_{SCK}	142.4 μs + 1.5/f_{SCK}
0	0	1	0	1	164.8 μs + 0.5/f_{SCK}	168.0 μs + 1.5/f_{SCK}
0	0	1	1	0	190.4 μs + 0.5/f_{SCK}	193.6 μs + 1.5/f_{SCK}
0	0	1	1	1	216.0 μs + 0.5/f_{SCK}	219.2 μs + 1.5/f_{SCK}
0	1	0	0	0	241.6 μs + 0.5/f_{SCK}	244.8 μs + 1.5/f_{SCK}
0	1	0	0	1	267.2 μs + 0.5/f_{SCK}	270.4 μs + 1.5/f_{SCK}
0	1	0	1	0	292.8 μs + 0.5/f_{SCK}	296.0 μs + 1.5/f_{SCK}
0	1	0	1	1	318.4 μs + 0.5/f_{SCK}	321.6 μs + 1.5/f_{SCK}
0	1	1	0	0	344.0 μs + 0.5/f_{SCK}	347.2 μs + 1.5/f_{SCK}
0	1	1	0	1	369.6 μs + 0.5/f_{SCK}	372.8 μs + 1.5/f_{SCK}
0	1	1	1	0	395.2 μs + 0.5/f_{SCK}	398.4 μs + 1.5/f_{SCK}
0	1	1	1	1	420.8 μs + 0.5/f_{SCK}	424.0 μs + 1.5/f_{SCK}

Notes 1. The interval is dependent only on CPU processing.

2. The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than 2/f_{SCK}, the minimum interval time is 2/f_{SCK}.

$$\text{Minimum} = (n + 1) \times \frac{2^6}{f_{XX}} + \frac{28}{f_{XX}} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^6}{f_{XX}} + \frac{36}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI during operation of automatic data transmit/receive function.

2. Bits 5 and 6 must be set to zero.

3. To control the data transfer interval by means of automatic transmission/reception with ADTI, busy control (refer to 18.4.3 (4) (a) Busy control option) is invalid.

Remarks 1. f_{XX} : Main system clock frequency (f_X or f_X/2)

2. f_X : Main system clock oscillation frequency

3. f_{SCK} : Serial clock frequency

Figure 18-5. Automatic Data Transmit/Receive Interval Specify Register Format (4/4)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data Transfer Interval Specification (f_{XX} = 2.5 MHz Operation)	
					Minimum Note	Maximum Note
1	0	0	0	0	446.4 μs + 0.5/f_{SCK}	449.6 μs + 1.5/f_{SCK}
1	0	0	0	1	472.0 μs + 0.5/f_{SCK}	475.2 μs + 1.5/f_{SCK}
1	0	0	1	0	497.6 μs + 0.5/f_{SCK}	500.8 μs + 1.5/f_{SCK}
1	0	0	1	1	523.2 μs + 0.5/f_{SCK}	526.4 μs + 1.5/f_{SCK}
1	0	1	0	0	548.8 μs + 0.5/f_{SCK}	552.0 μs + 1.5/f_{SCK}
1	0	1	0	1	574.4 μs + 0.5/f_{SCK}	577.6 μs + 1.5/f_{SCK}
1	0	1	1	0	600.0 μs + 0.5/f_{SCK}	603.2 μs + 1.5/f_{SCK}
1	0	1	1	1	625.6 μs + 0.5/f_{SCK}	628.8 μs + 1.5/f_{SCK}
1	1	0	0	0	651.2 μs + 0.5/f_{SCK}	654.4 μs + 1.5/f_{SCK}
1	1	0	0	1	676.8 μs + 0.5/f_{SCK}	680.0 μs + 1.5/f_{SCK}
1	1	0	1	0	702.4 μs + 0.5/f_{SCK}	705.6 μs + 1.5/f_{SCK}
1	1	0	1	1	728.0 μs + 0.5/f_{SCK}	731.2 μs + 1.5/f_{SCK}
1	1	1	0	0	753.6 μs + 0.5/f_{SCK}	756.8 μs + 1.5/f_{SCK}
1	1	1	0	1	779.2 μs + 0.5/f_{SCK}	782.4 μs + 1.5/f_{SCK}
1	1	1	1	0	804.8 μs + 0.5/f_{SCK}	808.0 μs + 1.5/f_{SCK}
1	1	1	1	1	830.4 μs + 0.5/f_{SCK}	833.6 μs + 1.5/f_{SCK}

Note The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than 2/f_{SCK}, the minimum interval time is 2/f_{SCK}.

$$\text{Minimum} = (n + 1) \times \frac{2^6}{f_{XX}} + \frac{28}{f_{XX}} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^6}{f_{XX}} + \frac{36}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI during operation of automatic data transmit/receive function.

2. Bits 5 and 6 must be set to zero.

3. To control the data transfer interval by means of automatic transmission/reception with ADTI, busy control (refer to 18.4.3 (4) (a) Busy control option) is invalid.

Remarks 1. f_{XX} : Main system clock frequency (f_X or f_X/2)

2. f_X : Main system clock oscillation frequency

3. f_{SCK} : Serial clock frequency

18.4 Serial Interface Channel 1 Operations

The following three operating modes are available to the serial interface channel 1.

- Operation stop mode
- 3-wire serial I/O mode
- 3-wire serial I/O mode with automatic transmit/receive function

18.4.1 Operation stop mode

Serial transfer is not carried out in the operation stop mode. Thus, power consumption can be reduced. The serial I/O shift register 1 (SIO1) does not carry out shift operation either, and thus it can be used as an ordinary 8-bit register.

In the operation stop mode, the P20/SI1, P21/SO1, P22/$\overline{\text{SCK1}}$, P23/STB/TxD1, and P24/BUSY/RxD1 pins can be used as ordinary input/output ports.

(1) Register setting

The operation stop mode is set with the serial operating mode register 1 (CSIM1).

CSIM1 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM1 to 00H.

Symbol	⑦	6	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM11	CSIM10	FF68H	00H	R/W

CSIE1	CSIM11	PM20	P20	PM21	P21	PM22	P22	Shift Register 1 Operation	Serial Clock Counter Operation Control	SI1/P20 Pin Function	SO1/P21 Pin Function	$\overline{\text{SCK1}}$/P22 Pin Function
0	×	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	Operation stop	Clear	P20 (CMOS input/output)	P21 (CMOS input/output)	P22 (CMOS input/output)
1	0	1[Note 2]	×[Note 2]	0	0	1	×	Operation enable	Count operation	SI1[Note 2] (Input)	SO1 (CMOS output)	$\overline{\text{SCK1}}$ (Input)
	1					0	1					$\overline{\text{SCK1}}$ (CMOS output)

Notes 1. Can be used freely as port function.

2. Can be used as P20 (CMOS input/output) when only transmission is performed (clear bit 7 (RE) of the automatic data transmit/receive control register (ADTC) to 0).

Remark × : don't care

PM×× : Port mode register

P×× : Port output latch

18.4.2 3-wire serial I/O mode operation

The 3-wire serial I/O mode is valid for connection of peripheral I/O units and display controllers which incorporate a conventional synchronous serial interface such as the 75X/XL, 78K and 17K Series.

Communication is carried out with three lines of serial clock ($\overline{\text{SCK1}}$), serial output (SO1) and serial input (SI1).

(1) Register setting

The 3-wire serial I/O mode is set with the serial operating mode register 1 (CSIM1).

CSIM1 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM1 to 00H.

Symbol	⑦	6	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM11	CSIM10	FF68H	00H	R/W

CSIM11	CSIM10	Serial Interface Channel 1 Clock Selection
0	×	External clock input to $\overline{\text{SCK1}}$ pin [Note]
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 4 to 7 of timer clock select register 3 (TCL3)

ATE	Serial Interface Channel 1 Operating Mode Selection
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

DIR	Start Bit	SO1 Pin Function	SO1 Pin Function
0	MSB	SI1/P20 (Input)	SO1 (CMOS output)
1	LSB		

Note If the external clock input has been selected by setting the CSIM11 to 0, set bit 1 (BUSY1) and bit 2 (STRB) of the automatic data transmit/receive control register (ADTC) to 0, 0.

Remark ×: don't care

CSIE1	CSIM11	PM20	P20	PM21	P21	PM22	P22	Shift Register 1 Operation	Serial Clock Counter Operation Control	SI1/P20 Pin Function	SO1/P21 Pin Function	$\overline{\text{SCK1}}$/P22 Pin Function
0	×	[Note 1] ×	[Note 1] ×	[Note 1] ×	[Note 1] ×	[Note 1] ×	[Note 1] ×	Operation stop	Clear	P20 (CMOS input/output)	P21 (CMOS input/output)	P22 (CMOS input/output)
1	0	[Note 2] 1	[Note 2] ×	0	0	1	×	Operation enable	Count operation	SI1[Note 2] (Input)	SO1 (CMOS output)	$\overline{\text{SCK1}}$ (Input)
	1					0	1					$\overline{\text{SCK1}}$ (CMOS output)

Notes 1. Can be used freely as port function.

2. Can be used as P20 (CMOS input/output) when only transmission is performed (clear bit 7 (RE) of ADTC to 0).

Remark × : don't care

PM×× : Port mode register

P×× : Port output latch

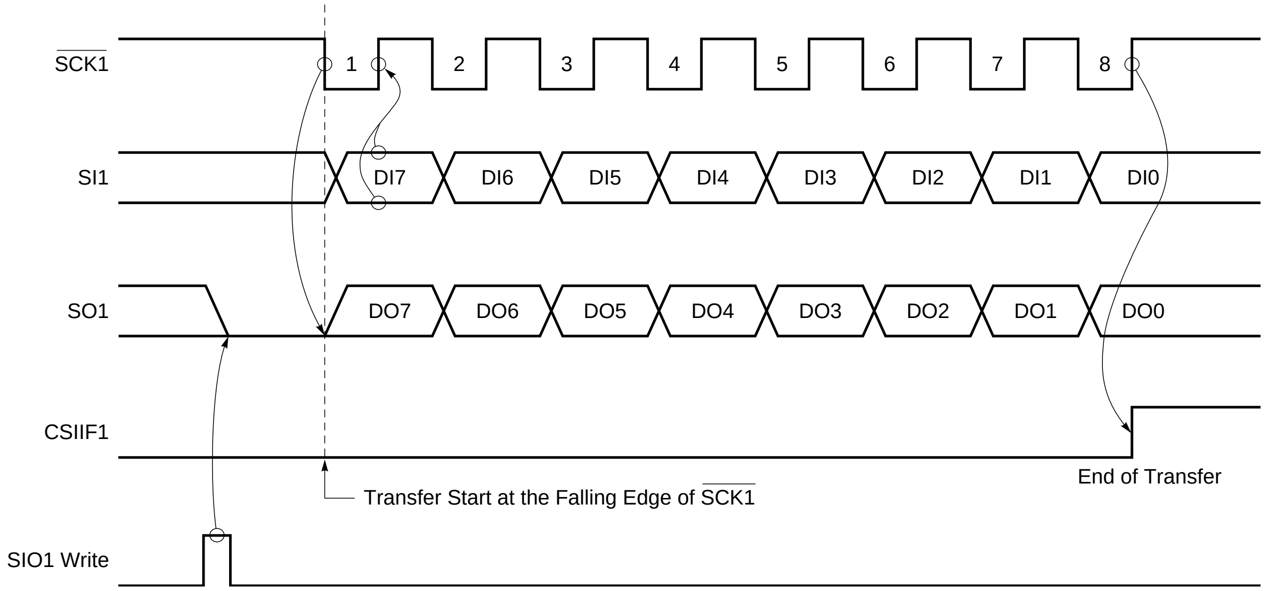
(2) Communication operation

The 3-wire serial I/O mode is used for data transmission/reception in 8-bit units. Bit-wise data transmission/reception is carried out in synchronization with the serial clock.

Shift operation of the serial I/O shift register 1 (SIO1) is carried out at the falling edge of the serial clock $\overline{\text{SCK1}}$. The transmit data is held in the SO1 latch and is output from the SO1 pin. The receive data input to the SI1 pin is latched into SIO1 at the rising edge of $\overline{\text{SCK1}}$.

Upon termination of 8-bit transfer, the SIO1 operation stops automatically and the interrupt request flag (CSIIF1) is set.

Figure 18-6. 3-wire Serial I/O Mode Timings



Caution SO1 pin becomes low level by SIO1 write.

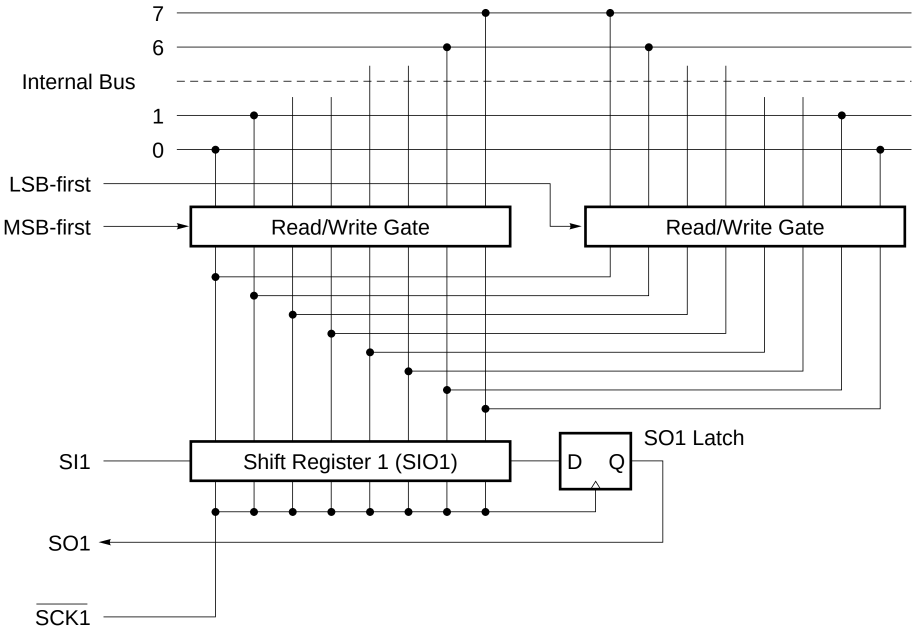
★ **(3) MSB/LSB switching as the start bit**

The 3-wire serial I/O mode enables to select transfer to start from MSB or LSB.

Figure 18-7 shows the configuration of the serial I/O shift register 1 (SIO1) and internal bus. As shown in the figure, MSB/LSB can be read/written in reverse form.

MSB/LSB switching as the start bit can be specified with bit 6 (DIR) of the serial operating mode register 1 (CSIM1).

Figure 18-7. Circuit of Switching in Transfer Bit Order



Start bit switching is realized by switching the bit order for data write to SIO1. The SIO1 shift order remains unchanged.

Thus, switching between MSB-first and LSB-first must be performed before writing data to the shift register.

(4) Transfer start

Serial transfer is started by setting transfer data to the serial I/O shift register 1 (SIO1) when the following two conditions are satisfied.

- Serial interface channel 1 operation control bit (CSIE1) = 1
- Internal serial clock is stopped or $\overline{\text{SCK1}}$ is a high level after 8-bit serial transfer.

Caution If CSIE1 is set to "1" after data write to SIO1, transfer does not start.

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (CSIIF1) is set.

18.4.3 3-wire serial I/O mode operation with automatic transmit/receive function

This 3-wire serial I/O mode is used for transmission/reception of a maximum of 32-byte data without the use of software. Once transfer is started, the data prestored in the RAM can be transmitted by the set number of bytes, and data can be received and stored in the RAM by the set number of bytes.

Handshake signals (STB and BUSY) are supported by hardware to transmit/receive data continuously. OSD (On Screen Display) LSI and peripheral LSI including LCD controller/driver can be connected without difficulty.

(1) Register setting

The 3-wire serial I/O mode with automatic transmit/receive function is set with the serial operating mode register 1 (CSIM1), the automatic data transmit/receive control register (ADTC) and the automatic data transmit/receive interval specify register (ADTI).

(a) Serial operating mode register 1 (CSIM1)

CSIM1 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM1 to 00H.

Symbol	⑦	6	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM11	CSIM10	FF68H	00H	R/W

CSIM11	CSIM10	Serial Interface Channel 1 Clock Selection
0	×	External clock input to $\overline{\text{SCK1}}$ pin[Note 1]
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 4 to 7 of timer clock select register 3 (TCL3)

ATE	Serial Interface Channel 1 Operating Mode Selection
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

DIR	Start Bit	SI1 Pin Function	SO1 Pin Function
0	MSB	SI1/P20 (Input)	SO1 (CMOS output)
1	LSB		

CSIE1	CSIM11	PM20	P20	PM21	P21	PM22	P22	Shift Register 1 Operation	Serial Clock Counter Operation Control	SI1/P20 Pin Function	SO1/P21 Pin Function	$\overline{\text{SCK1}}$/P22 Pin Function
0	×	[Note 2] ×	[Note 2] ×	[Note 2] ×	[Note 2] ×	[Note 2] ×	[Note 2] ×	Operation stop	Clear	P20 (CMOS input/output)	P21 (CMOS input/output)	P22 (CMOS input/output)
1	0	[Note 3] 1	[Note 3] ×	0	0	1	×	Operation enable	Count operation	SI1[Note 3] (Input)	SO1 (CMOS output)	$\overline{\text{SCK1}}$ (Input)
	1					0	1					$\overline{\text{SCK1}}$ (CMOS output)

Notes 1. If the external clock input has been selected by setting the CSIM11 to 0, set bit 1 (BUSY 1) and bit 2 (STRB) of the automatic data transmit/receive control register (ADTC) to 0, 0.

2. Can be used freely as port function.

3. Can be used as P20 (CMOS input/output) when only transmission is performed (clear bit 7 (RE) of ADTC to 0).

Remark		
	×	: don’t care
	PM××	: Port mode register
	P××	: Port output latch

(b) Automatic data transmit/receive control register (ADTC)

ADTC is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ADTC to 00H.

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
ADTC	RE	ARLD	ERCE	ERR	TRF	STRB	BUSY1	BUSY0	FF69H	00H	R/W Note 1

R/W

BUSY1	BUSY0	Busy Input Control
0	×	Not using busy input
1	0	Busy input enable (active high)
1	1	Busy input enable (active low)

R/W

STRB	Strobe Output Control
0	Strobe output disable
1	Strobe output enable

R

TRF	Status of Automatic Transmit/Receive Function Note 2
0	Detection of termination of automatic transmission/ reception (This bit is set to 0 upon suspension of automatic transmission/reception or when ARLD = 0.)
1	During automatic transmission/reception (This bit is set to 1 when data is written to SIO1.)

R

ERR	Error Detection of Automatic Transmit/Receive Function
0	No error (This bit is set to 0 when data is written to SIO1)
1	Error occurred

R/W

ERCE	Error Check Control of Automatic Transmit/ Receive Function
0	Error check disable
1	Error check enable (only when BUSY1 = 1)

R/W

ARLD	Operating Mode Selection of Automatic Transmit/ Receive Function
0	Single operating mode
1	Repetitive operating mode

R/W

RE	Receive Control of Automatic Transmit/Receive Function
0	Receive disable
1	Receive enable

Notes 1. Bits 3 and 4 (TRF and ERR) are read-only bits.

2. The termination of automatic transmission/reception should be judged by using TRF, not CSIIF1 (interrupt request flag).

Caution When an external clock input is selected by setting bit 1 (CSIM11) of the serial operating mode register 1 (CSIM1) to 0, set STRB and BUSY1 of ADTC to 0, 0 (handshake control cannot be executed when the external clock is input).

Remark ×: don't care

(c) Automatic data transmit/receive interval specify register (ADTI)

This register sets the automatic data transmit/receive function data transfer interval.

ADTI is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ADTI to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI7	Data Transfer Interval Control
0	No control of interval by ADTI Note 1
1	Control of interval by ADTI (ADTI0 to ADTI4)

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data Transfer Interval Specification (f_{XX} = 5.0 MHz Operation)	
					Minimum Note 2	Maximum Note 2
0	0	0	0	0	18.4 μs + 0.5/f_{SCK}	20.0 μs + 1.5/f_{SCK}
0	0	0	0	1	31.2 μs + 0.5/f_{SCK}	32.8 μs + 1.5/f_{SCK}
0	0	0	1	0	44.0 μs + 0.5/f_{SCK}	45.6 μs + 1.5/f_{SCK}
0	0	0	1	1	56.8 μs + 0.5/f_{SCK}	58.4 μs + 1.5/f_{SCK}
0	0	1	0	0	69.6 μs + 0.5/f_{SCK}	71.2 μs + 1.5/f_{SCK}
0	0	1	0	1	82.4 μs + 0.5/f_{SCK}	84.0 μs + 1.5/f_{SCK}
0	0	1	1	0	95.2 μs + 0.5/f_{SCK}	96.8 μs + 1.5/f_{SCK}
0	0	1	1	1	108.0 μs + 0.5/f_{SCK}	109.6 μs + 1.5/f_{SCK}
0	1	0	0	0	120.8 μs + 0.5/f_{SCK}	122.4 μs + 1.5/f_{SCK}
0	1	0	0	1	133.6 μs + 0.5/f_{SCK}	135.2 μs + 1.5/f_{SCK}
0	1	0	1	0	146.4 μs + 0.5/f_{SCK}	148.0 μs + 1.5/f_{SCK}
0	1	0	1	1	159.2 μs + 0.5/f_{SCK}	160.8 μs + 1.5/f_{SCK}
0	1	1	0	0	172.0 μs + 0.5/f_{SCK}	173.6 μs + 1.5/f_{SCK}
0	1	1	0	1	184.8 μs + 0.5/f_{SCK}	186.4 μs + 1.5/f_{SCK}
0	1	1	1	0	197.6 μs + 0.5/f_{SCK}	199.2 μs + 1.5/f_{SCK}
0	1	1	1	1	210.4 μs + 0.5/f_{SCK}	212.0 μs + 1.5/f_{SCK}

Notes 1. The interval is dependent only on CPU processing.

2. The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than 2/f_{SCK}, the minimum interval time is 2/f_{SCK}.

$$\text{Minimum} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{28}{f_{XX}} + \frac{0.5}{f_{SCK}},\quad \text{Maximum} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{36}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI during operation of automatic data transmit/receive function.

2. Zero must be set in bits 5 and 6.

3. To control the data transfer interval by means of automatic transmission/reception with ADTI, busy control (refer to 18.4.3 (4) (a) Busy control option) is invalid.

Remarks 1. f_{XX} : Main system clock frequency (f_X or f_X/2)

2. f_X : Main system clock oscillation frequency

3. f_{SCK} : Serial clock frequency

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data Transfer Interval Specification (f_{XX} = 5.0 MHz Operation)	
					Minimum Note	Maximum Note
1	0	0	0	0	223.2 μs + 0.5/f_{SCK}	224.8 μs + 1.5/f_{SCK}
1	0	0	0	1	236.0 μs + 0.5/f_{SCK}	237.6 μs + 1.5/f_{SCK}
1	0	0	1	0	248.8 μs + 0.5/f_{SCK}	250.4 μs + 1.5/f_{SCK}
1	0	0	1	1	261.6 μs + 0.5/f_{SCK}	263.2 μs + 1.5/f_{SCK}
1	0	1	0	0	274.4 μs + 0.5/f_{SCK}	276.0 μs + 1.5/f_{SCK}
1	0	1	0	1	287.2 μs + 0.5/f_{SCK}	288.8 μs + 1.5/f_{SCK}
1	0	1	1	0	300.0 μs + 0.5/f_{SCK}	301.6 μs + 1.5/f_{SCK}
1	0	1	1	1	312.8 μs + 0.5/f_{SCK}	314.4 μs + 1.5/f_{SCK}
1	1	0	0	0	325.6 μs + 0.5/f_{SCK}	327.2 μs + 1.5/f_{SCK}
1	1	0	0	1	338.4 μs + 0.5/f_{SCK}	340.0 μs + 1.5/f_{SCK}
1	1	0	1	0	351.2 μs + 0.5/f_{SCK}	352.8 μs + 1.5/f_{SCK}
1	1	0	1	1	364.0 μs + 0.5/f_{SCK}	365.6 μs + 1.5/f_{SCK}
1	1	1	0	0	376.8 μs + 0.5/f_{SCK}	378.4 μs + 1.5/f_{SCK}
1	1	1	0	1	389.6 μs + 0.5/f_{SCK}	391.2 μs + 1.5/f_{SCK}
1	1	1	1	0	402.4 μs + 0.5/f_{SCK}	404.0 μs + 1.5/f_{SCK}
1	1	1	1	1	415.2 μs + 0.5/f_{SCK}	416.8 μs + 1.5/f_{SCK}

Note The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than 2/f_{SCK}, the minimum interval time is 2/f_{SCK}.

$$\text{Minimum} = (n + 1) \times \frac{2^6}{f_{XX}} + \frac{28}{f_{XX}} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^6}{f_{XX}} + \frac{36}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI during operation of automatic data transmit/receive function.

2. Bits 5 and 6 must be set to zero.

3. To control the data transfer interval by means of automatic transmission/reception with ADTI, busy control (refer to 18.4.3 (4) (a) Busy control option) is invalid.

Remarks 1. f_{XX} : Main system clock frequency (f_X or f_X/2)

2. f_X : Main system clock oscillation frequency

3. f_{SCK} : Serial clock frequency

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI7	Data Transfer Interval Control
0	No control of interval by ADT I[Note 1]
1	Control of interval by ADTI (ADTI0 to ADTI4)

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data Transfer Interval Specification (f_{XX} = 2.5 MHz Operation)	
					Minimum [Note 2]	Maximum [Note 2]
0	0	0	0	0	36.8 μs + 0.5/f_{SCK}	40.0 μs + 1.5/f_{SCK}
0	0	0	0	1	62.4 μs + 0.5/f_{SCK}	65.6 μs + 1.5/f_{SCK}
0	0	0	1	0	88.0 μs + 0.5/f_{SCK}	91.2 μs + 1.5/f_{SCK}
0	0	0	1	1	113.6 μs + 0.5/f_{SCK}	116.8 μs + 1.5/f_{SCK}
0	0	1	0	0	139.2 μs + 0.5/f_{SCK}	142.4 μs + 1.5/f_{SCK}
0	0	1	0	1	164.8 μs + 0.5/f_{SCK}	168.0 μs + 1.5/f_{SCK}
0	0	1	1	0	190.4 μs + 0.5/f_{SCK}	193.6 μs + 1.5/f_{SCK}
0	0	1	1	1	216.0 μs + 0.5/f_{SCK}	219.2 μs + 1.5/f_{SCK}
0	1	0	0	0	241.6 μs + 0.5/f_{SCK}	244.8 μs + 1.5/f_{SCK}
0	1	0	0	1	267.2 μs + 0.5/f_{SCK}	270.4 μs + 1.5/f_{SCK}
0	1	0	1	0	292.8 μs + 0.5/f_{SCK}	296.0 μs + 1.5/f_{SCK}
0	1	0	1	1	318.4 μs + 0.5/f_{SCK}	321.6 μs + 1.5/f_{SCK}
0	1	1	0	0	344.0 μs + 0.5/f_{SCK}	347.2 μs + 1.5/f_{SCK}
0	1	1	0	1	369.6 μs + 0.5/f_{SCK}	372.8 μs + 1.5/f_{SCK}
0	1	1	1	0	395.2 μs + 0.5/f_{SCK}	398.4 μs + 1.5/f_{SCK}
0	1	1	1	1	420.8 μs + 0.5/f_{SCK}	424.0 μs + 1.5/f_{SCK}

Notes 1. The interval is dependent only on CPU processing.

2. The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than 2/f_{SCK}, the minimum interval time is 2/f_{SCK}.

$$\text{Minimum} = (n + 1) \times \frac{2^6}{f_{XX}} + \frac{28}{f_{XX}} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^6}{f_{XX}} + \frac{36}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI during operation of automatic data transmit/receive function.

2. Bits 5 and 6 must be set to zero.

3. To control the data transfer interval by means of automatic transmission/reception with ADTI, busy control (refer to 18.4.3 (4) (a) Busy control option) is invalid.

Remarks 1. f_{XX} : Main system clock frequency (f_X or f_X/2)

2. f_X : Main system clock oscillation frequency

3. f_{SCK} : Serial clock frequency

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data Transfer Interval Specification (f_{XX} = 2.5 MHz Operation)	
					Minimum Note	Maximum Note
1	0	0	0	0	446.4 μs + 0.5/f_{SCK}	449.6 μs + 1.5/f_{SCK}
1	0	0	0	1	472.0 μs + 0.5/f_{SCK}	475.2 μs + 1.5/f_{SCK}
1	0	0	1	0	497.6 μs + 0.5/f_{SCK}	500.8 μs + 1.5/f_{SCK}
1	0	0	1	1	523.2 μs + 0.5/f_{SCK}	526.4 μs + 1.5/f_{SCK}
1	0	1	0	0	548.8 μs + 0.5/f_{SCK}	552.0 μs + 1.5/f_{SCK}
1	0	1	0	1	574.4 μs + 0.5/f_{SCK}	577.6 μs + 1.5/f_{SCK}
1	0	1	1	0	600.0 μs + 0.5/f_{SCK}	603.2 μs + 1.5/f_{SCK}
1	0	1	1	1	625.6 μs + 0.5/f_{SCK}	628.8 μs + 1.5/f_{SCK}
1	1	0	0	0	651.2 μs + 0.5/f_{SCK}	654.4 μs + 1.5/f_{SCK}
1	1	0	0	1	676.8 μs + 0.5/f_{SCK}	680.0 μs + 1.5/f_{SCK}
1	1	0	1	0	702.4 μs + 0.5/f_{SCK}	705.6 μs + 1.5/f_{SCK}
1	1	0	1	1	728.0 μs + 0.5/f_{SCK}	731.2 μs + 1.5/f_{SCK}
1	1	1	0	0	753.6 μs + 0.5/f_{SCK}	756.8 μs + 1.5/f_{SCK}
1	1	1	0	1	779.2 μs + 0.5/f_{SCK}	782.4 μs + 1.5/f_{SCK}
1	1	1	1	0	804.8 μs + 0.5/f_{SCK}	808.0 μs + 1.5/f_{SCK}
1	1	1	1	1	830.4 μs + 0.5/f_{SCK}	833.6 μs + 1.5/f_{SCK}

Note The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than 2/f_{SCK}, the minimum interval time is 2/f_{SCK}.

$$\text{Minimum} = (n + 1) \times \frac{2^6}{f_{XX}} + \frac{28}{f_{XX}} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^6}{f_{XX}} + \frac{36}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI during operation of automatic data transmit/receive function.

2. Bits 5 and 6 must be set to zero.

3. To control the data transfer interval by means of automatic transmission/reception with ADTI, busy control (refer to 18.4.3 (4) (a) Busy control option) is invalid.

Remarks 1. f_{XX} : Main system clock frequency (f_X or f_X/2)

2. f_X : Main system clock oscillation frequency

3. f_{SCK} : Serial clock frequency

(2) Automatic transmit/receive data setting

(a) Transmit data setting

<1> Write transmit data from the least significant address FAC0H of buffer RAM (up to FADFH at maximum). The transmit data should be in the order from high-order address to low-order address.

<2> Set to the automatic data transmit/receive address pointer (ADTP) the value obtained by subtracting 1 from the number of transmit data bytes.

(b) Automatic transmit/receive mode setting

<1> Set CSIE1 and ATE of the serial operating mode register 1 (CSIM1) to 1.

<2> Set RE of the automatic data transmit/receive control register (ADTC) to 1.

<3> Set a data transmit/receive interval in the automatic data transmit/receive interval specify register (ADTI).

<4> Write any value to the serial I/O shift register 1 (SIO1) (transfer start trigger).

Caution Writing any value to SIO1 orders the start of automatic transmit/receive operation and the written value has no meaning.

The following operations are automatically carried out when (a) and (b) are carried out.

- After the buffer RAM data specified with ADTP is transferred to SIO1, transmission is carried out (start of automatic transmission/reception).
- The received data is written to the buffer RAM address specified with ADTP.
- ADTP is decremented and the next data transmission/reception is carried out. Data transmission/reception continues until the ADTP decremental output becomes 00H and address FAC0H data is output (end of automatic transmission/reception).
- When automatic transmission/reception is terminated, TRF is cleared to 0.

(3) Communication operation

(a) Basic transmission/reception mode

This transmission/reception mode is the same as the 3-wire serial I/O mode in which specified number of data are transmitted/received in 8-bit units.

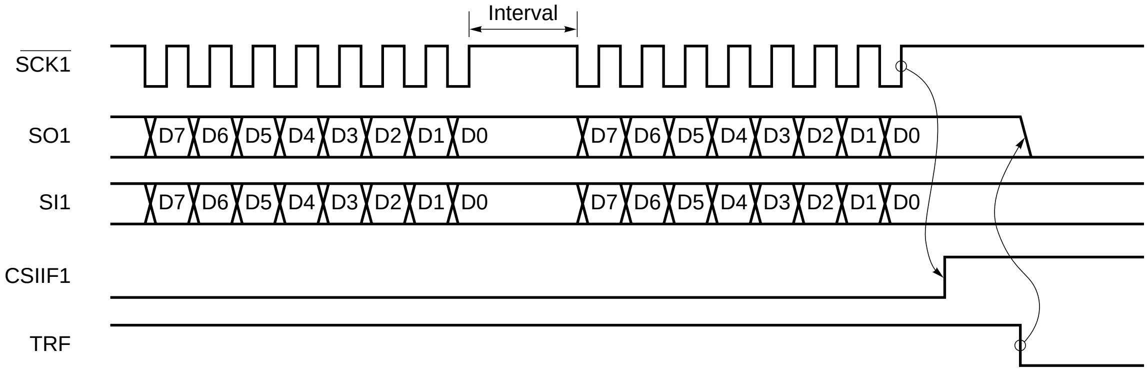
Serial transfer is started when any data is written to the serial I/O shift register 1 (SIO1) while bit 7 (CSIE1) of the serial operating mode register 1 (CSIM1) is set to 1.

When the final byte has been sent, an interrupt request flag (CSIIF1) is set. However, judge the termination of auto transmit and receive, not by CSIIF1 but by bit 3 (TRF) of the automatic data transmit/receive control register (ADTC).

If busy control and strobe control are not executed, the P23/STB/TxD1 and P24/BUSY/RxD1 pins can be used as normal input/output ports.

Figure 18-8 shows the basic transmission/reception mode operation timings, and Figure 18-9 shows the operation flowchart. Figure 18-10 shows the operation of the internal buffer RAM when 6 bytes of data are transmitted or received.

Figure 18-8. Basic Transmission/Reception Mode Operation Timings

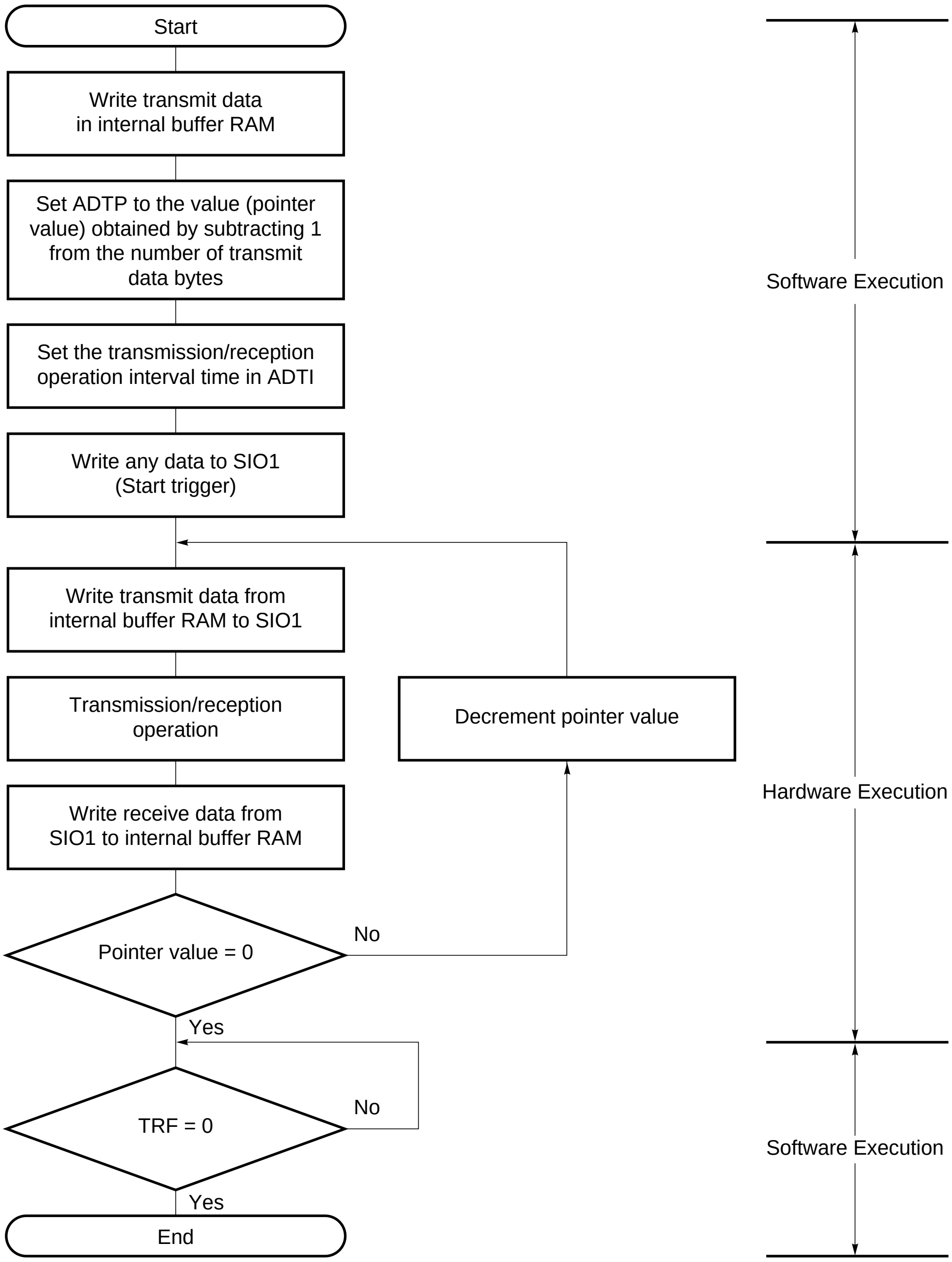


Cautions 1. Because, in the basic transmission/reception mode, the automatic transmit/receive function writes/reads data to/from the internal buffer RAM after 1-byte transmission/reception, an interval is inserted till the next transmission/reception. As the buffer RAM write/read is performed at the same time as CPU processing, the maximum interval is dependent upon CPU processing and the value of the automatic data transmit/receive interval specify register (ADTI) (see (5) Automatic data transmit/receive interval).

2. When TRF is cleared, the SO1 pin becomes low level.

Remark CSIIF1 : Interrupt request flag
TRF : Bit 3 of automatic data transmit/receive control register (ADTC)

Figure 18-9. Basic Transmission/Reception Mode Flowchart


Start
Write transmit data in internal buffer RAM
Set ADTP to the value (pointer value) obtained by subtracting 1 from the number of transmit data bytes
Set the transmission/reception operation interval time in ADTI
Write any data to SIO1 (Start trigger)
Write transmit data from internal buffer RAM to SIO1
Transmission/reception operation
Decrement pointer value
Write receive data from SIO1 to internal buffer RAM
Pointer value = 0
No
Yes
TRF = 0
No
Yes
End
Software Execution
Hardware Execution
Software Execution

ADTP:	Automatic data transmit/receive address pointer
ADTI:	Automatic data transmit/receive interval specify register
SIO1:	Serial I/O shift register 1
TRF:	Bit 3 of automatic data transmit/receive control register (ADTC)

In 6-byte transmission/reception (ARLD = 0, RE = 1) in basic transmit/receive mode, internal buffer RAM operates as follows.

(i) Before transmission/reception (see Figure 18-10 (a).)

After any data has been written to serial I/O shift register 1 (SIO1) (start trigger: this data is not transferred), transmit data 1 (T1) is transferred from the internal buffer RAM to SIO1. When transmission of the first byte is completed, the receive data 1 (R1) is transferred from SIO1 to the buffer RAM, and automatic data transmit/receive address pointer (ADTP) is decremented. Then transmit data 2 (T2) is transferred from the internal buffer RAM to SIO1.

(ii) 4th byte transmission/reception point (see Figure 18-10 (b).)

Transmission/reception of the third byte is completed, and transmit data 4 (T4) is transferred from the internal buffer RAM to SIO1. When transmission of the fourth byte is completed, the receive data 4 (R4) is transferred from SIO1 to the internal buffer RAM, and ADTP is decremented.

(iii) Completion of transmission/reception (see Figure 18-10 (c).)

When transmission of the sixth byte is completed, the receive data 6 (R6) is transferred from SIO1 to the internal buffer RAM, and the interrupt request flag (CSIIF1) is set (INTCSI1 generation).

Figure 18-10. Internal Buffer RAM Operation in 6-byte Transmission/Reception (in Basic Transmit/Receive Mode) (1/2)

(a) Before transmission/reception

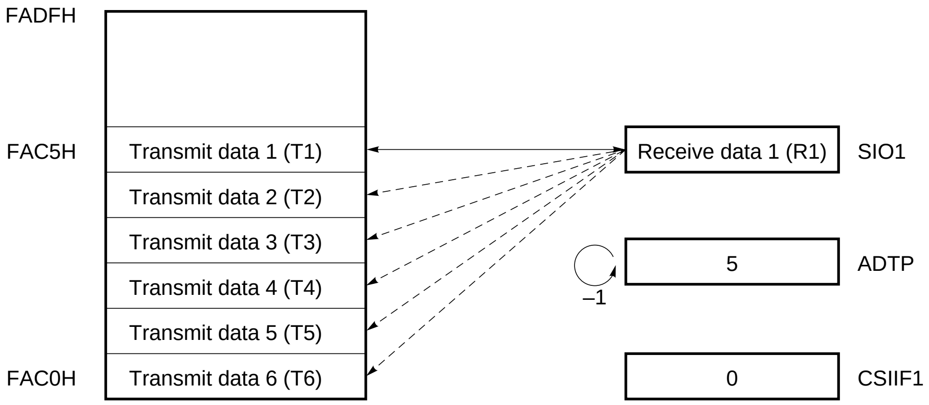
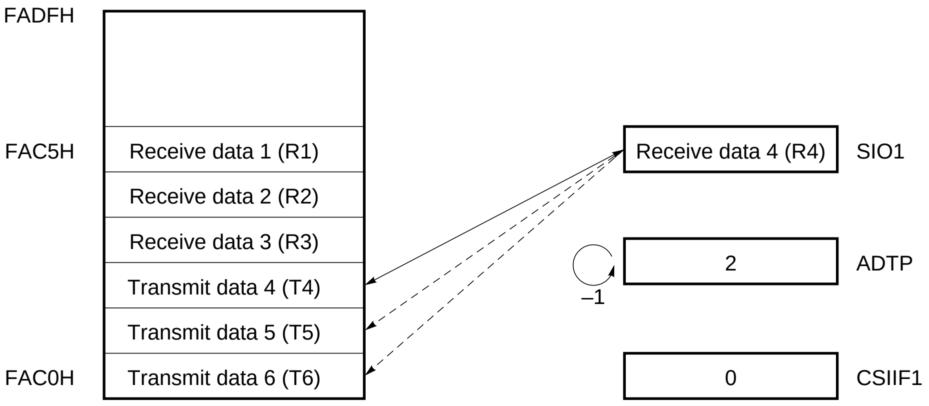
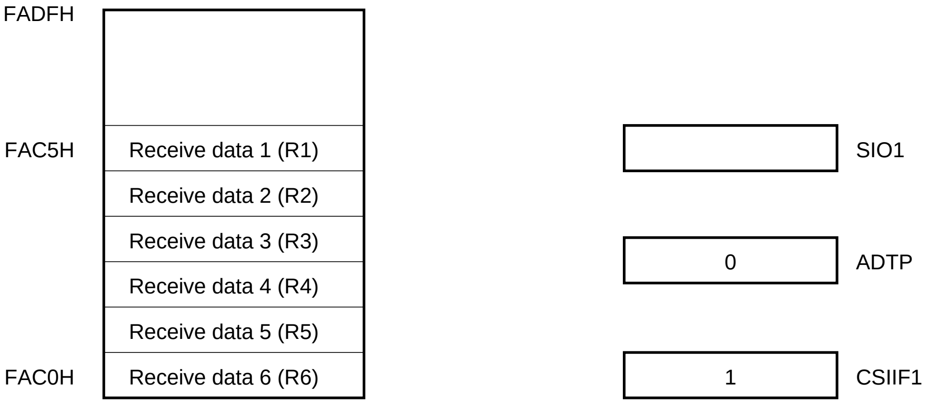


Figure 18-10. Internal Buffer RAM Operation in 6-byte Transmission/Reception (in Basic Transmit/Receive Mode) (2/2)

(b) 4th byte transmission/reception



(c) Completion of transmission/reception



(b) Basic transmission mode

In this mode, the specified number of 8-bit unit data are transmitted.

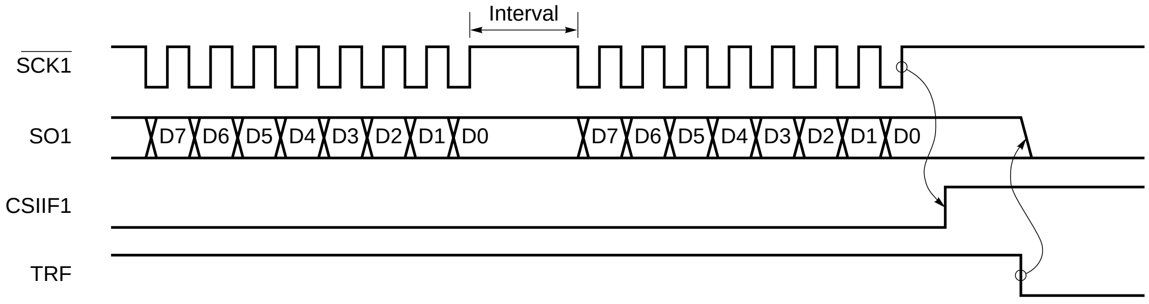
Serial transfer is started when any data is written to the serial I/O shift register 1 (SIO1) while bit 7 (CSIE1) of the serial operating mode register 1 (CSIM1) is set to 1.

When the final byte has been sent, an interrupt request flag (CSIIF1) is set. However, judge the termination of automatic transmit and receive, not by CSIIF1 but by bit 3 (TRF) of the automatic data transmit/receive control register (ADTC).

If receive operation, busy control and strobe control are not executed, the P20/SI1, P23/STB/TxD1 and P24/BUSY/RxD1 pins can be used as normal input/ports.

Figure 18-11 shows the basic transmission mode operation timings, and Figure 18-12 shows the operation flowchart. Figure 18-13 shows the operation of the internal buffer RAM when 6 bytes of data are transmitted or received.

Figure 18-11. Basic Transmission Mode Operation Timings



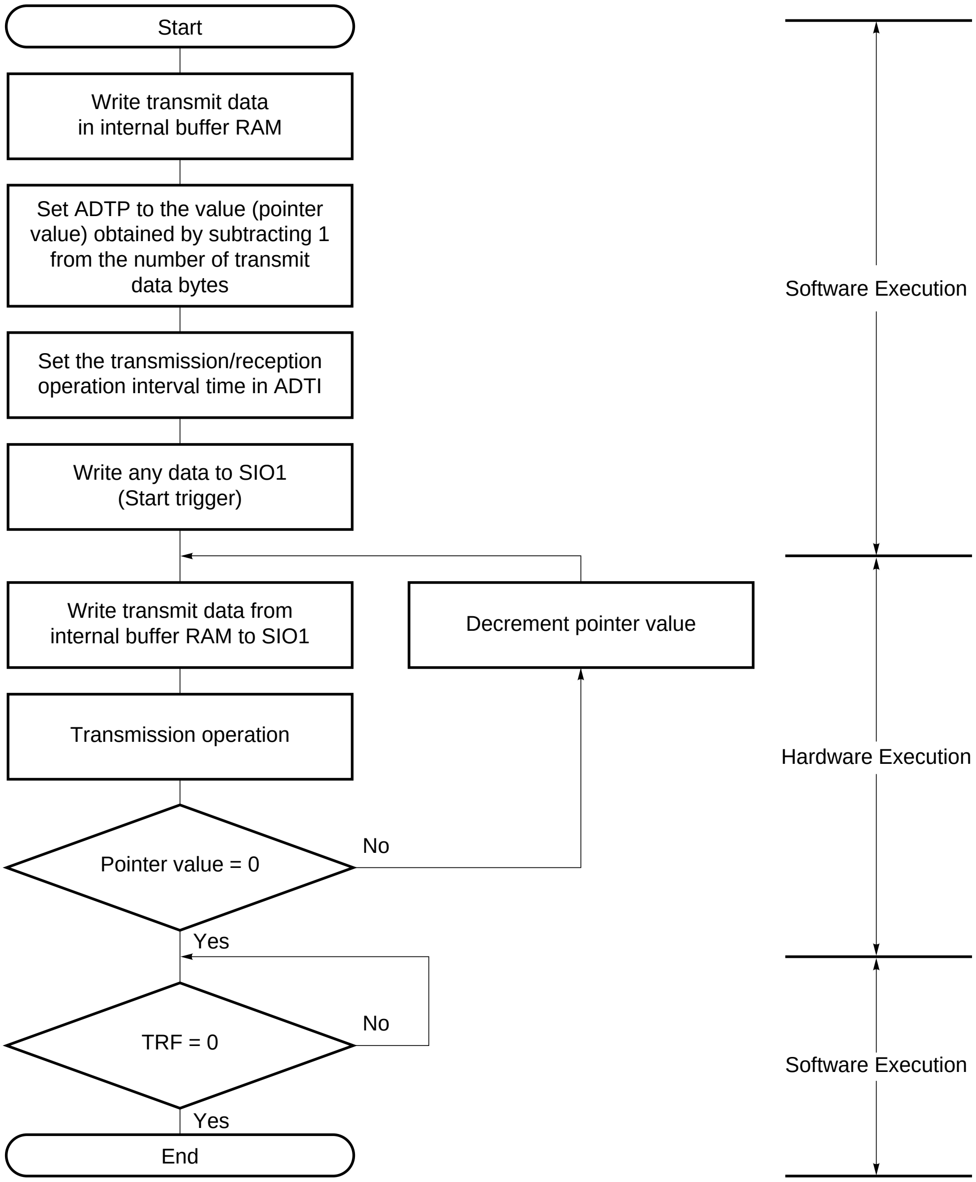
Cautions 1. Because, in the basic transmission mode, the automatic transmit/receive function reads data from the internal buffer RAM after 1-byte transmission, an interval is inserted till the next transmission. As the buffer RAM read is performed at the same time as CPU processing, the maximum interval is dependent upon CPU processing and the value of the automatic data transmit/receive interval specify register (ADTI) (see (5) Automatic data transmit/receive interval).

2. When TRF is cleared, the SO1 pin becomes low level.

Remark CSIIF1 : Interrupt request flag

TRF : Bit 3 of automatic data transmit/receive control register (ADTC)

Figure 18-12. Basic Transmission Mode Flowchart


Start
Write transmit data in internal buffer RAM
Set ADTP to the value (pointer value) obtained by subtracting 1 from the number of transmit data bytes
Set the transmission/reception operation interval time in ADTI
Write any data to SIO1 (Start trigger)
Write transmit data from internal buffer RAM to SIO1
Decrement pointer value
Transmission operation
Pointer value = 0
No
Yes
TRF = 0
No
Yes
End
Software Execution
Hardware Execution
Software Execution

ADTP:	Automatic data transmit/receive address pointer
ADTI:	Automatic data transmit/receive interval specify register
SIO1:	Serial I/O shift register 1
TRF:	Bit 3 of automatic data transmit/receive control register (ADTC)

In 6-byte transmission (ARLD = 0, RE = 0) in basic transmit mode, internal buffer RAM operates as follows.

(i) Before transmission (see Figure 18-13 (a).)

After any data has been written to serial I/O shift register 1 (SIO1) (start trigger: this data is not transferred), transmit data 1 (T1) is transferred from the internal buffer RAM to SIO1. When transmission of the first byte is completed, automatic data transmit/receive address pointer (ADTP) is decremented. Then transmit data 2 (T2) is transferred from the internal buffer RAM to SIO1.

(ii) 4th byte transmission point (see Figure 18-13 (b).)

Transmission of the third byte is completed, and transmit data 4 (T4) is transferred from the internal buffer RAM to SIO1. When transmission of the fourth byte is completed, ADTP is decremented.

(iii) Completion of transmission (see Figure 18-13 (c).)

When transmission of the sixth byte is completed, the interrupt request flag (CSIIF1) is set (INTCSI1 generation).

Figure 18-13. Internal Buffer RAM Operation in 6-byte Transmission (in Basic Transmit Mode) (1/2)

(a) Before transmission

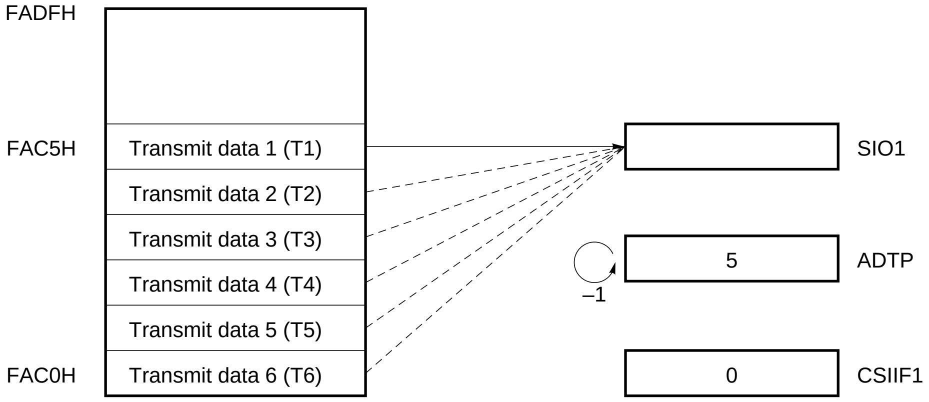
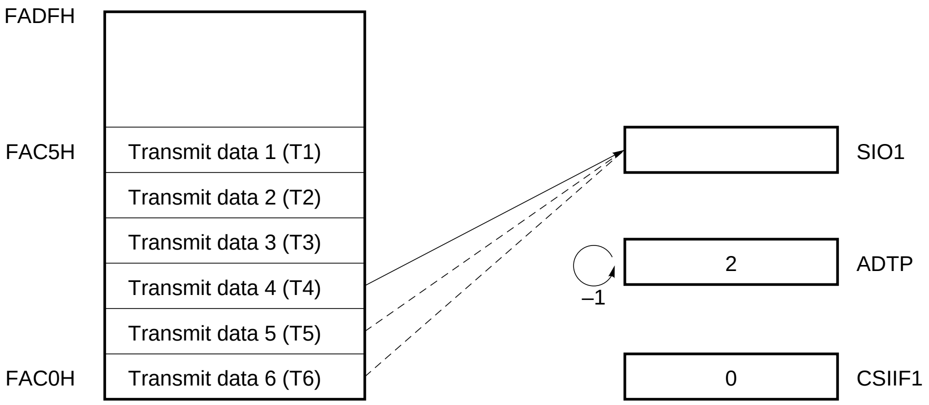
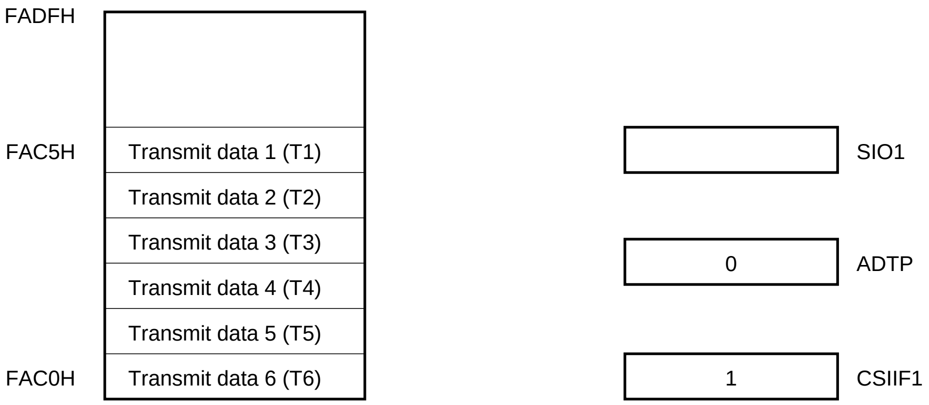


Figure 18-13.   Internal Buffer RAM Operation in 6-byte Transmission (in Basic Transmit Mode) (2/2)

(b) 4th byte transmission point



(c) Completion of transmission



(c) Repeat transmission mode

In this mode, data stored in the internal buffer RAM is transmitted repeatedly.

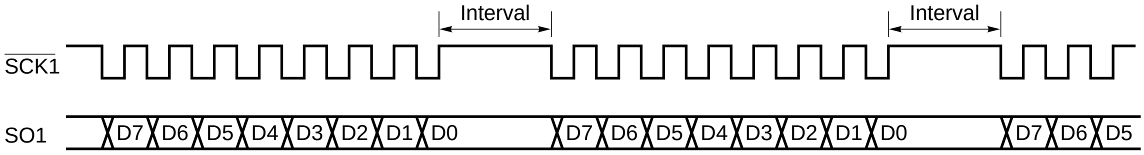
Serial transmission is started by writing any data to serial I/O shift register 1 (SIO1) when 1 is set in bit 7 (CSIE1) of the serial operating mode register 1 (CSIM1).

Unlike the basic transmission mode, after the final byte (data in address FAC0H) has been transmitted, the interrupt request flag (CSIIF1) is not set, the value at the time when transmission was started is set in the automatic data transmit/receive address pointer (ADTP) again, and the internal buffer RAM contents are transmitted again.

When a reception operation, busy control and strobe control are not performed, the P20/SI1, P23/STB/TxD1 and P24/BUSY/RxD1 pins can be used as ordinary input/output ports.

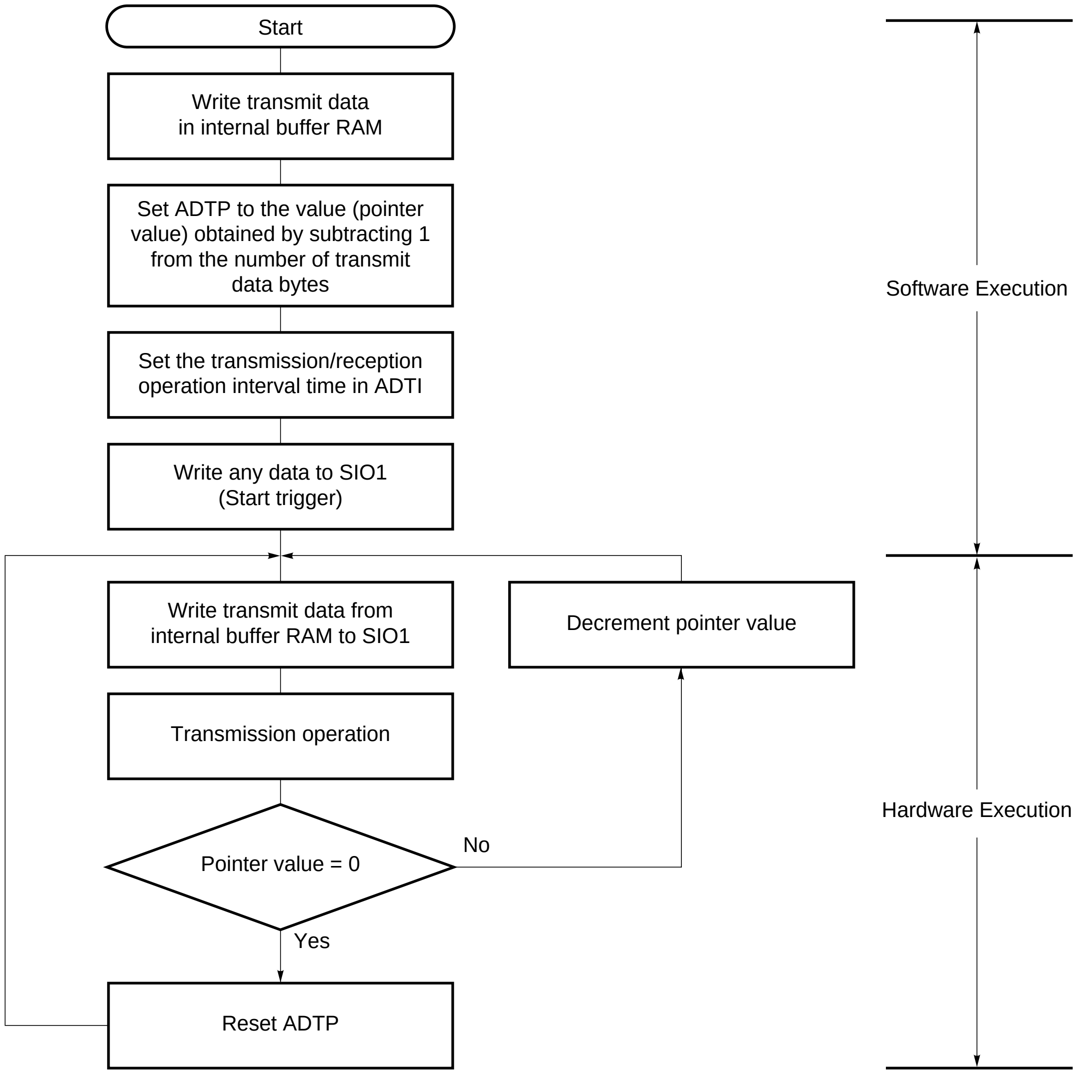
The repeat transmission mode operation timing is shown in Figure 18-14, and the operation flowchart in Figure 18-15. Figure 18-16 shows the operation of the internal buffer RAM when 6 bytes of data are transmitted in the repeat transmission mode.

Figure 18-14. Repeat Transmission Mode Operation Timing



Caution **Because, in the repeat transmission mode, a read is performed on the buffer RAM after the transmission of one byte, the interval is included in the period up to the next transmission. As the buffer RAM read is performed at the same time as CPU processing, the maximum interval is dependent upon the CPU operation and the value of the automatic data transmit/receive interval specify register (ADTI) (see (5) Automatic data transmit/receive interval).**

Figure 18-15. Repeat Transmission Mode Flowchart


Start
Write transmit data in internal buffer RAM
Set ADTP to the value (pointer value) obtained by subtracting 1 from the number of transmit data bytes
Set the transmission/reception operation interval time in ADTI
Write any data to SIO1 (Start trigger)
Write transmit data from internal buffer RAM to SIO1
Decrement pointer value
Transmission operation
Pointer value = 0
No
Yes
Reset ADTP
Software Execution
Hardware Execution

ADTP: Automatic data transmit/receive address pointer
ADTI: Automatic data transmit/receive interval specify register
SIO1:   Serial I/O shift register 1

In 6-byte transmission (ARLD = 1, RE = 0) in repeat transmit mode, internal buffer RAM operates as follows.

(i) Before transmission (see Figure 18-16 (a).)
After any data has been written to serial I/O shift register 1 (SIO1) (start trigger: this data is not transferred), transmit data 1 (T1) is transferred from the internal buffer RAM to SIO1. When transmission of the first byte is completed, automatic data transmit/receive address pointer (ADTP) is decremented. Then transmit data 2 (T2) is transferred from the internal buffer RAM to SIO1.

(ii) Upon completion of transmission of 6 bytes (see Figure 18-16 (b).)
When transmission of the sixth byte is completed, the interrupt request flag (CSIIF1) is not set. The internal pointer value is reset in ADTP.

(iii) 7th byte transmission point (see Figure 18-16 (c).)
Transmit data 1 (T1) is transferred from the internal buffer RAM to SIO1 again. When transmission of the first byte is completed, ADTP is decremented. Then transmit data 2 (T2) is transferred from the internal buffer RAM to SIO1.

Figure 18-16. Internal Buffer RAM Operation in 6-byte Transmission (in Repeat Transmit Mode) (1/2)

(a) Before transmission

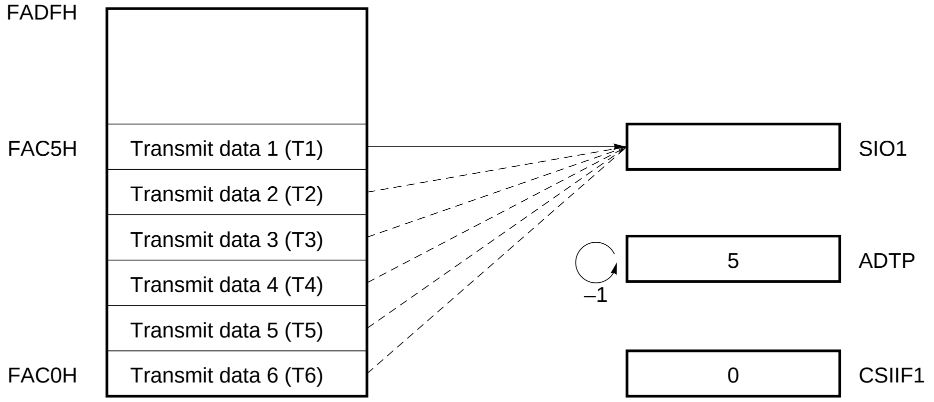
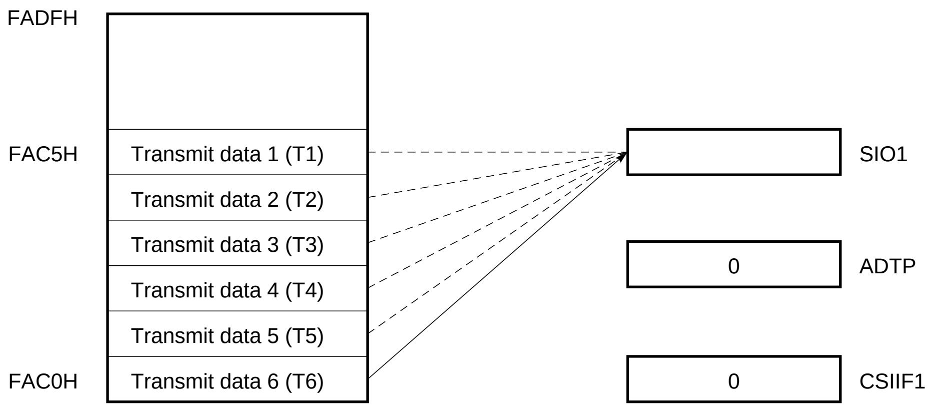
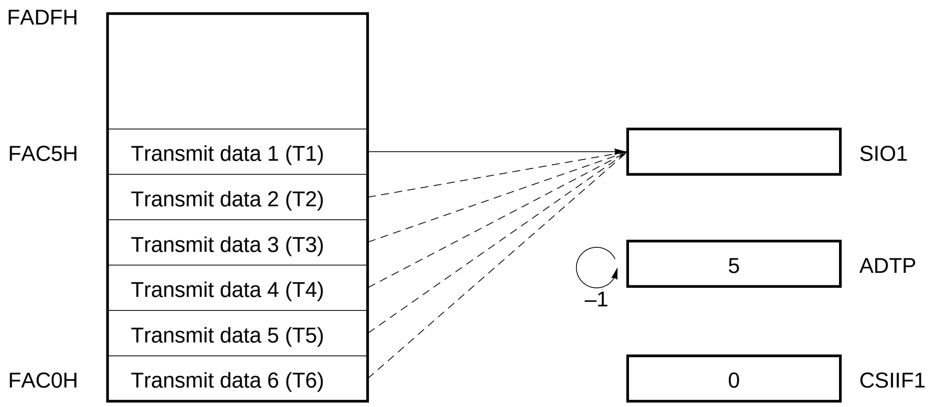


Figure 18-16. Internal Buffer RAM Operation in 6-byte Transmission (in Repeat Transmit Mode) (2/2)

(b) Upon completion of transmission of 6 bytes



(c) 7th byte transmission point



(d) Automatic transmission/reception suspending and restart

Automatic transmission/reception can be temporarily suspended by setting bit 7 (CSIE1) of the serial operating mode register 1 (CSIM1) to 0.

If during 8-bit data transfer, the transmission/reception is not suspended if bit 7 (CSIE1) is set to 0. It is suspended upon completion of 8-bit data transfer.

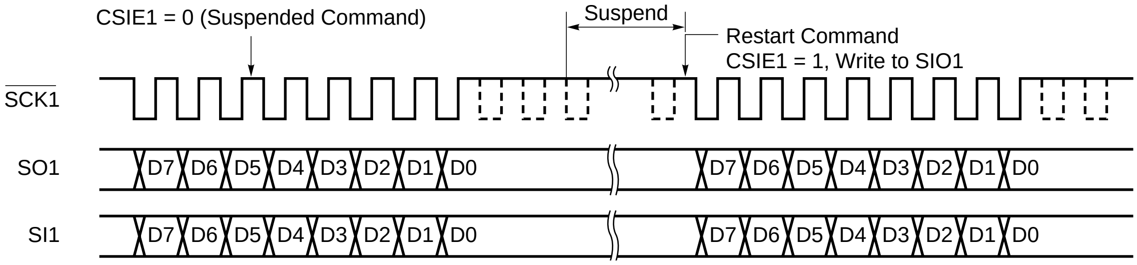
When suspended, bit 3 (TRF) of the automatic data transmit/receive control register (ADTC) is set to 0 after transfer of the 8th bit, and all the port pins used with the serial interface pins for dual function (P20/SI1, P21/SO1, P22/$\overline{\text{SCK1}}$, P23/STB/TxD1 and P24/BUSY/RxD1) are set to the port mode.

To restart automatic transmission/reception, set CSIE1 at 1 and write the desired value in serial I/O shift register 1 (SIO1). The remaining data can be transmitted in this way.

Cautions 1. If the HALT instruction is executed during automatic transmission/reception, transfer is suspended and the HALT mode is set if during 8-bit data transfer. When the HALT mode is cleared, automatic transmission/reception is restarted from the suspended point.

2. When suspending automatic transmission/reception, do not change the operating mode to 3-wire serial I/O mode while TRF = 1.

Figure 18-17. Automatic Transmission/Reception Suspension and Restart



CSIE1: Bit 7 of serial operating mode register 1 (CSIM1)

★ **(4) Synchronization control**

Busy control and strobe control are functions to synchronize transmission/reception between the master device and a slave device.

By using these functions, a shift in bits being transmitted or received can be detected.

(a) Busy control option

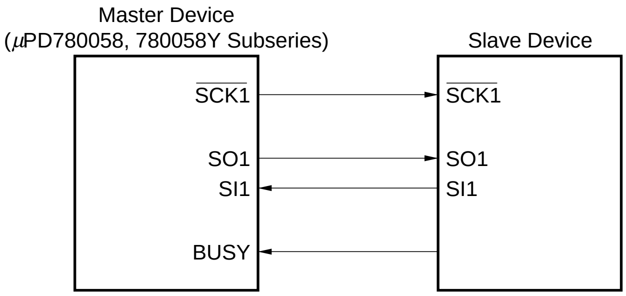
Busy control is a function to keep the serial transmission/reception by the master device waiting while the busy signal output by a slave device to the master is active.

When using this busy control option, the following conditions must be satisfied.

- Bit 5 (ATE) of the serial operating mode register 1 (CSIM1) is set to 1.
- Bit 1 (BUSY1) of the automatic data transmit/receive control register (ADTC) is set to 1.

Figure 18-18 shows the system configuration of the master device and a slave device when the busy control option is used.

Figure 18-18. System Configuration when Busy Control Option Is Used



The master device inputs the busy signal output by the slave device to the BUSY/P24 pin. The master device samples the input busy signal in synchronization with the falling of the serial clock. Even if the busy signal becomes active while 8-bit data is being transmitted or received, transmission/reception by the master is not kept waiting. If the busy signal is active at the rising edge of the serial clock 2 clocks after completion of transmission/reception of the 8-bit data, the busy input becomes valid. After that, the master transmission/reception is kept waiting while the busy signal is active.

The active level of the busy signal is set by bit 0 (BUSY0) of ADTC.

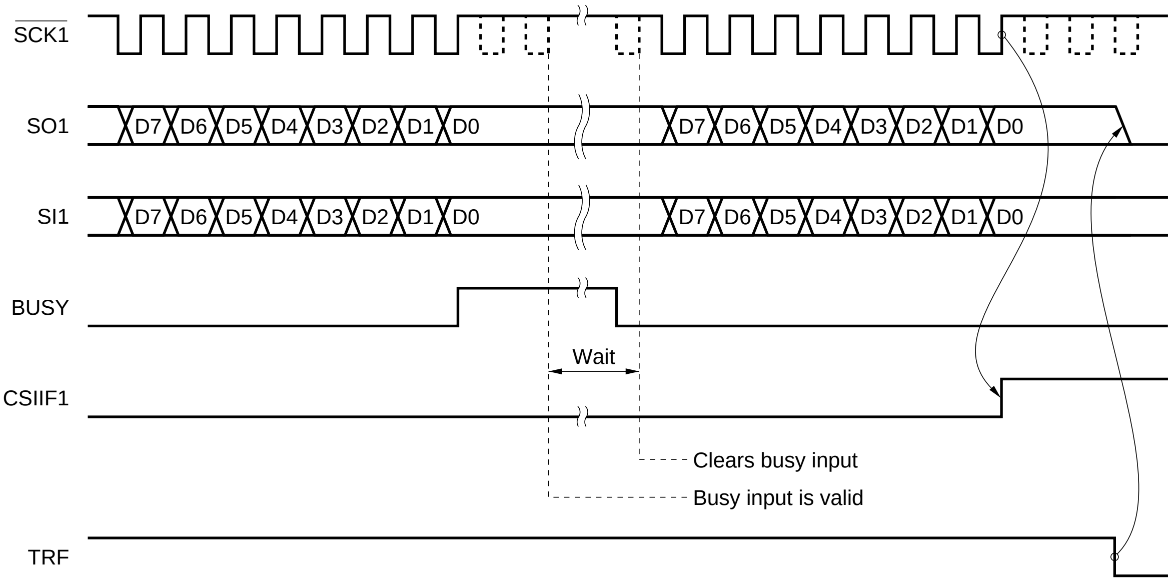
BUSY0 = 0: Active high
BUSY0 = 1: Active low

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 18-19 shows the operation timing when the busy control option is used.

Caution Busy control cannot be used simultaneously with the interval time control function of the automatic data transmit/receive interval specification register (ADTI). If used, busy control is invalid.

Figure 18-19. Operation Timing when Busy Control Option Is Used (when BUSY0 = 0)

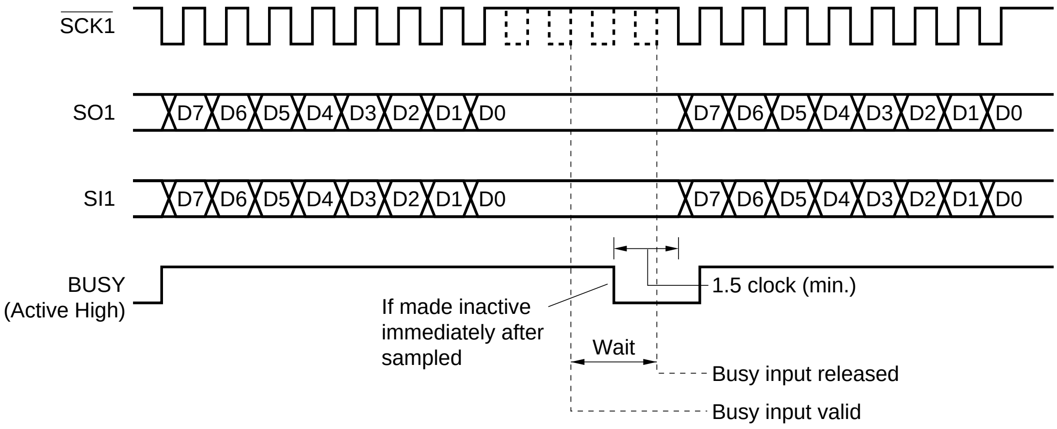


Caution If the TRF is cleared, the SO1 pin goes low.

Remark CSIIF1: Interrupt request flag
TRF : Bit 3 of automatic data transmit/receive control register (ADTC)

When the busy signal becomes inactive, waiting is released. If the sampled busy signal is inactive, transmission/reception of the next 8-bit data is started at the falling edge of the next clock.
Because the busy signal is asynchronous with the serial clock, it takes up to 1 clock until the busy signal, even if made inactive by the slave, is sampled. It takes 0.5 clock until data transfer is started after the busy signal was sampled.
To accurately release waiting, the slave must keep the busy signal inactive at least for the duration of 1.5 clock.
Figure 18-20 shows the timing of the busy signal and releasing the waiting. This figure shows an example where the busy signal is active as soon as transmission/reception has been started.

Figure 18-20. Busy Signal and Wait Release (when BUSY0 = 0)



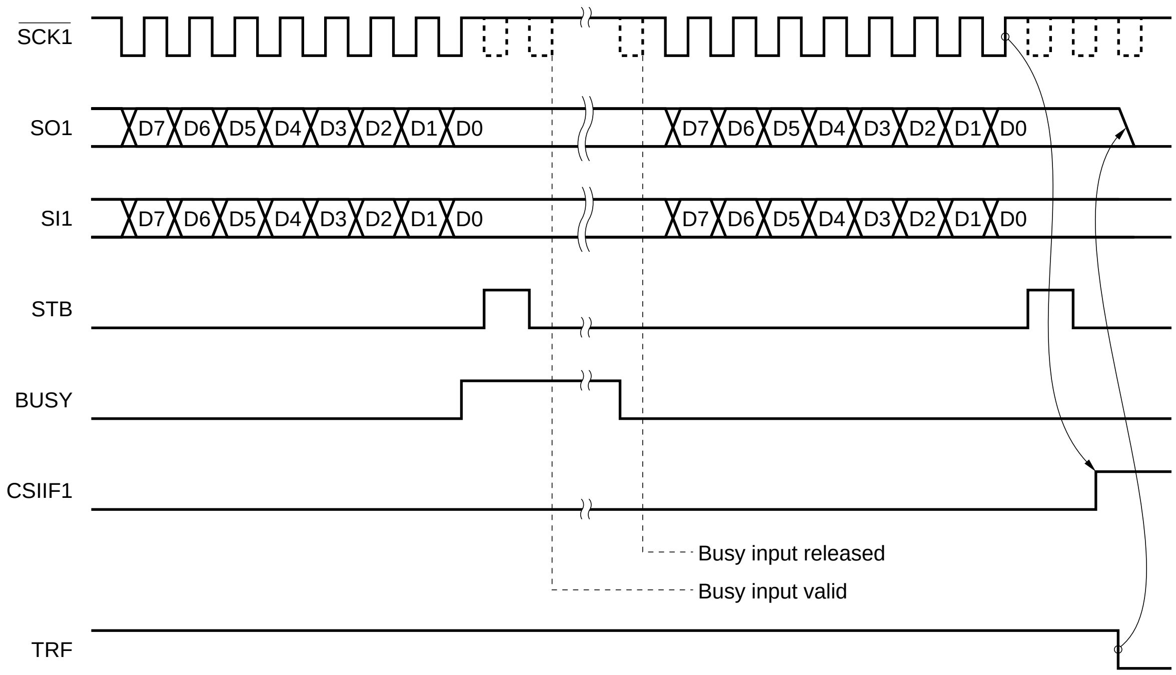
(b) Busy & strobe control option

Strobe control is a function to synchronize data transmission/reception between the master and slave devices. The master device outputs the strobe signal from the STB/P23 pin when 8-bit transmission/reception has been completed. By this signal, the slave device can determine the timing of the end of data transmission. Therefore, synchronization is established even if a bit shift occurs because noise is superimposed on the serial clock, and transmission of the next byte is not affected by the bit shift.
To use the strobe control option, the following conditions must be satisfied:

- Bit 5 (ATE) of the serial operating mode register 1 (CSIM1) is set to 1.
- Bit 2 (STRB) of the automatic data transmit/receive control register (ADTC) is set to 1.

Usually, the busy control and strobe control options are simultaneously used as handshake signals. In this case, the strobe signal is output from the STB/P23 pin, and the BUSY/P24 pin is sampled, and transmission/reception can be kept waiting while the busy signal is input.
When the strobe control option is not used, the P23/STB pin can be used as a normal I/O port pin.
Figure 18-21 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 18-21. Operation Timing when Busy & Strobe Control Options Are Used (when BUSY0 = 0)



Caution When TRF is cleared, the SO1 pin goes low.

Remark CSIIF1: Interrupt request flag

TRF : Bit 3 of automatic data transmit/receive control register (ADTC)

(c) Bit shift detection by busy signal

During automatic transmission/reception, a bit shift of the serial clock of the slave device may occur because noise is superimposed on the serial clock signal output by the master device. Unless the strobe control option is used at this time, the bit shift affects transmission of the next byte. In this case, the master can detect the bit shift by checking the busy signal during transmission by using the busy control option.

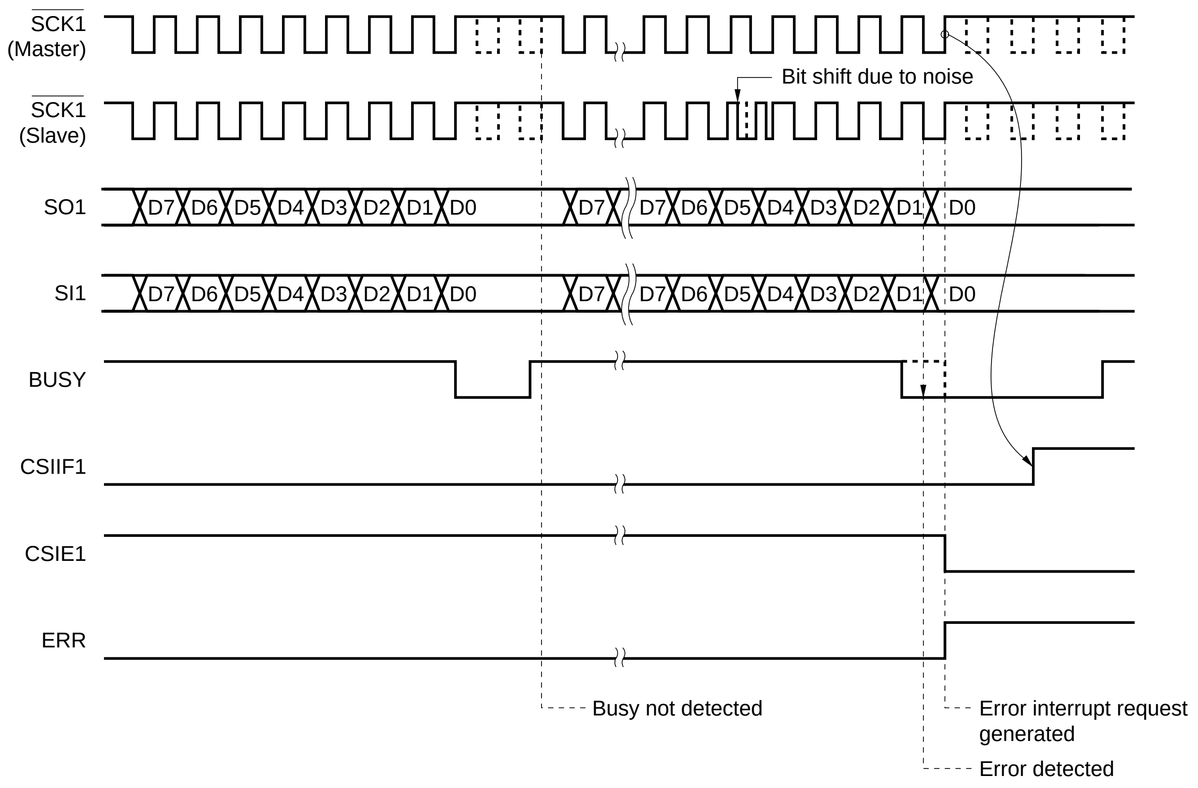
A bit shift is detected by using the busy signal as follows:

The slave outputs the busy signal after the rising of the eighth serial clock during data transmission/ reception (to not keep transmission/reception waiting by the busy 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 (ERR) of the automatic data transmit/receive control register (ADTC) to 1).

Figure 18-22 shows the operation timing of the bit shift detection function by the busy signal.

Figure 18-22. Operation Timing of Bit Shift Detection Function by Busy Signal (when BUSY0 = 1)


SCK1
(Master)
SCK1
(Slave)
Bit shift due to noise
SO1
D7 D6 D5 D4 D3 D2 D1 D0 D7 D7 D6 D5 D4 D3 D2 D1 D0
SI1
D7 D6 D5 D4 D3 D2 D1 D0 D7 D7 D6 D5 D4 D3 D2 D1 D0
BUSY
CSIIF1
CSIE1
ERR
Busy not detected
Error interrupt request generated
Error detected

CSIIF1 : Interrupt request flag

CSIE1 : Bit 7 of serial operating mode register 1 (CSIM1)

ERR : Bit 4 of automatic data transmit/receive control register (ADTC)

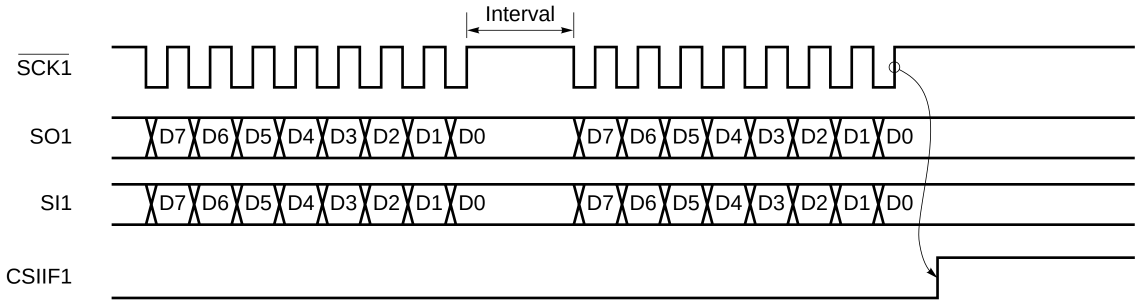
(5) Automatic transmit/receive interval time

When using the automatic transmit/receive function, the read/write operations from/to the internal buffer RAM are performed after transmitting/receiving one byte. Therefore, an interval is inserted before the next transmit/ receive.

Since the read/write operations from/to the buffer RAM are performed in parallel with the CPU processing when using the automatic transmit/receive function by the internal clock, the interval depends on the value which is set in the automatic transmit/receive interval specification register (ADTI) and the CPU processing at the rising edge of the eighth serial clock. Whether it depends on the ADTI or not can be selected by the setting of its bit 7 (ADTI7). When it is set to 0, the interval depends only on the CPU processing. When it is set to 1, the interval depends on the contents of the ADTI or CPU processing, whichever is greater.

When the automatic transmit/receive function is used by an external clock, it must be selected so that the interval may be longer than the value indicated by paragraph (b).

Figure 18-23. Automatic Data Transmit/Receive Interval Time



CSIIF1: Interrupt request flag

(a) When the automatic transmit/receive function is used by the internal clock

If the bit 1 (CSIM11) of serial operating mode register (CSIM1) is set at (1), the internal clock operates. If the automatic transmit/receive function is operated by the internal clock, interval timing by CPU processing is as follows.

When bit 7 (ADTI7) of automatic data transmit/receive interval specify register (ADTI) is set to 0, the interval depends on the CPU processing. When ADTI7 is set to 1, it depends on the contents of the ADTI or CPU processing, whichever is greater.

Refer to **Figure 18-5 Automatic Data Transmit/Receive Interval Specify Register Format** for the intervals which are set by the ADTI.

Table 18-2. Interval Timing through CPU Processing (When the Internal Clock Is Operating)

CPU Processing	Interval Time
When using multiplication instruction	Max. (2.5T_{SCK}, 13T_{CPU})
When using division instruction	Max. (2.5T_{SCK}, 20T_{CPU})
External access 1 wait mode	Max. (2.5T_{SCK}, 9T_{CPU})
Other than above	Max. (2.5T_{SCK}, 7T_{CPU})

T_{SCK} : 1/f_{SCK}

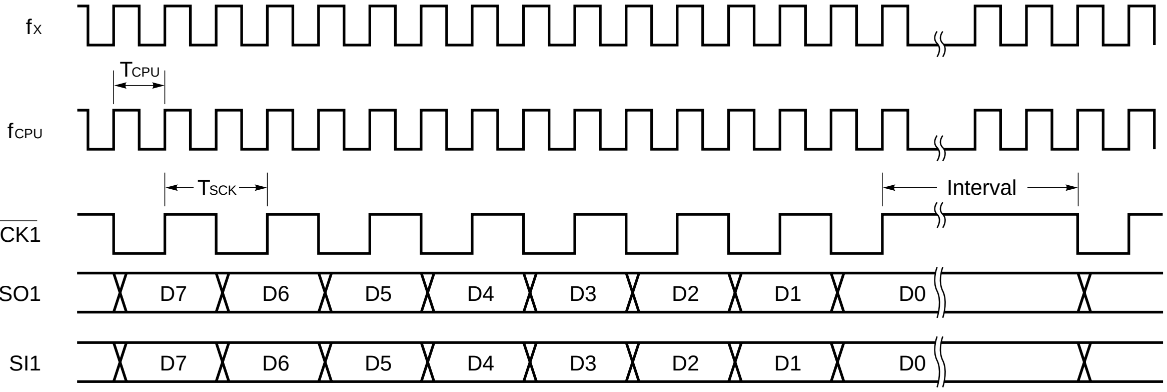
f_{SCK} : Serial clock frequency

T_{CPU} : 1/f_{CPU}

f_{CPU} : CPU clock (set by bits 0 to 2 (PCC0 to PCC2) of the processor clock control register (PCC) and bit 0 (MCS) of the oscillation mode selection register (OSMS))

MAX. (a, b): a or b, whichever is greater

Figure 18-24. Operation Timing with Automatic Data Transmit/Receive Function Performed by Internal Clock



f_X : Main system clock oscillation frequency

f_{CPU} : CPU clock (set by bits 0 to 2 (PCC0 to PCC2) of the processor clock control register (PCC))

T_{CPU} : 1/f_{CPU}

T_{SCK} : 1/f_{SCK}

f_{SCK} : Serial clock frequency

(b) When using automatic transmit/receive function with external clock

The external clock is used when bit 1 (CSIM11) of the serial operating mode register 1 (CSIM1) is cleared to 0.

To use the automatic transmit/receive function with the external clock, the external clock must be input such that the interval time is as follows:

Table 18-3. Interval Time by CPU Processing (with External Clock)

CPU Processing	Interval Time
When using multiplication instruction	13T_{CPU} or more
When using division instruction	20T_{CPU} or more
External access 1 wait mode	9T_{CPU} or more
Other than above	7T_{CPU} or more

T_{CPU} : 1/f_{CPU}

f_{CPU} : CPU clock (set by the bits 0 to 2 (PCC0 to PCC2) of the processor clock control register (PCC) and bit 0 (MCS) of the oscillation mode selection register (OSMS))

[MEMO]

CHAPTER 19 SERIAL INTERFACE CHANNEL 2

19.1 Serial Interface Channel 2 Functions

Serial interface channel 2 has the following three modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode (with time-division transfer function)
- 3-wire serial I/O mode

(1) Operation stop mode

This mode is used when serial transfer is not carried out to reduce power consumption.

(2) Asynchronous serial interface (UART) mode (with time-division transfer function)

In this mode, one byte of data is transmitted/received following the start bit, and full-duplex operation is possible.

A dedicated UART baud rate generator is incorporated, allowing communication over a wide range of baud rates. In addition, the baud rate can be defined by scaling the input clock to the ASCK pin.

The MIDI standard baud rate (31.25 kbps) can be used by employing the dedicated UART baud rate generator.

Two sets of data I/O pins (RxD and TxD) are provided, and the pin to be used can be selected by software (time-division transfer function). However, only one set of pins can be used at one time.

Cautions 1. If it is not necessary to change the data I/O pin, use of the RxD0/SI2/P70 and TxD0/SO2/P71 pins is recommended. If only port 2 (RxD1/BUSY/P24 and TxD1/STB/P23) is used as data I/O pins, the function of port 7 is limited.

2. When using the busy control option or busy & strobe control option in the three-wire serial I/O mode with automatic transmit/receive function of the serial interface channel 1, the RxD1/BUSY/P24 and TxD1/STB/P23 pins cannot be used as data I/O pins.

(3) 3-wire serial I/O mode (MSB-first/LSB-first switchable)

In this mode, 8-bit data transfer is performed using three lines: the serial clock ($\overline{\text{SCK2}}$), and serial data lines (SI2, SO2).

In the 3-wire serial I/O mode, simultaneous transmission and reception is possible, increasing the data transfer processing speed.

Either the MSB or LSB can be specified as the start bit for an 8-bit data serial transfer, allowing connection to devices using either as the start bit.

The 3-wire serial I/O mode is useful for connection to peripheral I/Os and display controllers, etc., which incorporate a conventional synchronous clocked serial interface, such as the 75X/XL Series, 78K Series, 17K Series, etc.

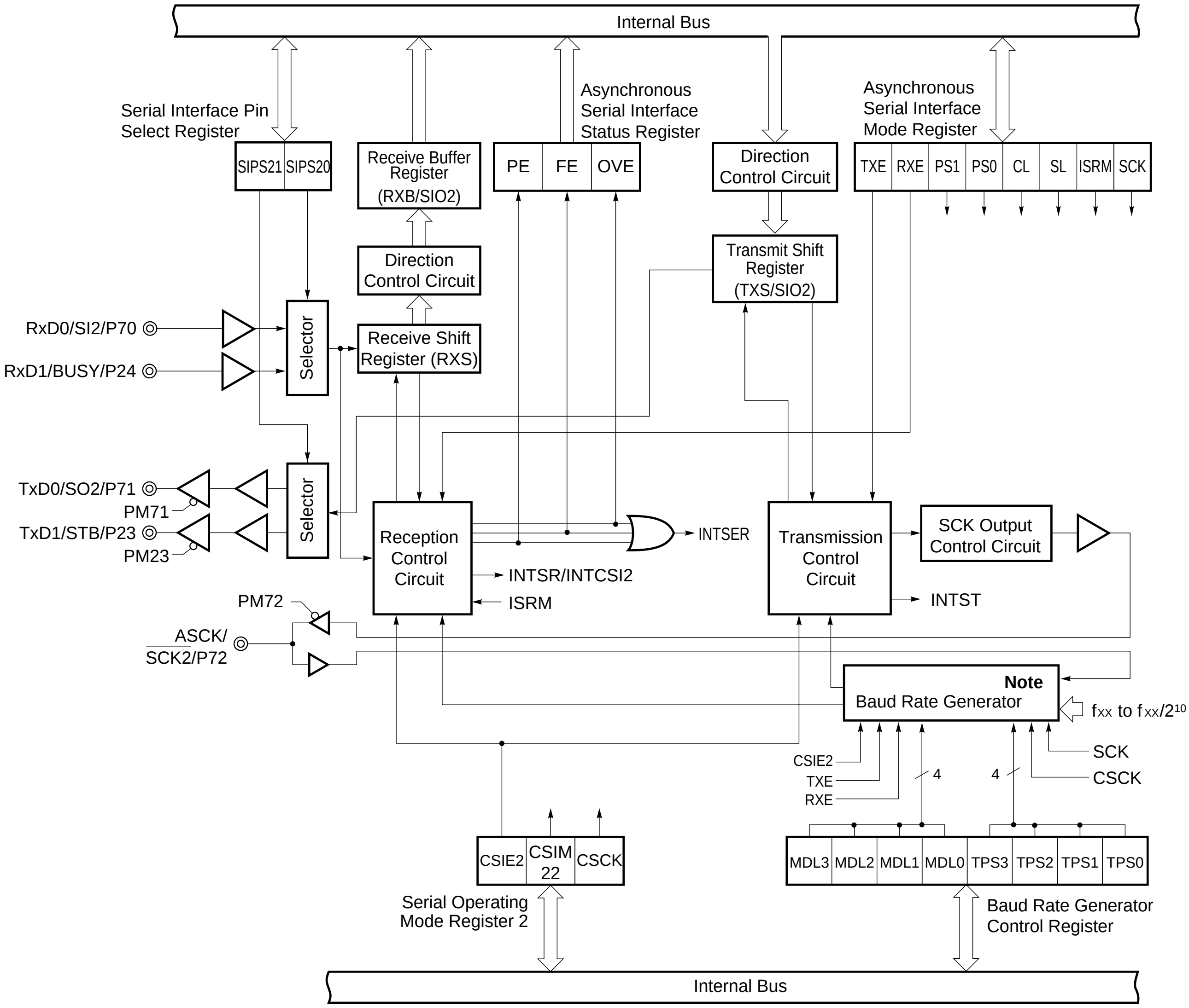
19.2 Serial Interface Channel 2 Configuration

Serial interface channel 2 consists of the following hardware.

Table 19-1. Serial Interface Channel 2 Configuration

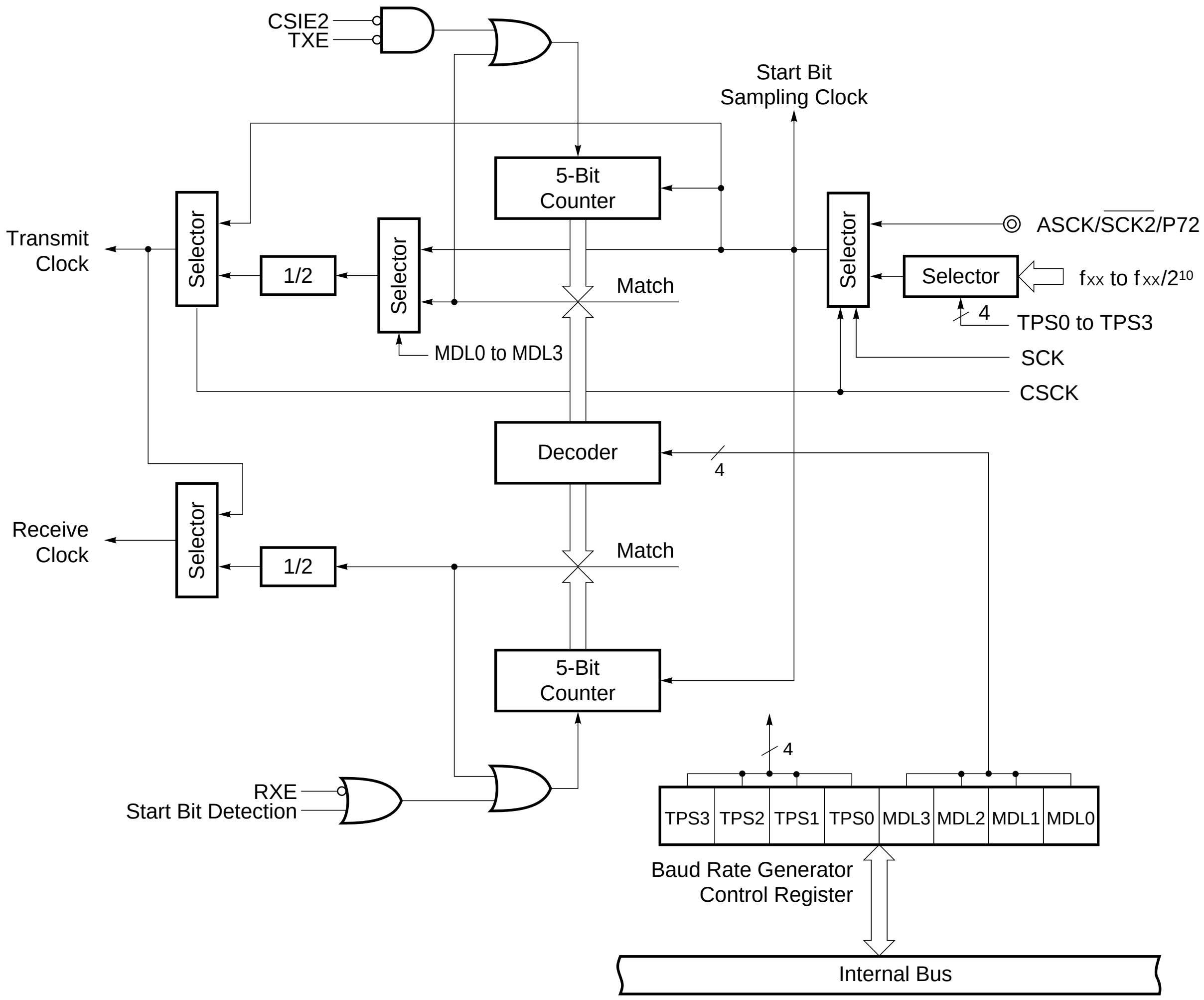
Item	Configuration
Register	Transmit shift register (TXS) Receive shift register (RXS) Receive buffer register (RXB)
Control register	Serial operating mode register 2 (CSIM2) Asynchronous serial interface mode register (ASIM) Asynchronous serial interface status register (ASIS) Baud rate generator control register (BRGC) Serial interface pin select register (SIPS)

Figure 19-1. Serial Interface Channel 2 Block Diagram


Internal Bus
Serial Interface Pin Select Register
SIPS21
SIPS20
Receive Buffer Register (RXB/SIO2)
Asynchronous Serial Interface Status Register
PE
FE
OVE
Direction Control Circuit
Asynchronous Serial Interface Mode Register
TXE
RXE
PS1
PS0
CL
SL
ISRM
SCK
Transmit Shift Register (TXS/SIO2)
Direction Control Circuit
RxD0/SI2/P70
RxD1/BUSY/P24
Selector
Receive Shift Register (RXS)
TxD0/SO2/P71
PM71
TxD1/STB/P23
PM23
Selector
Reception Control Circuit
INTSER
Transmission Control Circuit
SCK Output Control Circuit
INTSR/INTCSI2
ISRM
INTST
PM72
ASCK/
SCK2/P72
Note
Baud Rate Generator
fxx to fxx/2¹⁰
SCK
CSCK
CSIE2
TXE
RXE
4
4
CSIE2
CSIM 22
CSCK
MDL3
MDL2
MDL1
MDL0
TPS3
TPS2
TPS1
TPS0
Serial Operating Mode Register 2
Baud Rate Generator Control Register
Internal Bus

Note Refer to Figure 19-2 for the baud rate generator configuration.

Figure 19-2. Baud Rate Generator Block Diagram



(1) Transmit shift register (TXS)

This register is used to set the transmit data. The data written in the TXS is transmitted as serial data.
If the data length is specified as 7 bits, bits 0 to 6 of the data written in the TXS are transferred as transmit data. Writing data to the TXS starts the transmit operation.
The TXS is written to with an 8-bit memory manipulation instruction. It cannot be read.
$\overline{\text{RESET}}$ input sets the TXS to FFH.

Caution TXS must not be written to during a transmit operation. TXS and the receive buffer register (RXB) are allocated to the same address, and when a read is performed, the value of RXB is read.

(2) Receive shift register (RXS)

This register is used to convert serial data input to the RxD0 (RxD1) pin to parallel data. When one byte of data is received, the receive data is transferred to the receive buffer register (RXB).
RXS cannot be directly manipulated by a program.

(3) Receive buffer register (RXB)

This register holds receive data. Each time one byte of data is received, new receive data is transferred from the receive shift register (RXS).
If the data length is specified as 7 bits, the receive data is transferred to bits 0 to 6 of the RXB, and the MSB of the RXB is always set to 0.
The RXB is read with an 8-bit memory manipulation instruction. It cannot be written to.
$\overline{\text{RESET}}$ input sets the RXB to FFH.

Caution RXB and the transmit shift register (TXS) are allocated to the same address, and when a write is performed, the value is written to TXS.

(4) Transmission control circuit

This circuit performs transmit operation control such as the addition of a start bit, parity bit and stop bit to data written in the transmit shift register (TXS) in accordance with the contents set in the asynchronous serial interface mode register (ASIM).

(5) Reception control circuit

This circuit controls receive operations in accordance with the contents set in the asynchronous serial interface mode register (ASIM). It performs error checks for parity errors, etc., during a receive operation, and if an error is detected, sets a value in the asynchronous serial interface status register (ASIS) in accordance with the error contents.

19.3 Serial Interface Channel 2 Control Registers

Serial interface channel 2 is controlled by the following five registers.

- Serial Operating Mode Register 2 (CSIM2)
- Asynchronous Serial Interface Mode Register (ASIM)
- Asynchronous Serial Interface Status Register (ASIS)
- Baud Rate Generator Control Register (BRGC)
- Serial Interface Pin Select Register (SIPS)

(1) Serial operating mode register 2 (CSIM2)

This register is set when serial interface channel 2 is used in the 3-wire serial I/O mode.
The CSIM2 is set with a 1-bit or an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the CSIM2 to 00H.

Figure 19-3. Serial Operating Mode Register 2 Format

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
CSIM2	CSIE2	0	0	0	0	CSIM 22	CSCK	0	FF72H	00H	R/W

CSCK	Clock Selection in 3-wire Serial I/O Mode
0	Input clock from off-chip to $\overline{\text{SCK2}}$ pin
1	Dedicated baud rate generator output

CSIM22	First Bit Specification
0	MSB
1	LSB

CSIE2	Operation Control in 3-wire Serial I/O Mode
0	Operation stopped
1	Operation enabled

Cautions 1. Ensure that bits 0 and 3 to 6 are set to 0.

2. When UART mode is selected, CSIM2 should be set to 00H.

(2) Asynchronous serial interface mode register (ASIM)

This register is set when serial interface channel 2 is used in the asynchronous serial interface mode.

ASIM is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIM to 00H.

Figure 19-4. Asynchronous Serial Interface Mode Register Format

Symbol	⑦	⑥	5	4	3	2	1	0	Address	After Reset	R/W
ASIM	TXE	RXE	PS1	PS0	CL	SL	ISRM	SCK	FF70H	00H	R/W

SCK	Clock Selection in Asynchronous Serial Interface Mode
0	Input clock from off-chip to ASCK pin
1	Dedicated baud rate generator output Note

ISRM	Control of Reception Completion Interrupt Request in Case of Error Generation
0	Reception completion interrupt request generated in case of error generation
1	Reception completion interrupt request not generated in case of error generation

SL	Transmit Data Stop Bit Length Specification
0	1 bit
1	2 bits

CL	Character Length Specification
0	7 bits
1	8 bits

PS1	PS0	Parity Bit Specification
0	0	No Parity
0	1	0 parity always added in transmission No parity test in reception (parity error not generated)
1	0	Odd parity
1	1	Even parity

RXE	Receive Operation Control
0	Receive operation stopped
1	Receive operation enabled

TXE	Transmit Operation Control
0	Transmit operation stopped
1	Transmit operation enabled

Note When SCK is set to 1 and the baud rate generator output is selected, the ASCK pin can be used as an input/output port.

Cautions 1. When the 3-wire serial I/O mode is selected, 00H should be set in ASIM.

2. The serial transmit/receive operation must be stopped before changing the operating mode.

Table 19-2. Serial Interface Channel 2 Operating Mode Settings (1/2)

(1) Operation Stop Mode

ASIM			CSIM2			SIPS		PM70	P70	PM71	P71	PM23	P23	PM24	P24	PM72	P72	Start Bit	Shift Clock	P70/SI2/ RxD0 Pin Functions	P71/SO2/ TxD0 Pin Functions	P23/STB/ TxD1 Pin Functions	P24/BUSY/ RxD1 Pin Functions	P72/$\overline{SCK2}$ /ASCK Pin Functions
TXE	RXE	SCK	CSIE2	CSIM22	CSCK	SIPS21	SIPS20																	
0	0	×	0	×	×	×	×	× Note1	× Note1	× Note1	× Note1	× Note1	× Note1	× Note1	× Note1	× Note1	× Note1	—	—	P70	P71	P23/STB	P24/BUSY	P72
Other than above																		Setting prohibited						

(2) 3-wire Serial I/O Mode

ASIM			CSIM2			SIPS		PM70	P70	PM71	P71	PM23	P23	PM24	P24	PM72	P72	Start Bit	Shift Clock	P70/SI2/ RxD0 Pin Functions	P71/SO2/ TxD0 Pin Functions	P23/STB/ TxD1 Pin Functions	P24/BUSY/ RxD1 Pin Functions	P72/$\overline{SCK2}$ /ASCK Pin Functions
TXE	RXE	SCK	CSIE2	CSIM22	CSCK	SIPS21	SIPS20																	
0	0	0	1	0	0	×	×	× Note2	× Note2	0	1	× Note1	× Note1	× Note1	× Note1	1	×	MSB	External clock	SI2 Note2	SO2 (CMOS output)	P23/STB	P24/BUSY	$\overline{SCK2}$ input
					1											0	1		Internal clock					$\overline{SCK2}$ output
			1	1	0											1	×	LSB	External clock	SI2 Note2	SO2 (CMOS output)			$\overline{SCK2}$ input
					1											0	1		Internal clock					$\overline{SCK2}$ output
Other than above																		Setting prohibited						

Notes 1. Can be used freely as port function.

2. Can be used as P70 (CMOS input/output) when only transmission is performed.

Remark × : don't care
PM×× : Port mode register
P×× : Port output latch

Table 19-2. Serial Interface Channel 2 Operating Mode Settings (2/2)

(3) Asynchronous Serial Interface Mode

ASIM			CSIM2			SIPS		PM70	P70	PM71	P71	PM23	P23	PM24	P24	PM72	P72	Start Bit	Shift Clock	P70/SI2/ RxD0 Pin Functions	P71/SO2/ TxD0 Pin Functions	P23/STB/ TxD1 Pin Functions	P24/BUSY/ RxD1 Pin Functions	P72/SCK2 /ASCK Pin Functions
TXE	RXE	SCK	CSIE2	CSIM22	CSCK	SIPS21	SIPS20																	
1	0	0	0	0	0	0	0	×Note	×Note	0	1	×Note	×Note	×Note	×Note	1	×	LSB	External clock	P70	TxD0 (CMOS output)	P23/STB	P24/BUSY	ASCK input
		1														×Note	×Note		Internal clock					P72
0	1	0	0	0	0	0	0	1	×	×Note	×Note	×Note	×Note	×Note	×Note	1	×		External clock	RxD0	P71			ASCK input
		1														×Note	×Note		Internal clock					P72
1	1	0	0	0	0	0	0	1	×	0	1	×Note	×Note	×Note	×Note	1	×		External clock		TxD0 (CMOS output)			ASCK input
		1														×Note	×Note		Internal clock					P72
1	0	0	0	0	0	1	0	×Note	×Note	0	1	0	1	×Note	×Note	1	×		External clock	P70	High output	TxD1	P24/BUSY	ASCK input
		1														×Note	×Note		Internal clock					P72
0	1	0	0	0	0	0	1	1	×	×Note	×Note	×Note	×Note	1	×	1	×		External clock	P70 (Input)	P71	P23/STB	RxD1	ASCK input
		1														×Note	×Note		Internal clock					P72
1	1	0	0	0	0	1	1	1	×	0	1	0	1	1	×	1	×		External clock	P70 (Input)	High output	TxD1	RxD1	ASCK input
		1														×Note	×Note		Internal clock					P72
Other than above																		Setting prohibited						

Note Can be used freely as port function.

Remark × : don’t care
PM×× : Port mode register
P×× : Port output latch

(3) Asynchronous serial interface status register (ASIS)

This is a register which displays the type of error when a reception error is generated in the asynchronous serial interface mode.

ASIS is read with a 1-bit or 8-bit memory manipulation instruction.

In 3-wire serial I/O mode, the contents of ASIS are undefined.

$\overline{\text{RESET}}$ input clears ASIS to 00H.

Figure 19-5. Asynchronous Serial Interface Status Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ASIS	0	0	0	0	0	PE	FE	OVE	FF71H	00H	R

OVE	Overrun Error Flag
0	Overrun error not generated
1	Overrun error generated[Note 1] (When next receive operation is completed before data is read from receive buffer register)

FE	Framing Error Flag
0	Framing error not generated
1	Framing error generated[Note 2] (When stop bit is not detected)

PE	Parity Error Flag
0	Parity error not generated
1	Parity error generated (When transmit data parity does not match)

Notes 1. The receive buffer register (RXB) must be read when an overrun error is generated. Overrun errors will continue to be generated until RXB is read.

2. Even if the stop bit length has been set as 2 bits by bit 2 (SL) of the asynchronous serial interface mode register (ASIM), only single stop bit detection is performed during reception.

(4) Baud rate generator control register (BRGC)

This register sets the serial clock for serial interface channel 2.

BRGC is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears BRGC to 00H.

Figure 19-6. Baud Rate Generator Control Register Format (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
BRGC	TPS3	TPS2	TPS1	TPS0	MDL3	MDL2	MDL1	MDL0	FF73H	00H	R/W

MDL3	MDL2	MDL1	MDL0	Baud Rate Generator Input Clock Selection	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	f_{SCK}Note	—

Note Can only be used in 3-wire serial I/O mode.

Remarks **1.** f_{SCK} : 5-bit counter source clock

2. k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

Figure 19-6. Baud Rate Generator Control Register Format (2/2)

TPS3	TPS2	TPS1	TPS0	5-Bit Counter Source Clock Selection					n
					MCS = 1		MCS = 0		
0	0	0	0	$f_{XX}/2^{10}$	$f_{XX}/2^{10}$	(4.9 kHz)	$f_X/2^{11}$	(2.4 kHz)	11
0	1	0	1	f_{XX}	f_X	(5.0 MHz)	$f_X/2$	(2.5 MHz)	1
0	1	1	0	$f_{XX}/2$	$f_X/2$	(2.5 MHz)	$f_X/2^2$	(1.25 MHz)	2
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$	(1.25 MHz)	$f_X/2^3$	(625 kHz)	3
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$	(625 kHz)	$f_X/2^4$	(313 kHz)	4
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$	(313 kHz)	$f_X/2^5$	(156 kHz)	5
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$	(156 kHz)	$f_X/2^6$	(78.1 kHz)	6
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$	(78.1 kHz)	$f_X/2^7$	(39.1 kHz)	7
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$	(39.1 kHz)	$f_X/2^8$	(19.5 kHz)	8
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$	(19.5 kHz)	$f_X/2^9$	(9.8 kHz)	9
1	1	1	0	$f_{XX}/2^9$	$f_X/2^9$	(9.8 kHz)	$f_X/2^{10}$	(4.9 kHz)	10
Other than above				Setting prohibited					

Caution When a write is performed to BRGC during a communication operation, baud rate generator output is disrupted and communication cannot be performed normally. Therefore, BRGC must not be written to during a communication operation.

Remarks 1. f_X : Main system clock oscillation frequency

2. f_{XX} : Main system clock frequency (f_X or $f_X/2$)

3. MCS : Bit 0 of oscillation mode selection register (OSMS)

4. n : Value set in TPS0 to TPS3 ($1 \leq n \leq 11$)

5. Figures in parentheses apply to operation with f_X = 5.0 MHz

The baud rate transmit/receive clock generated is either a signal scaled from the main system clock, or a signal scaled from the clock input from the ASCK pin.

(a) Generation of baud rate transmit/receive clock by means of main system clock

The transmit/receive clocks generated by scaling the main system clock. The baud rate generated from the main system clock is found from the following expression.

$$[\text{Baud rate}] = \frac{f_{XX}}{2^n \times (k + 16)} \ [\text{Hz}]$$

where, f_X : Main system clock oscillation frequency
f_{XX} : Main system clock frequency (f_X or $f_X/2$)
n : Value set in TPS0 to TPS3 ($1 \leq n \leq 11$)
k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

Table 19-3. Relationships Between Main System Clock and Baud Rate

Baud Rate (bps)	f_X = 5.0 MHz				f_X = 4.19 MHz			
	MCS = 1		MCS = 0		MCS = 1		MCS = 0	
	BRGC Set Value	Error (%)	BRGC Set Value	Error (%)	BRGC Set Value	Error (%)	BRGC Set Value	Error (%)
75	–		00H	1.73	0BH	1.14	EBH	1.14
110	06H	0.88	E6H	0.88	03H	–2.01	E3H	–2.01
150	00H	1.73	E0H	1.73	EBH	1.14	DBH	1.14
300	E0H	1.73	D0H	1.73	DBH	1.14	CBH	1.14
600	D0H	1.73	C0H	1.73	CBH	1.14	BBH	1.14
1,200	C0H	1.73	B0H	1.73	BBH	1.14	ABH	1.14
2,400	B0H	1.73	A0H	1.73	ABH	1.14	9BH	1.14
4,800	A0H	1.73	90H	1.73	9BH	1.14	8BH	1.14
9,600	90H	1.73	80H	1.73	8BH	1.14	7BH	1.14
19,200	80H	1.73	70H	1.73	7BH	1.14	6BH	1.14
31,250	74H	0	64H	0	71H	–1.31	61H	–1.31
38,400	70H	1.73	60H	1.73	6BH	1.14	5BH	1.14
76,800	60H	1.73	50H	1.73	5BH	1.14	—	—

Remark MCS: Bit 0 of oscillation mode selection register (OSMS)

(b) Generation of baud rate transmit/receive clock by means of external clock from ASCK pin

The transmit/receive clock is generated by scaling the clock input from the ASCK pin. The baud rate generated from the clock input from the ASCK pin is obtained with the following expression.

$$[\text{Baud rate}] = \frac{f_{\text{ASCK}}}{2 \times (k + 16)} \ [\text{Hz}]$$

f_{ASCK} : Frequency of clock input to ASCK pin

k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

Table 19-4. Relationships Between ASCK Pin Input Frequency and Baud Rate (When BRGC Is Set to 00H)

Baud Rate (bps)	ASCK Pin Input Frequency
75	2.4 kHz
110	3.52 kHz
150	4.8 kHz
300	9.6 kHz
600	19.2 kHz
1,200	38.4 kHz
2,400	76.8 kHz
4,800	153.6 kHz
9,600	307.2 kHz
19,200	614.4 kHz
31,250	1,000.0 kHz
38,400	1,228.8 kHz

(5) Serial interface pin select register (SIPS)

This register selects input/output pins when the serial interface channel 2 is used in the asynchronous serial interface mode (with time-division transfer function).

SIPS is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears SIPS to 00H.

To select input/output pins, the port mode register and the output latch of the port must be set. For details, refer to **Table 19-2 Serial Interface Channel 2 Operating Mode Settings.**

Figure 19-7. Serial Interface Pin Select Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
SIPS	0	0	SIPS21	SIPS20	0	0	0	0	FF75H	00H	R/W

SIPS21	SIPS20	Selects Input/Output Pin of Asynchronous Serial Interface
0	0	Input pin: RxD0/SI2/P70 Output pin: TxD0/SO2/P71
0	1	Input pin: RxD1/BUSY/P24 Output pin: TxD0/SO2/P71
1	0	Input pin: RxD0/SI2/P70 Output pin: TxD1/STB/P23
1	1	Input pin: RxD1/BUSY/P24 Output pin: TxD1/STB/P23

Cautions 1. Select input/output pins after stopping serial transmission/reception.

2. When using the busy control option or busy & strobe control option in the three-wire serial I/O mode with automatic transmit/receive function of the serial interface channel 1, the RxD1/BUSY/P24 and TxD1/STB/P23 pins cannot be used as data I/O pins.

19.4 Serial Interface Channel 2 Operation

The operating mode of serial interface channel 2 has the following three types.

- Operation stop mode
- Asynchronous serial interface (UART) mode (with time-division transfer function)
- 3-wire serial I/O mode

19.4.1 Operation stop mode

In the operation stop mode, serial transfer is not performed, and therefore power consumption can be reduced.

In the operation stop mode, the P70/SI2/RxD0, P71/SO2/TxD0, and P72/$\overline{\text{SCK2}}$/ASCK pins can be used as normal input/output ports and the P23/STB/TxD1, P24/BUSY/RxD1 pins can be used as normal input/output ports or strobe output and busy input for serial interface automatic transmit/receive.

(1) Register setting

Operation stop mode is set with serial operating mode register 2 (CSIM2) and the asynchronous serial interface mode register (ASIM).

(a) Serial operating mode register 2 (CSIM2)

CSIM2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM2 to 00H.

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
CSIM2	CSIE2	0	0	0	0	CSIM 22	CSCK	0	FF72H	00H	R/W

CSIE2	Operation Control in 3-wire Serial I/O Mode
0	Operation stopped
1	Operation enabled

Caution Ensure that bits 0 and 3 to 6 are set to 0.

(b) Asynchronous serial interface mode register (ASIM)

ASIM is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIM to 00H.

Symbol	⑦	⑥	5	4	3	2	1	0	Address	After Reset	R/W
ASIM	TXE	RXE	PS1	PS0	CL	SL	ISRM	SCK	FF70H	00H	R/W

RXE	Receive Operation Control
0	Receive operation stopped
1	Receive operation enabled

TXE	Transmit Operation Control
0	Transmit operation stopped
1	Transmit operation enabled

19.4.2 Asynchronous serial interface (UART) mode (with time-division transfer function)

In this mode, one byte of data is transmitted/received following the start bit, and full-duplex operation is possible.

A dedicated UART baud rate generator is incorporated, allowing communication over a wide range of baud rates. In addition, the baud rate can be defined by scaling the input clock to the ASCK pin.

The MIDI standard baud rate (31.25 kbps) can be used by employing the dedicated UART baud rate generator.

Two sets of data I/O pins (RxD and TxD) are provided, and the pin to be used can be selected by software (time-division transfer function). However, only one set of pins can be used at one time.

Cautions 1. If it is not necessary to change the data I/O pin, use of the RxD0/SI2/P70 and TxD0/SO2/P71 pins is recommended. If only port 2 (RxD1/BUSY/P24 and TxD1/STB/P23) is used as data I/O pins, the function of port 7 is limited.

2. When using the busy control option or busy & strobe control option in the 3-wire serial I/O mode with automatic transmit/receive function of the serial interface channel 1, the RxD1/BUSY/P24 and TxD1/STB/P23 pins cannot be used as data I/O pins.

(1) Register setting

UART mode (with time-division transfer function) is set with serial operating mode register 2 (CSIM2), the asynchronous serial interface mode register (ASIM), the asynchronous serial interface status register (ASIS), the baud rate generator control register (BRGC), and serial interface pin select register (SIPS).

(a) Serial operating mode register 2 (CSIM2)

CSIM2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM2 to 00H.

When the UART mode is selected, 00H should be set in CSIM2.

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
CSIM2	CSIE2	0	0	0	0	CSIM 22	CSCK	0	FF72H	00H	R/W

CSCK	Clock Selection in 3-wire Serial I/O Mode
0	Input clock from off-chip to $\overline{\text{SCK2}}$ pin
1	Dedicated baud rate generator output

CSIM22	First Bit Specification
0	MSB
1	LSB

CSIE2	Operation Control in 3-wire Serial I/O Mode
0	Operation stopped
1	Operation enabled

Caution Ensure that bits 0 and 3 to 6 are set to 0.

(b) Asynchronous serial interface mode register (ASIM)

ASIM is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIM to 00H.

Symbol	⑦	⑥	5	4	3	2	1	0	Address	After Reset	R/W
ASIM	TXE	RXE	PS1	PS0	CL	SL	ISRM	SCK	FF70H	00H	R/W

SCK	Clock Selection in Asynchronous Serial Interface Mode
0	Input clock from off-chip to ASCK pin
1	Dedicated baud rate generator output Note

ISRM	Control of Reception Completion Interrupt Request in Case of Error Generation
0	Reception completion interrupt request generated in case of error generation
1	Reception completion interrupt request not generated in case of error generation

SL	Transmit Data Stop Bit Length Specification
0	1 bit
1	2 bits

CL	Character Length Specification
0	7 bits
1	8 bits

PS1	PS0	Parity Bit Specification
0	0	No Parity
0	1	0 parity always added in transmission No parity test in reception (parity error not generated)
1	0	Odd parity
1	1	Even parity

RXE	Receive Operation Control
0	Receive operation stopped
1	Receive operation enabled

TXE	Transmit Operation Control
0	Transmit operation stopped
1	Transmit operation enabled

Note When SCK is set to 1 and the baud rate generator output is selected, the ASCK pin can be used as an input/output port.

Caution The serial transmit/receive operation must be stopped before changing the operating mode.

(c) Asynchronous serial interface status register (ASIS)

ASIS is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIS to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ASIS	0	0	0	0	0	PE	FE	OVE	FF71H	00H	R

OVE	Overrun Error Flag
0	Overrun error not generated
1	Overrun error generated Note 1 (When next receive operation is completed before data from receive buffer register is read)

FE	Framing Error Flag
0	Framing error not generated
1	Framing error generated Note 2 (When stop bit is not detected)

PE	Parity Error Flag
0	Parity error not generated
1	Parity error generated (When transmit data parity does not match)

Notes **1.** The receive buffer register (RXB) must be read when an overrun error is generated. Overrun errors will continue to be generated until RXB is read.

2. Even if the stop bit length has been set as 2 bits by bit 2 (SL) of the asynchronous serial interface mode register (ASIM), only single stop bit detection is performed during reception.

(d) Baud rate generator control register (BRGC)

BRGC is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears BRGC to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
BRGC	TPS3	TPS2	TPS1	TPS0	MDL3	MDL2	MDL1	MDL0	FF73H	00H	R/W

MDL3	MDL2	MDL1	MDL0	Baud Rate Generator Input Clock Selection	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14

(Cont'd)

Remark f_{SCK} : 5-bit counter source clock
k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

TPS3	TPS2	TPS1	TPS0	5-Bit Counter Source Clock Selection					n
					MCS = 1		MCS = 0		
0	0	0	0	$f_{XX}/2^{10}$	$f_X/2^{10}$	(4.9 kHz)	$f_X/2^{11}$	(2.4 kHz)	11
0	1	0	1	f_{XX}	f_X	(5.0 MHz)	$f_X/2$	(2.5 MHz)	1
0	1	1	0	$f_{XX}/2$	$f_X/2$	(2.5 MHz)	$f_X/2^2$	(1.25 MHz)	2
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$	(1.25 MHz)	$f_X/2^3$	(625 kHz)	3
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$	(625 kHz)	$f_X/2^4$	(313 kHz)	4
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$	(313 kHz)	$f_X/2^5$	(156 kHz)	5
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$	(156 kHz)	$f_X/2^6$	(78.1 kHz)	6
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$	(78.1 kHz)	$f_X/2^7$	(39.1 kHz)	7
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$	(39.1 kHz)	$f_X/2^8$	(19.5 kHz)	8
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$	(19.5 kHz)	$f_X/2^9$	(9.8 kHz)	9
1	1	1	0	$f_{XX}/2^9$	$f_X/2^9$	(9.8 kHz)	$f_X/2^{10}$	(4.9 kHz)	10
Other than above				Setting prohibited					

Caution When a write is performed to BRGC during a communication operation, baud rate generator output is disrupted and communication cannot be performed normally. Therefore, BRGC must not be written to during a communication operation.

Remarks 1. f_X : Main system clock oscillation frequency
2. f_{XX} : Main system clock frequency (f_X or $f_X/2$)
3. MCS : Bit 0 of oscillation mode selection register (OSMS)
4. n : Value set in TPS0 to TPS3 ($1 \leq n \leq 11$)
5. Figures in parentheses apply to operation with f_X = 5.0 MHz.

The baud rate transmit/receive clock generated is either a signal scaled from the main system clock, or a signal scaled from the clock input from the ASCK pin.

(i) Generation of baud rate transmit/receive clock by means of main system clock

The transmit/receive clock is generated by scaling the main system clock. The baud rate generated from the main system clock is obtained with the following expression.

$$[\text{Baud rate}] = \frac{f_{XX}}{2^n \times (k + 16)} \text{ [Hz]}$$

where, f_X : Main system clock oscillation frequency
f_{XX} : Main system clock frequency (f_X or $f_X/2$)
n : Value set in TPS0 to TPS3 ($1 \le n \le 11$)
k : Value set in MDL0 to MDL3 ($0 \le k \le 14$)

Table 19-5. Relationships Between Main System Clock and Baud Rate

Baud Rate (bps)	fx = 5.0 MHz				fx = 4.19 MHz			
	MCS = 1		MCS = 0		MCS = 1		MCS = 0	
	BRGC Set Value	Error (%)	BRGC Set Value	Error (%)	BRGC Set Value	Error (%)	BRGC Set Value	Error (%)
75	–		00H	1.73	0BH	1.14	EBH	1.14
110	06H	0.88	E6H	0.88	03H	–2.01	E3H	–2.01
150	00H	1.73	E0H	1.73	EBH	1.14	DBH	1.14
300	E0H	1.73	D0H	1.73	DBH	1.14	CBH	1.14
600	D0H	1.73	C0H	1.73	CBH	1.14	BBH	1.14
1,200	C0H	1.73	B0H	1.73	BBH	1.14	ABH	1.14
2,400	B0H	1.73	A0H	1.73	ABH	1.14	9BH	1.14
4,800	A0H	1.73	90H	1.73	9BH	1.14	8BH	1.14
9,600	90H	1.73	80H	1.73	8BH	1.14	7BH	1.14
19,200	80H	1.73	70H	1.73	7BH	1.14	6BH	1.14
31,250	74H	0	64H	0	71H	–1.31	61H	–1.31
38,400	70H	1.73	60H	1.73	6BH	1.14	5BH	1.14
76,800	60H	1.73	50H	1.73	5BH	1.14	—	—

Remark MCS: Bit 0 of oscillation mode selection register (OSMS)

(ii) Generation of baud rate transmit/receive clock by means of external clock from ASCK pin

The transmit/receive clock is generated by scaling the clock input from the ASCK pin. The baud rate generated from the clock input from the ASCK pin is obtained with the following expression.

$$[\text{Baud rate}] = \frac{f_{\text{ASCK}}}{2 \times (k + 16)} \text{ [Hz]}$$

where, f_{ASCK} : Frequency of clock input to ASCK pin

k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

Table 19-6. Relationships Between ASCK Pin Input Frequency and Baud Rate (When BRGC Is Set to 00H)

Baud Rate (bps)	ASCK Pin Input Frequency
75	2.4 kHz
110	3.52 kHz
150	4.8 kHz
300	9.6 kHz
600	19.2 kHz
1,200	38.4 kHz
2,400	76.8 kHz
4,800	153.6 kHz
9,600	307.2 kHz
19,200	614.4 kHz
31,250	1,000.0 kHz
38,400	1,228.8 kHz

(e) Serial interface pin select register (SIPS)

SIPS is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears SIPS to 00H.

To select input/output pins, the port mode register and the output latch of the port must be set. For details, refer to **Table 19-2 Serial Interface Channel 2 Operating Mode Settings**.

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
SIPS	0	0	SIPS21	SIPS20	0	0	0	0	FF75H	00H	R/W

SIPS21	SIPS20	Selects Input/Output Pin of Asynchronous Serial Interface
0	0	Input pin: RxD0/SI2/P70 Output pin: TxD0/SO2/P71
0	1	Input pin: RxD1/BUSY/P24 Output pin: TxD0/SO2/P71
1	0	Input pin: RxD0/SI2/P70 Output pin: TxD1/STB/P23
1	1	Input pin: RxD1/BUSY/P24 Output pin: TxD1/STB/P23

Cautions 1. Select input/output pins after stopping serial transmission/reception.

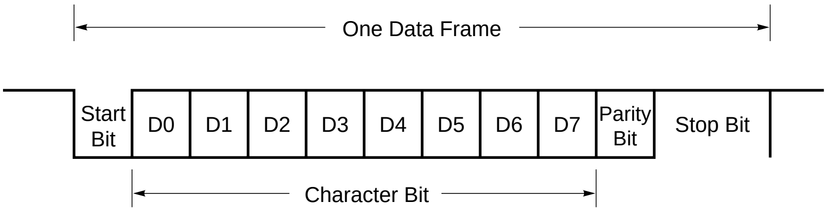
2. When using the busy control option or busy & strobe control option in the three-wire serial I/O mode with automatic transmit/receive function of the serial interface channel 1, the RxD1/BUSY/P24 and TxD1/STB/P23 pins cannot be used as data I/O pins.

(2) Communication operation

(a) Data format

The transmit/receive data format is as shown in Figure 19-8.

Figure 19-8. Asynchronous Serial Interface Transmit/Receive Data Format



One data frame consists of the following bits:

• Start bits	1 bit
• Character bits	7 bits/8 bits
• Parity bits	Even parity/odd parity/0 parity/no parity
• Stop bit(s)	1 bit/2 bits

The specification of character bit length, parity selection, and specification of stop bit length for each data frame is carried out with the asynchronous serial interface mode register (ASIM).

When 7 bits are selected as the number of character bits, only the lower 7 bits (bits 0 to 6) are valid; in transmission the most significant bit (bit 7) is ignored, and in reception the most significant bit (bit 7) is always "0".

The serial transfer rate is selected by means of ASIM and the baud rate generator control register (BRGC).

If a serial data receive error is generated, the receive error contents can be determined by reading the status of the asynchronous serial interface status register (ASIS).

(b) Parity types and operation

The parity bit is used to detect a bit error in the communication data. Normally, the same kind of parity bit is used on the transmitting side and the receiving side. With even parity and odd parity, a one-bit (odd number) error can be detected. With 0 parity and no parity, an error cannot be detected.

(i) Even parity

- **Transmission**

  The number of bits with a value of “1”, including the parity bit, in the transmit data is controlled to be even.
  The value of the parity bit is as follows:

  Number of bits with a value of “1” in transmit data is odd: 1
  Number of bits with a value of “1” in transmit data is even: 0

- **Reception**

  The number of bits with a value of “1”, including the parity bit, in the receive data is counted. If it is odd, a parity error occurs.

(ii) Odd parity

- **Transmission**

  Conversely to the situation with even parity, the number of bits with a value of “1”, including the parity bit, in the transmit data is controlled to be odd. The value of the parity bit is as follows:

  Number of bits with a value of “1” in transmit data is odd: 0
  Number of bits with a value of “1” in transmit data is even: 1

- **Reception**

  The number of bits with a value of “1”, including the parity bit, in the receive data is counted. If it is even, a parity error occurs.

(iii) 0 Parity

When transmitting, the parity bit is set to "0" irrespective of the transmit data.
At reception, a parity bit check is not performed. Therefore, a parity error is not generated, irrespective of whether the parity bit is set to "0" or "1".

(iv) No parity

A parity bit is not added to the transmit data. At reception, data is received assuming that there is no parity bit. Since there is no parity bit, a parity error is not generated.

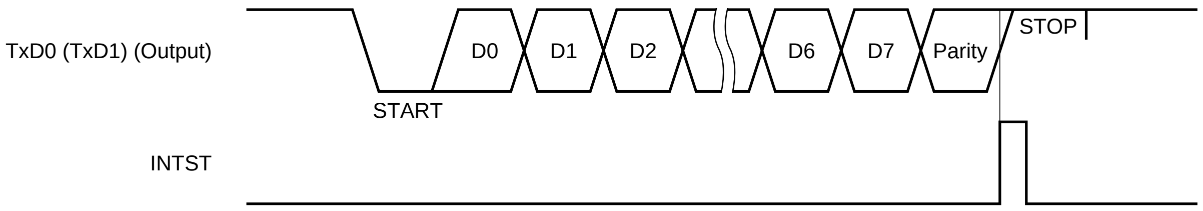
(c) Transmission

A transmit operation is started by writing transmit data to the transmit shift register (TXS). The start bit, parity bit and stop bit(s) are added automatically.

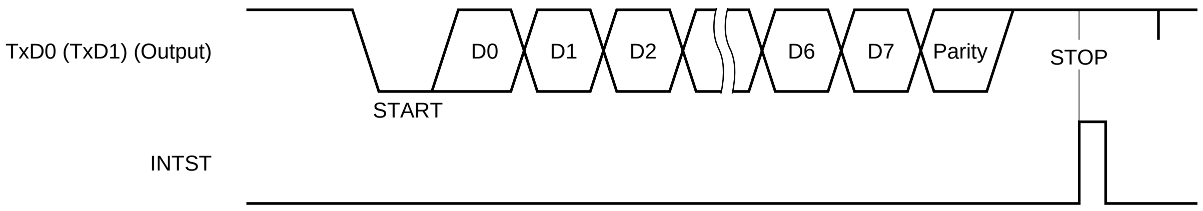
When the transmit operation starts, the data in the transmit shift register (TXS) is shifted out, and when the transmit shift register (TXS) is empty, a transmission completion interrupt request (INTST) is generated.

Figure 19-9. Asynchronous Serial Interface Transmission Completion Interrupt Request Generation Timing

(a) Stop bit length: 1



(b) Stop bit length: 2



Caution Rewriting of the asynchronous serial interface mode register (ASIM) should not be performed during a transmit operation. If rewriting of the ASIM is performed during transmission, subsequent transmit operations may not be possible (the normal state is restored by $\overline{\text{RESET}}$ input).

It is possible to determine whether transmission is in progress by software by using a transmission completion interrupt request (INTST) or the interrupt request flag (STIF) set by the INTST.

(d) Reception

When the RXE bit of the asynchronous serial interface mode register (ASIM) is set (to 1), a receive operation is enabled and sampling of the RxD0 (RxD1) pin input is started.

RxD0 (RxD1) pin input sampling is performed using the serial clock specified by ASIM.

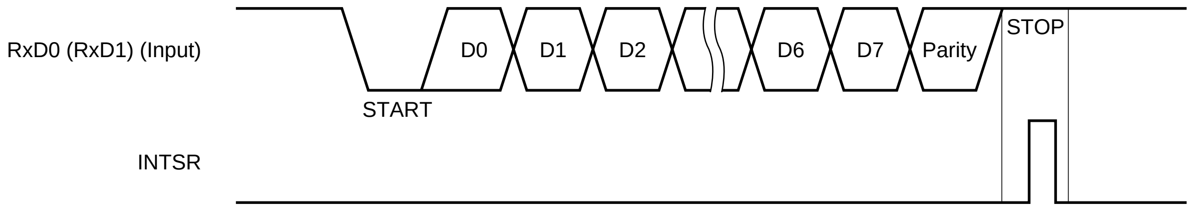
When the RxD0 (RxD1) pin input becomes low, the 5-bit counter of the baud rate generator (see **Figure 19-2**) starts counting, and at the time when the half time determined by specified baud rate has passed, the data sampling start timing signal is output. If the RxD0 (RxD1) pin input sampled again as a result of this start timing signal is low, it is identified as a start bit, the 5-bit counter is initialized and starts counting, and data sampling is performed. When character data, a parity bit and one stop bit are detected after the start bit, reception of one frame of data ends.

When one frame of data has been received, the receive data in the shift register is transferred to the receive buffer register (RXB), and a reception completion interrupt request (INTSR) is generated.

If an error is generated, the receive data in which the error was generated is still transferred to RXB. If bit 1 (ISRM) of ASIM is cleared (to 0) on occurrence of the error, INTSR is generated.

If the RXE bit is reset (to 0) during the receive operation, the receive operation is stopped immediately. In this case, the contents of RXB and the asynchronous serial interface status register (ASIS) are not changed, and INTSR and INTSER are not generated.

Figure 19-10. Asynchronous Serial Interface Reception Completion Interrupt Request Generation Timing



Caution The receive buffer register (RXB) must be read even if a receive error is generated. If RXB is not read, an overrun error will be generated when the next data is received, and the receive error state will continue indefinitely.

(e) Receive errors

Three kinds of errors can occur during a receive operation: a parity error, framing error, or overrun error. The data reception result error flag is set in the asynchronous serial interface status register (ASIS) and a receive error interrupt request (INTSER) is generated. The receive error interrupt is generated faster than receive completion interrupt (INTSR). Receive error causes are shown in Table 19-7.

It is possible to determine what kind of error was generated during reception by reading the contents of ASIS in the reception error interrupt servicing (see **Figures 19-10** and **19-11**).

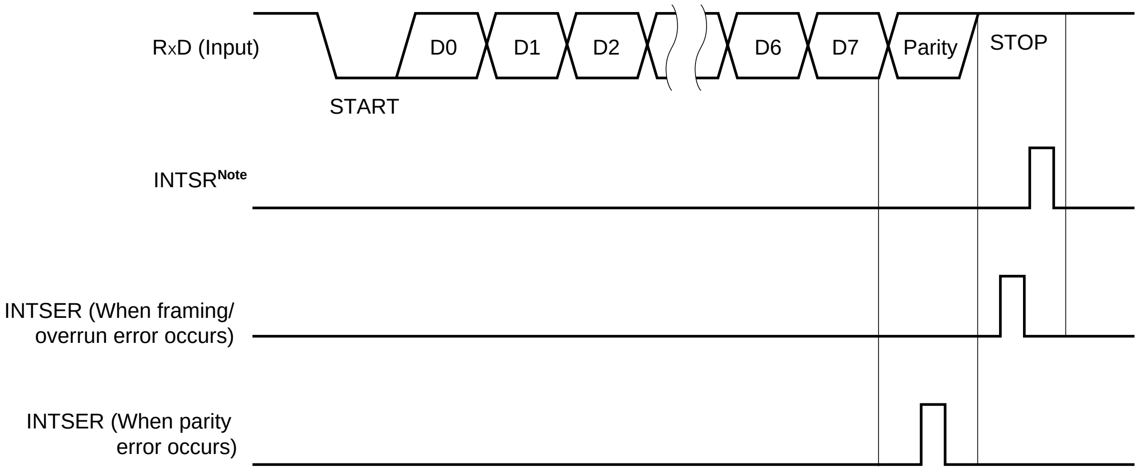
The contents of ASIS are reset (to 0) by reading the receive buffer register (RXB) or receiving the next data (if there is an error in the next data, the corresponding error flag is set).

Table 19-7. Receive Error Causes

Receive Errors	Cause
Parity error	Transmission-time parity specification and reception data parity do not match
Framing error	Stop bit not detected
Overrun error	Reception of next data is completed before data is read from receive register buffer

★

Figure 19-11. Receive Error Timing


RxD (Input)
D0
D1
D2
D6
D7
Parity
STOP
START
INTSRNote
INTSER (When framing/ overrun error occurs)
INTSER (When parity error occurs)

Note INTSR is not generated if a receive error is generated while bit 1 (ISRM) of the asynchronous serial interface mode register (ASIM) is set (to 1).

Cautions 1. The contents of the asynchronous serial interface status register (ASIS) are reset (to 0) by reading the receive buffer register (RXB) or receiving the next data. To ascertain the error contents, ASIS must be read before reading RXB.

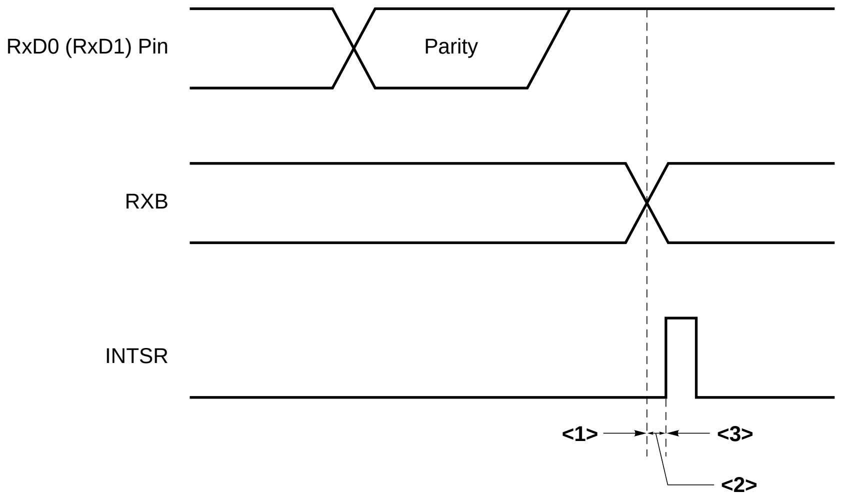
2. The receive buffer register (RXB) must be read even if a receive error is generated. If RXB is not read, an overrun error will be generated when the next data is received, and the receive error state will continue indefinitely.

(3) UART mode cautions

(a) When the transmission under execution has been stopped by clearing bit 7 (TXE) of the asynchronous serial interface mode register (ASIM) to 0, be sure to set the transmit shift register (TXS) to FFH, then set the TXE to 1 before executing the next transmission.

(b) When the reception under execution has been stopped by clearing bit 6 (RXE) of the asynchronous serial interface mode register (ASIM) to 0, the status of receive buffer register (RXB) and whether the receive completion interrupt request (INTSR) occurs differ depending on the timing with which reception is stopped.
Figure 19-12 shows the timing.

Figure 19-12. Status of Receive Buffer Register (RXB) and Generation of Interrupt Request (INTSR) when Reception Is Stopped



When RXE is set to 0 at a time indicated by **<1>**, RXB holds the previous data and does not generate INTSR.
When RXE is set to 0 at a time indicated by **<2>**, RXB renews the data and does not generate INTSR.
When RXE is set to 0 at a time indicated by **<3>**, RXB renews the data and generates INTSR.

19.4.3 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connection of peripheral I/Os and display controllers, etc., which incorporate a conventional synchronous clocked serial interface, such as the 75X/XL Series, 78K Series, 17K Series, etc.

Communication is performed using three lines: the serial clock ($\overline{\text{SCK2}}$), serial output (SO2), and serial input (SI2).

In the 3-wire serial I/O mode, the P23/STB/TxD1, P24/BUSY/RxD1 pins can be used as normal input/output ports.

(1) Register setting

3-wire serial I/O mode is set with serial operating mode register 2 (CSIM2), the asynchronous serial interface mode register (ASIM), and the baud rate generator control register (BRGC).

(a) Serial operating mode register 2 (CSIM2)

CSIM2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM2 to 00H.

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
CSIM2	CSIE2	0	0	0	0	CSIM 22	CSCK	0	FF72H	00H	R/W

CSCK	Clock Selection in 3-wire Serial I/O Mode
0	Input clock from off-chip to $\overline{\text{SCK2}}$ pin
1	Dedicated baud rate generator output

CSIM22	First Bit Specification
0	MSB
1	LSB

CSIE2	Operation Control in 3-wire Serial I/O Mode
0	Operation stopped
1	Operation enabled

Caution Ensure that bits 0 and 3 to 6 are set to 0.

(b) Asynchronous serial interface mode register (ASIM)

ASIM is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIM to 00H.

When the 3-wire serial I/O mode is selected, 00H should be set in ASIM.

Symbol	⑦	⑥	5	4	3	2	1	0	Address	After Reset	R/W
ASIM	TXE	RXE	PS1	PS0	CL	SL	ISRM	SCK	FF70H	00H	R/W

SCK	Clock Selection in Asynchronous Serial Interface Mode
0	Input clock from off-chip to ASCK pin
1	Dedicated baud rate generator output

ISRM	Control of Reception Completion Interrupt Request in Case of Error Generation
0	Reception completion interrupt request generated in case of error generation
1	Reception completion interrupt request not generated in case of error generation

SL	Transmit Data Stop Bit Length Specification
0	1 bit
1	2 bits

CL	Character Length Specification
0	7 bits
1	8 bits

PS1	PS0	Parity Bit Specification
0	0	No Parity
0	1	0 parity always added in transmission No parity test in reception (parity error not generated)
1	0	Odd parity
1	1	Even parity

RXE	Receive Operation Control
0	Receive operation stopped
1	Receive operation enabled

TXE	Transmit Operation Control
0	Transmit operation stopped
1	Transmit operation enabled

(c) Baud rate generator control register (BRGC)

BRGC is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears BRGC to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
BRGC	TPS3	TPS2	TPS1	TPS0	MDL3	MDL2	MDL1	MDL0	FF73H	00H	R/W

MDL3	MDL2	MDL1	MDL0	Baud Rate Generator Input Clock Selection	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	f_{SCK}	—

(Cont'd)

Remark f_{SCK} : 5-bit counter source clock

k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

TPS3	TPS2	TPS1	TPS0	5-Bit Counter Source Clock Selection					n
					MCS = 1		MCS = 0		
0	0	0	0	$f_{XX}/2^{10}$	$f_X/2^{10}$	(4.9 kHz)	$f_X/2^{11}$	(2.4 kHz)	11
0	1	0	1	f_{XX}	f_X	(5.0 MHz)	$f_X/2$	(2.5 MHz)	1
0	1	1	0	$f_{XX}/2$	$f_X/2$	(2.5 MHz)	$f_X/2^2$	(1.25 MHz)	2
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$	(1.25 MHz)	$f_X/2^3$	(625 kHz)	3
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$	(625 kHz)	$f_X/2^4$	(313 kHz)	4
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$	(313 kHz)	$f_X/2^5$	(156 kHz)	5
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$	(156 kHz)	$f_X/2^6$	(78.1 kHz)	6
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$	(78.1 kHz)	$f_X/2^7$	(39.1 kHz)	7
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$	(39.1 kHz)	$f_X/2^8$	(19.5 kHz)	8
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$	(19.5 kHz)	$f_X/2^9$	(9.8 kHz)	9
1	1	1	0	$f_{XX}/2^9$	$f_X/2^9$	(9.8 kHz)	$f_X/2^{10}$	(4.9 kHz)	10
Other than above				Setting prohibited					

Caution **When a write is performed to BRGC during a communication operation, baud rate generator output is disrupted and communication cannot be performed normally. Therefore, BRGC must not be written to during a communication operation.**

Remarks **1.** f_X : Main system clock oscillation frequency

2. f_{XX} : Main system clock frequency (f_X or $f_X/2$)

3. MCS : Bit 0 of oscillation mode selection register (OSMS)

4. n : Value set in TPS0 to TPS3 ($1 \leq n \leq 11$)

5. Figures in parentheses apply to operation with f_X = 5.0 MHz.

When the internal clock is used as the serial clock in the 3-wire serial I/O mode, set BRGC as described below. BRGC setting is not required if an external serial clock is used.

(i) When the baud rate generator is not used:

Select a serial clock frequency with TPS0 to TPS3. Be sure then to set MDL0 to MDL3 to 1,1,1,1. The serial clock frequency becomes the same as the source clock frequency for the 5-bit counter.

(ii) When the baud rate generator is used:

Select a serial clock frequency with TPS0 to TPS3. Be sure then to set MDL0 to MDL3 to 1,1,1,1.

The serial clock frequency is calculated by the following formula:

$$\text{Serial clock frequency} = \frac{f_{XX}}{2^n \text{ x } (k + 16)} \text{ [Hz]}$$

Remarks 1. f_X : Main system clock oscillation frequency

2. f_{XX} : Main system clock frequency (f_X or $f_X/2$)

3. n : Value set in TPS0 to TPS3 ($1 \leq n \leq 11$)

4. k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

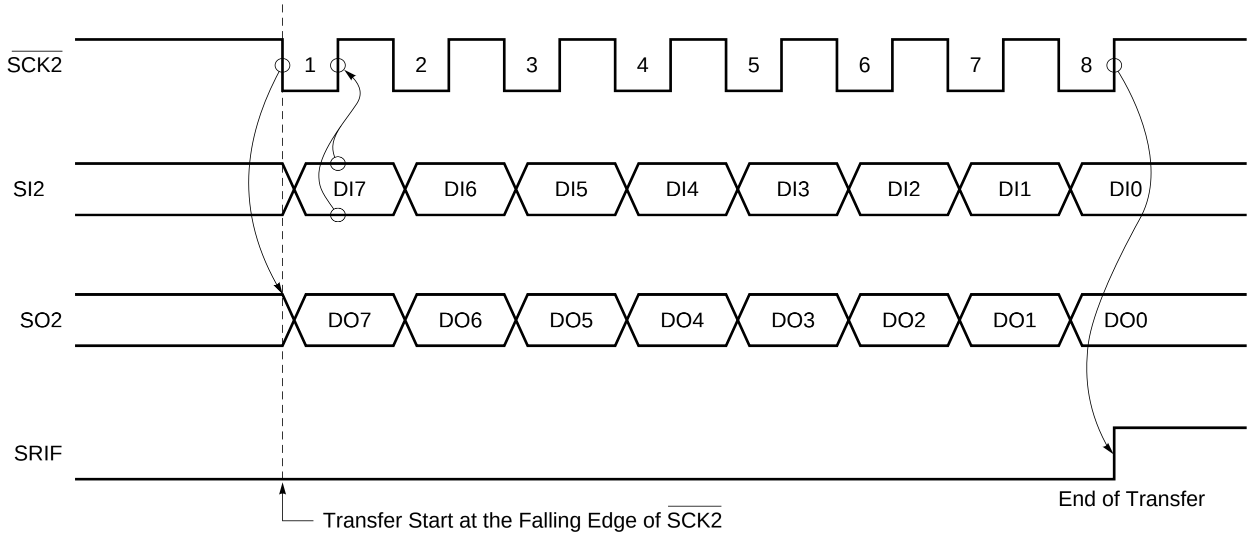
(2) Communication operation

In the 3-wire serial I/O mode, data transmission/reception is performed in 8-bit units. Data is transmitted/ received bit by bit in synchronization with the serial clock.

Transmit shift register (TXS/SIO2) and receive shift register (RXS) shift operations are performed in synchronization with the fall of the serial clock $\overline{\text{SCK2}}$. Then transmit data is held in the SO2 latch and output from the SO2 pin. Also, receive data input to the SI2 pin is latched in the receive buffer register (RXB/SIO2) on the rise of $\overline{\text{SCK2}}$.

At the end of an 8-bit transfer, the operation of the TXS/SIO2 or RXS stops automatically, and the interrupt request flag (SRIF) is set.

Figure 19-13. 3-wire Serial I/O Mode Timing



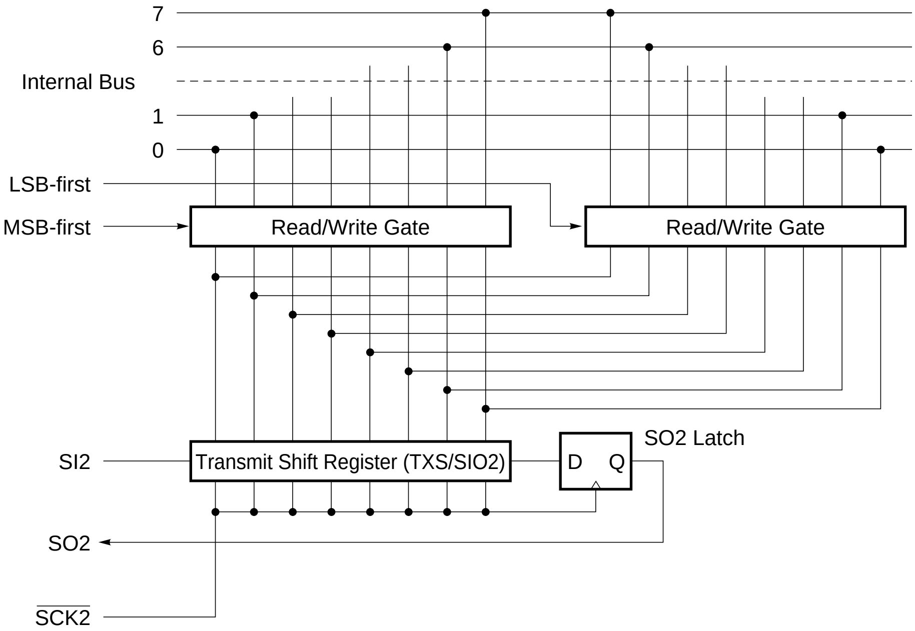
★ **(3) MSB/LSB switching as the start bit**

The 3-wire serial I/O mode enables to select transfer to start from MSB or LSB.

Figure 19-14 shows the configuration of the transmit shift register (TXS/SIO2) and internal bus. As shown in the figure, MSB/LSB can be read/written in reverse form.

MSB/LSB switching as the start bit can be specified with bit 2 (CSIM22) of the serial operating mode register 2 (CSIM2).

Figure 19-14. Circuit of Switching in Transfer Bit Order



Start bit switching is realized by switching the bit order for data write to SIO2. The SIO2 shift order remains unchanged.

Thus, switching between MSB-first and LSB-first must be performed before writing data to the shift register.

(4) Transfer start

Serial transfer is started by setting transfer data to the transmission shift register (TXS/SIO2) when the following two conditions are satisfied.

- Serial interface channel 2 operation control bit (CSIE2) = 1
- Internal serial clock is stopped or $\overline{\text{SCK2}}$ is a high level after 8-bit serial transfer.

Caution   If CSIE2 is set to "1" after data write to TXS/SIO2, transfer does not start.

Remark   CSIE2: Bit 7 of serial operating mode register 2 (CSIM2)

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (SRIF) is set.

★

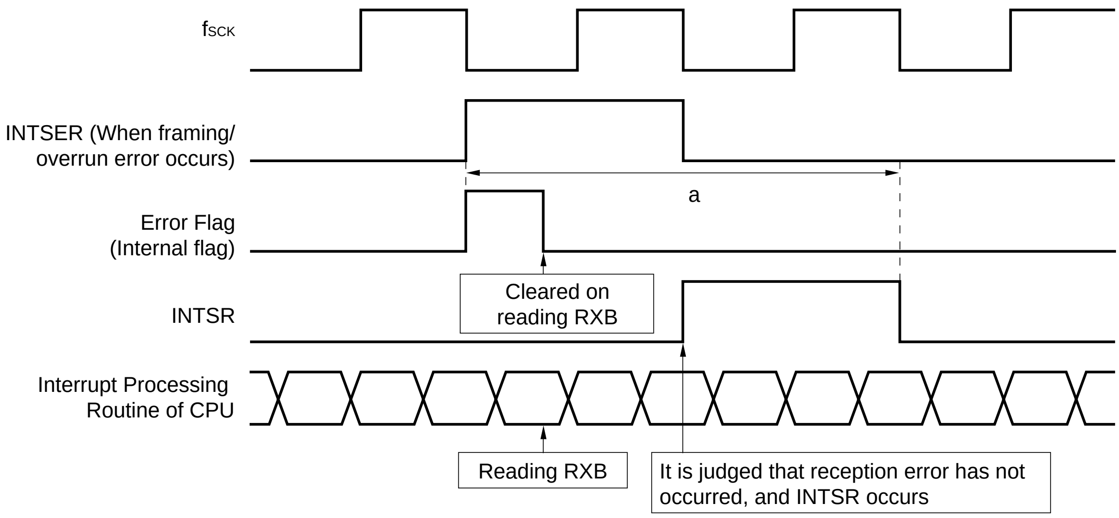
19.4.4 Restrictions in UART mode

In the UART mode, the reception completion interrupt request (INTSR) occurs a certain time after the reception error interrupt (INTSER) has occurred and then cleared. Consequently, the following phenomenon may occur.

- **Description**

  If bit 1 (ISRM) of the asynchronous serial interface mode register (ASIM) is set to 1, the reception completion interrupt request (INTSR) does not occur on occurrence of a reception error. If the receive buffer register (RXB) is read at certain timing ("a" in Figure 19-15) during the reception error interrupt (INTSER) processing, the internal error flag is cleared to 0. As a result, it is judged that no reception error has occurred, and INTSR, which must not occur, occurs. Figure 19-15 illustrates this operation.

Figure 19-15. Reception Completion Interrupt Request Generation Timing (When ISRM = 1)


f_{SCK}
INTSER (When framing/ overrun error occurs)
a
Error Flag (Internal flag)
Cleared on reading RXB
INTSR
Interrupt Processing Routine of CPU
Reading RXB
It is judged that reception error has not occurred, and INTSR occurs

Remark ISRM : Bit 1 of asynchronous serial interface mode register (ASIM)
f_{SCK} : Source clock of 5-bit counter of baud rate generator
RXB : Receive buffer register

To avoid this phenomenon, take the following measures:

- **Preventive measures**

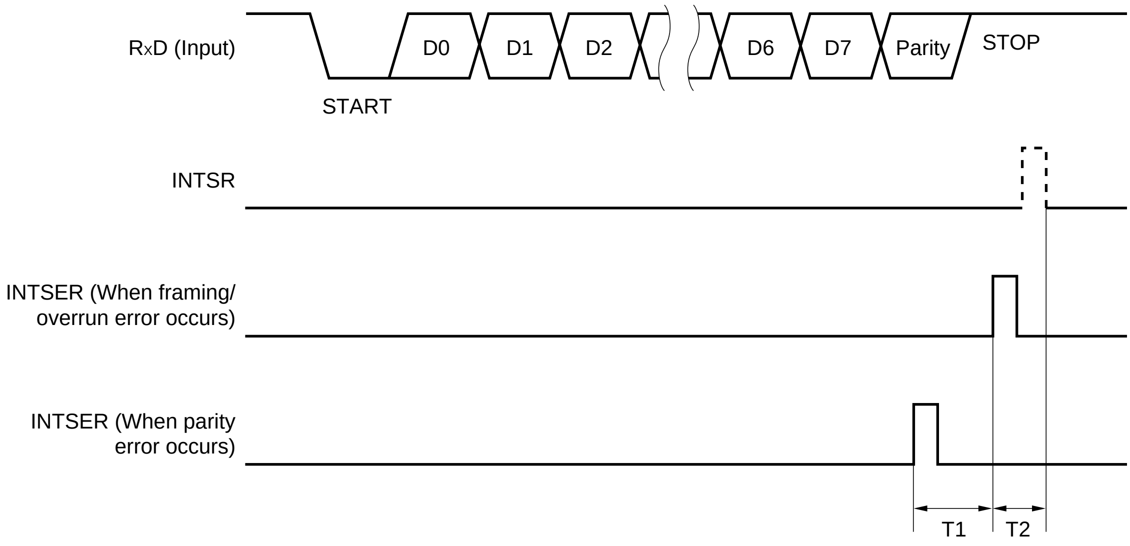
  - **In case of framing error or overrun error**

    Disable the receive buffer register (RXB) from being read for a certain period (T2 in Figure 19-16) after the reception error interrupt request (INTSER) has occurred.

  - **In case of parity error**

    Disable the receive buffer register (RXB) from being read for a certain period (T1 + T2 in Figure 19-16) after the reception error interrupt request (INTSER) has occurred.

Figure 19-16. Receive Buffer Register Read Disable Period


RxD (Input)
START
D0
D1
D2
D6
D7
Parity
STOP
INTSR
INTSER (When framing/ overrun error occurs)
INTSER (When parity error occurs)
T1
T2

T1 : Time of one data of baud rate selected by baud rate generator control register (BRGC) (1/baud rate)

T2 : Time of 2 clocks of source clock (f_{SCK}) of 5-bit counter selected by BRGC

- **Example of preventive measures**

Here is an example of the above preventive measures.

[Condition]

f_X = 5.0 MHz

Processor clock control register (PCC) = 00H

Oscillation mode select register (OSMS) = 01H

Baud rate generator control register (BRGC) = B0H (2,400 bps selected as baud rate)

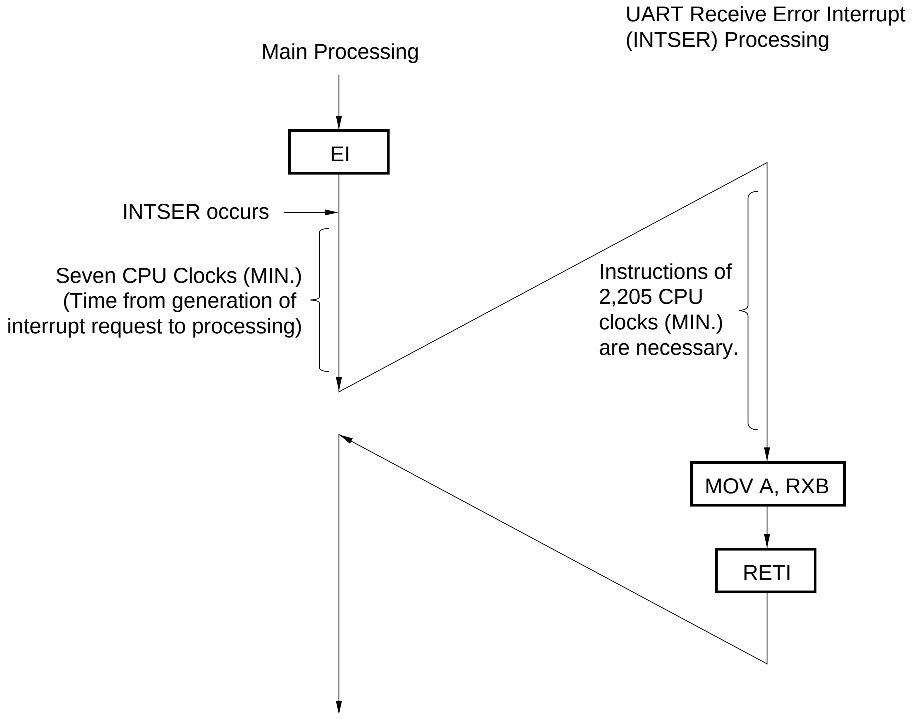
$T_{CY} = 0.4\ \mu s\ (t_{CY} = 0.2\ \mu s)$

$$T1 = \frac{1}{2,400} = 416.7\ \mu s$$

$$T2 = 12.8 \times 2 = 25.6\ \mu s$$

$$\frac{T1 + T2}{t_{CY}} = 2,212 \text{ (clocks)}$$

[Example]


Main Processing
UART Receive Error Interrupt (INTSER) Processing
EI
INTSER occurs
Seven CPU Clocks (MIN.) (Time from generation of interrupt request to processing)
Instructions of 2,205 CPU clocks (MIN.) are necessary.
MOV A, RXB
RETI

[MEMO]

CHAPTER 20 REAL-TIME OUTPUT PORT

20.1 Real-Time Output Port Functions

Data set previously in the real-time output buffer register can be transferred to the output latch by hardware concurrently with the generation of a timer interrupt request or external interrupt request, then output externally. This is called the real-time output function. The pins that output data externally are called real-time output ports.

By using a real-time output, a signal which has no jitter can be output. This port is therefore suitable for control of stepping motors, etc.

Port mode/real-time output port mode can be specified bit-wise.

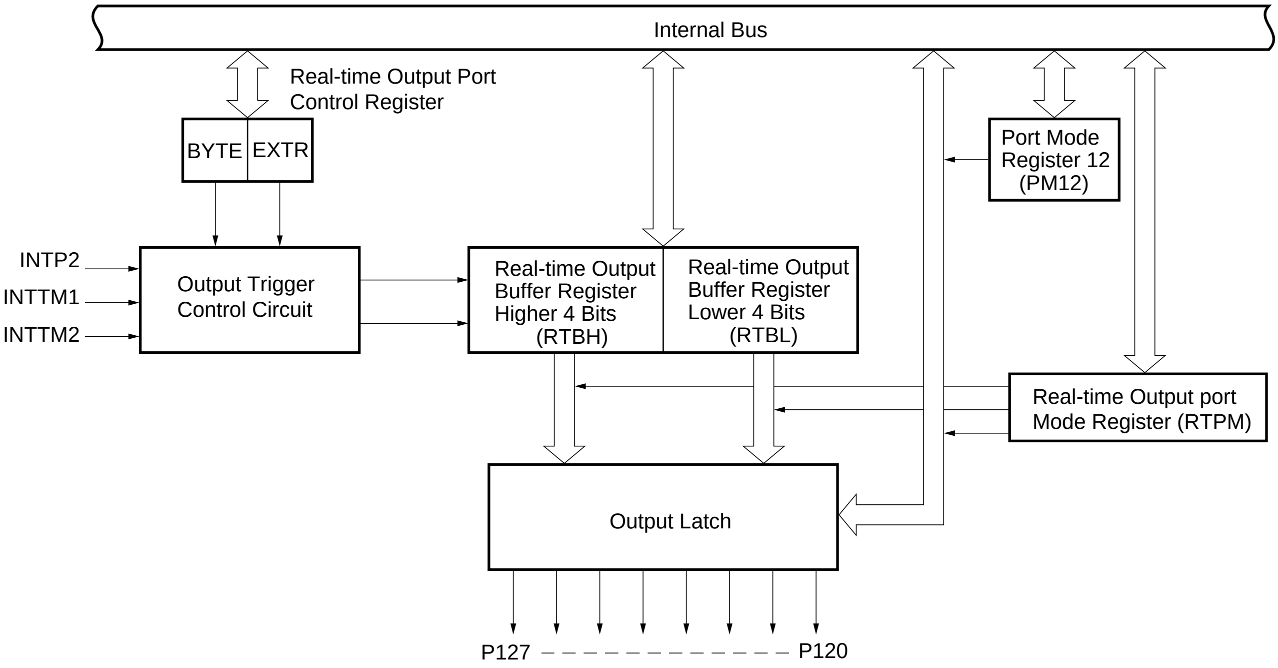
20.2 Real-Time Output Port Configuration

The real-time output port consists of the following hardware.

Table 20-1. Real-time Output Port Configuration

Item	Configuration
Register	Real-time output buffer register (RTBL, RTBH)
Control register	Port mode register 12 (PM12) Real-time output port mode register (RTPM) Real-time output port control register (RTPC)

Figure 20-1. Real-time Output Port Block Diagram


Internal Bus
Real-time Output Port Control Register
BYTE
EXTR
Port Mode Register 12 (PM12)
INTP2
INTTM1
INTTM2
Output Trigger Control Circuit
Real-time Output Buffer Register Higher 4 Bits (RTBH)
Real-time Output Buffer Register Lower 4 Bits (RTBL)
Real-time Output port Mode Register (RTPM)
Output Latch
P127
P120

(1) Real-time output buffer register (RTBL, RTBH)

Addresses of RTBL and RTBH are mapped individually in the Special function register (SFR) area as shown in Figure 20-2.

When specifying 4 bits × 2 channels as the operating mode, data are set individually in RTBL and RTBH. When specifying 8 bits × 1 channel as the operating mode, data are set to both RTBL and RTBH by writing 8-bit data to either RTBL or RTBH.

Table 20-2 shows operations during manipulation of RTBL and RTBH.

Figure 20-2. Real-time Output Buffer Register Configuration

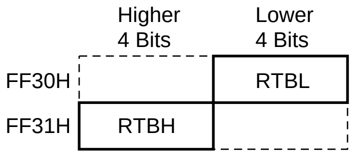


Table 20-2. Operation in Real-time Output Buffer Register Manipulation

Operating Mode	Register to be Manipulated	In Read Note 1		In Write Note 2	
		Higher 4 Bits	Lower 4 Bits	Higher 4 Bits	Lower 4 Bits
4 Bits × 2 Channels	RTBL	RTBH	RTBL	Invalid	RTBL
	RTBH	RTBH	RTBL	RTBH	Invalid
8 Bits × 1 Channel	RTBL	RTBH	RTBL	RTBH	RTBL
	RTBH	RTBH	RTBL	RTBH	RTBL

Notes 1. Only the bits set in the real-time output port mode can be read. When a bit set in the port mode is read, 0 is read.

2. After setting data in the real-time output port, output data should be set in RTBL and RTBH by the time a real-time output trigger is generated.

20.3 Real-Time Output Port Control Registers

The following three registers control the real-time output port.

- Port mode register 12 (PM12)
- Real-time output port mode register (RTPM)
- Real-time output port control register (RTPC)

(1) Port mode register 12 (PM12)

This register sets the input or output mode of port 12 pins (P120 to P127) which are multiplexed with real-time output pins (RTP0 to RTP7). To use port 12 as a real-time output port, the port pin that performs real-time output must be set in the output mode (PM12n = 0: n = 0 to 7).

PM12 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM12 to FFH.

Figure 20-3. Port Mode Register 12 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
PM12	PM127	PM126	PM125	PM124	PM123	PM122	PM121	PM120	FF2CH	FFH	R/W

PM12n	Selects I/O Mode of P12n Pin (n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (ourput buffer OFF)

(2) Real-time output port mode register (RTPM)

This register selects the real-time output port mode/port mode bit-wise.

RTPM is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears RTPM to 00H.

Figure 20-4. Real-time Output Port Mode Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
RTPM	RTPM7	RTPM6	RTPM5	RTPM4	RTPM3	RTPM2	RTPM1	RTPM0	FF34H	00H	R/W

RTPMn	Real-time Output Port Selection (n = 0 to 7)
0	Port mode
1	Real-time output port mode

Cautions 1. When using these bits as a real-time output port, set the ports to which real-time output is performed to the output mode (clear the corresponding bit of the port mode register 12 (PM12) to 0).

2. In the port specified as a real-time output port, data cannot be set to the output latch. Therefore, when setting an initial value, data should be set to the output latch before setting the real-time output mode.

(3) Real-time output port control register (RTPC)

This register sets the real-time output port operating mode and output trigger.

Table 20-3 shows the relation between the operating mode of the real-time output port and output trigger.

RTPC is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears RTPC to 00H.

Figure 20-5. Real-time Output Port Control Register Format

Symbol	7	6	5	4	3	2	①	⓪	Address	After Reset	R/W
RTPC	0	0	0	0	0	0	BYTE	EXTR	FF36H	00H	R/W

EXTR	Real-time Output Control by INTP2
0	INTP2 not specified as real-time output trigger
1	INTP2 specified as real-time output trigger

BYTE	Real-time Output Port Operating Mode
0	4 bits × 2 channels
1	8 bits × 1 channel

Table 20-3. Real-time Output Port Operating Mode and Output Trigger

BYTE	EXTR	Operating Mode	RTBH → Port Output	RTBL → Port Output
0	0	4 bits × 2 channels	INTTM2	INTTM1
	1		INTTM1	INTP2
1	0	8 bits × 1 channel	INTTM1	
	1		INTP2	

[MEMO]

CHAPTER 21 INTERRUPT AND TEST FUNCTIONS

21.1 Interrupt Function Types

The following three types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally even in the interrupt disabled status. It does not undergo interrupt priority control and is given top priority over all other interrupt requests.
It generates a standby release signal.
One interrupt source from the watchdog timer is provided for a non-maskable interrupt source.

(2) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into a high interrupt priority group and a low interrupt priority group by setting the priority specify flag register (PR0L, PR0H, PR1L).
Multiple high priority interrupts can be applied to low priority interrupts. If two or more interrupts with the same priority are simultaneously generated, each interrupts has a predetermined priority (see **Table 21-1**).
A standby release signal is generated.
Six external interrupt sources and thirteen internal interrupt sources are provided for maskable interrupt sources.

(3) Software interrupt

This is a vectored interrupt that occurs when the BRK instruction is executed. It is acknowledged even in a disabled state. The software interrupt does not undergo interrupt priority control.

21.2 Interrupt Sources and Configuration

A total of 21 non-maskable, maskable, and software interrupts are provided as the interrupt sources (see **Table 21-1**).

Table 21-1. Interrupt Source List (1/2)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2
		Name	Trigger			
Non-maskable	—	INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (with interval timer mode selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	(D)
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTP4			000EH	
	6	INTP5			0010H	
	7	INTCSI0	End of serial interface channel 0 transfer	Internal	0014H	(B)
	8	INTCSI1	End of serial interface channel 1 transfer		0016H	
	9	INTSER	Serial interface channel 2 UART reception error generation		0018H	
	10	INTSR	End of serial interface channel 2 UART reception		001AH	
		INTCSI2	End of serial interface channel 2 3-wire transfer			
	11	INTST	End of serial interface channel 2 UART transfer		001CH	

Notes 1. Default priorities are intended for two or more simultaneously generated maskable interrupt requests. 0 is the highest priority and 17 is the lowest priority.

2. Basic configuration types (A) to (E) correspond to (A) to (E) of Figure 21-1.

Table 21-1. Interrupt Source List (2/2)

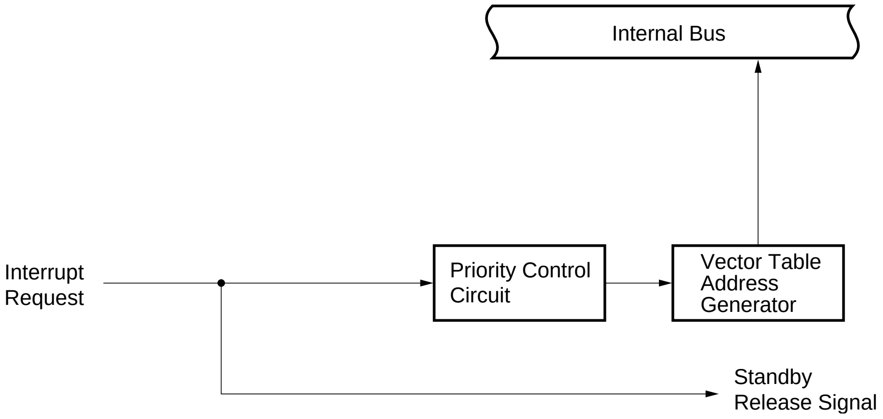
Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2
		Name	Trigger			
Maskable	12	INTTM3	Reference time interval signal from watch timer	Internal	001EH	(B)
	13	INTTM00	Generation of 16-bit timer register, capture/compare register 00 (CR00) match signal		0020H	
	14	INTTM01	Generation of 16-bit timer register, capture/compare register 01 (CR01) match signal		0022H	
	15	INTTM1	Generation of 8-bit timer/event counter 1 match signal		0024H	
	16	INTTM2	Generation of 8 bit timer/event counter 2 match signal		0026H	
	17	INTAD	End of A/D converter conversion		0028H	
Software	—	BRK	BRK instruction execution	—	003EH	(E)

Notes **1.** Default priorities are intended for two or more simultaneously generated maskable interrupt requests. 0 is the highest priority and 17 is the lowest priority.

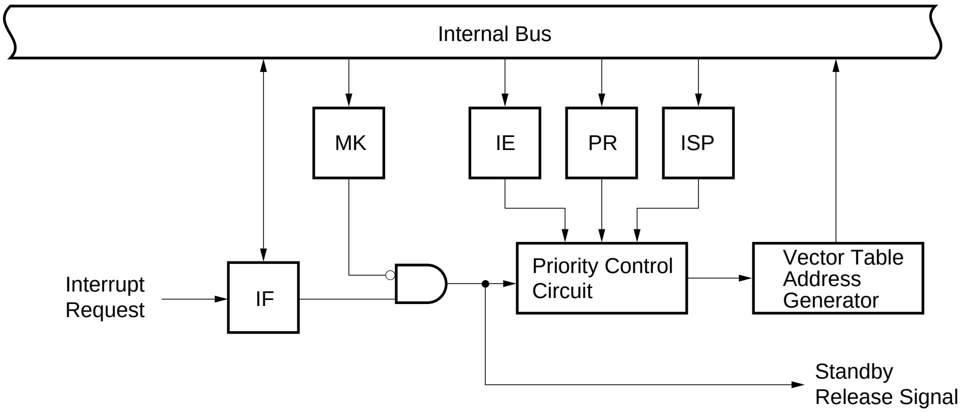
2. Basic configuration types (A) to (E) correspond to (A) to (E) of Figure 21-1.

Figure 21-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt (INTP0)

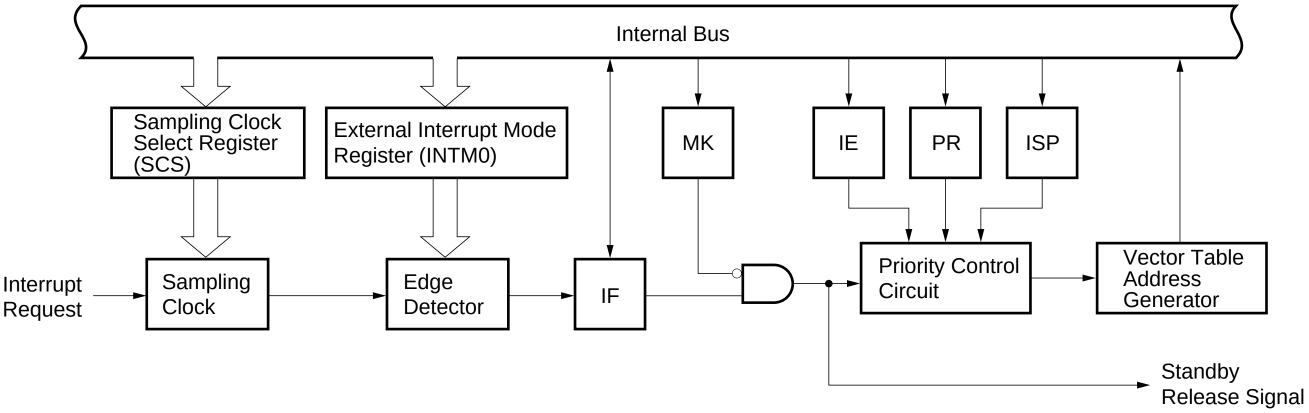
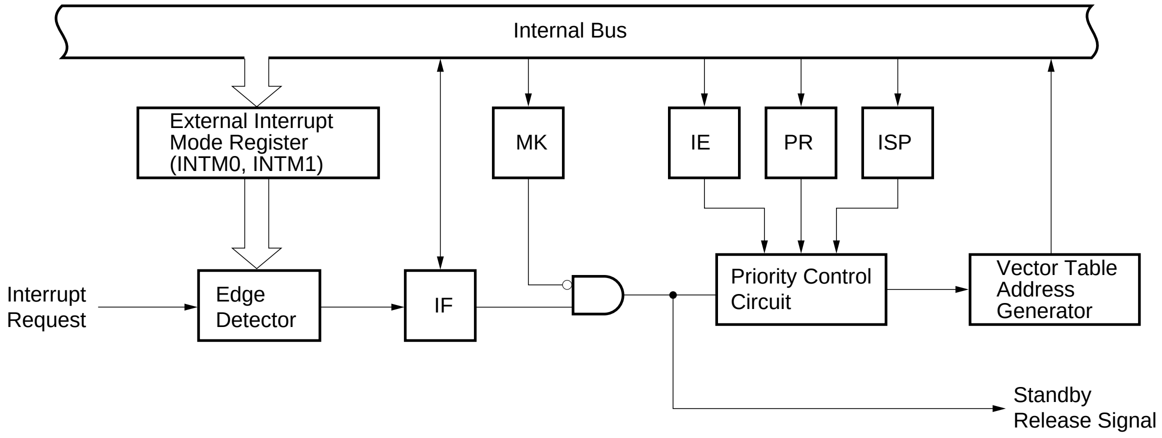
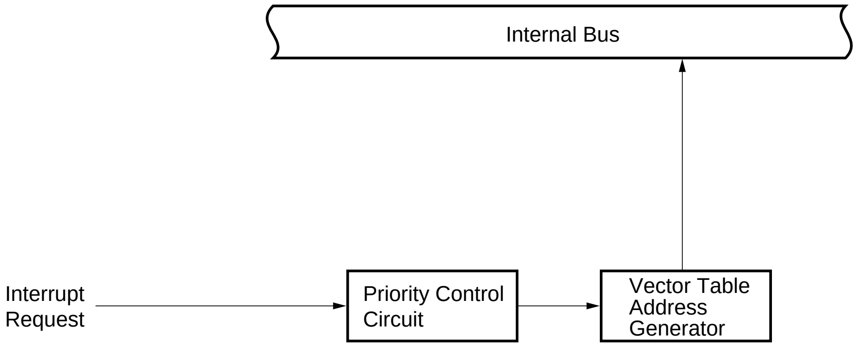


Figure 21-1. Basic Configuration of Interrupt Function (2/2)

(D) External maskable interrupt (except INTP0)



(E) Software interrupt



Remark			
	IF	:	Interrupt request flag
	IE	:	Interrupt enable flag
	ISP	:	Inservice priority flag
	MK	:	Interrupt mask flag
	PR	:	Priority specify flag

21.3 Interrupt Function Control Registers

The following six types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H, IF1L)
- Interrupt mask flag register (MK0L, MK0H, MK1L)
- Priority specify flag register (PR0L, PR0H, PR1L)
- External interrupt mode register (INTM0, INTM1)
- Sampling clock select register (SCS)
- Program status word (PSW)

Table 21-2 gives a listing of interrupt request flags, interrupt mask flags, and priority specify flags corresponding to interrupt request sources.

Table 21-2. Various Flags Corresponding to Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specify Flag	
		Register		Register		Register
INTWDT INTP0 INTP1 INTP2 INTP3 INTP4 INTP5	TMIF4 PIF0 PIF1 PIF2 PIF3 PIF4 PIF5	IF0L	TMMK4 PMK0 PMK1 PMK2 PMK3 PMK4 PMK5	MK0L	TMPR4 PPR0 PPR1 PPR2 PPR3 PPR4 PPR5	PR0L
INTCSI0 INTCSI1 INTSER INTSR/INTCSI2 INTST INTTM3 INTTM00 INTTM01	CSIIF0 CSIIF1 SERIF SRIF STIF TMIF3 TMIF00 TMIF01	IF0H	CSIMK0 CSIMK1 SERMK SRMK STMK TMMK3 TMMK00 TMMK01	MK0H	CSIPR0 CSIPR1 SERPR SRPR STPR TMPR3 TMPR00 TMPR01	PR0H
INTTM1 INTTM2 INTAD	TMIF1 TMIF2 ADIF	IF1L	TMMK1 TMMK2 ADMK	MK1L	TMPR1 TMPR2 ADPR	PR1L

(1) Interrupt request flag registers (IF0L, IF0H, IF1L)

The interrupt request flag is set to 1 when the corresponding interrupt request is generated or an instruction is executed. It is cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon application of $\overline{\text{RESET}}$ input.

IF0L, IF0H, and IF1L are set with a 1-bit or an 8-bit memory manipulation instruction. If IF0L and IF0H are used as a 16-bit register IF0, use a 16-bit memory manipulation instruction for the setting.

$\overline{\text{RESET}}$ input clears these registers to 00H.

Figure 21-2. Interrupt Request Flag Register Format

Symbol	7	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
IF0L	0	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	TMIF4	FFE0H	00H	R/W
	⑦	⑥	⑤	④	③	②	①	⓪			
IF0H	TMIF01	TMIF00	TMIF3	STIF	SRIF	SERIF	CSIIF1	CSIIF0	FFE1H	00H	R/W
	⑦	6	5	4	3	②	①	⓪			
IF1L	WTIF[Note]	0	0	0	0	ADIF	TMIF2	TMIF1	FFE2H	00H	R/W

×× IF×	Interrupt Request Flag
0	No interrupt request signal
1	Interrupt request signal is generated; Interrupt request state

Note WTIF is test input flag. Vectored interrupt request is not generated.

Cautions 1. TMIF4 flag is R/W enabled only when a watchdog timer is used as an interval timer. If a watchdog timer is used in watchdog timer mode 1, set TMIF4 flag to 0.

2. Set always 0 to IF0L bit 7 and IF1L bits 3 to 6.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L)

The interrupt mask flag is used to enable/disable the corresponding maskable interrupt service and to set standby clear enable/disable.

MK0L, MK0H, and MK1L are set with a 1-bit or an 8-bit memory manipulation instruction. If MK0L and MK0H are used as a 16-bit register MK0, use a 16-bit memory manipulation instruction for the setting.

$\overline{\text{RESET}}$ input sets these registers to FFH.

Figure 21-3. Interrupt Mask Flag Register Format

Symbol	7	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
MK0L	1	PMK5	PMK4	PMK3	PMK2	PMK	PMK	TMMK4	FFE4H	FFH	R/W
	⑦	⑥	⑤	④	③	②	①	⓪			
MK0H	TMMK01	TMMK00	TMMK3	STMK	SRMK	SERMK	CSIMK1	CSIMK0	FFE5H	FFH	R/W
	⑦	6	5	4	3	②	①	⓪			
MK1L	WTMK[Note]	1	1	1	1	ADMK	TMMK2	TMMK1	FFE6H	FFH	R/W

××MK×	Interrupt Servicing Control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Note WTMK controls standby mode release enable/disable. Does not control the interrupt function.

Cautions 1. If TMMK4 flag is read when a watchdog timer is used in watchdog timer mode 1, MK0 value becomes undefined.

2. Because port 0 also functions as the external interrupt request input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, 1 should be set to the interrupt mask flag before using the output mode.

3. Set always 1 to MK0L bit 7 and MK1L bits 3 to 6.

(3) Priority specify flag registers (PR0L, PR0H, and PR1L)

The priority specify flag is used to set the corresponding maskable interrupt priority orders.

PR0L, PR0H, and PR1L are set with a 1-bit or an 8-bit memory manipulation instruction. If PR0L and PR0H are used as a 16-bit register PR0, use a 16-bit memory manipulation instruction for the setting.

$\overline{\text{RESET}}$ input sets these registers to FFH.

Figure 21-4. Priority Specify Flag Register Format

Symbol	7	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
PR0L	1	PPR5	PPR4	PPR3	PPR2	PPR1	PPR0	TMPR4	FFE8H	FFH	R/W
	⑦	⑥	⑤	④	③	②	①	⓪			
PR0H	TMPR01	TMPR00	TMPR3	STPR	SRPR	SERPR	CSIPR1	CSIPR0	FFE9H	FFH	R/W
	7	6	5	4	3	②	①	⓪			
PR1L	1	1	1	1	1	ADPR	TMPR2	TMPR1	FFEAH	FFH	R/W

×× PR ×	Priority Level Selection
0	High priority level
1	Low priority level

Cautions 1. If a watchdog timer is used in watchdog timer mode 1, set TMPR4 flag to 1.

2. Set always 1 to PR0L bit 7 and PR1L bits 3 to 7.

(4) External interrupt mode register (INTM0, INTM1)

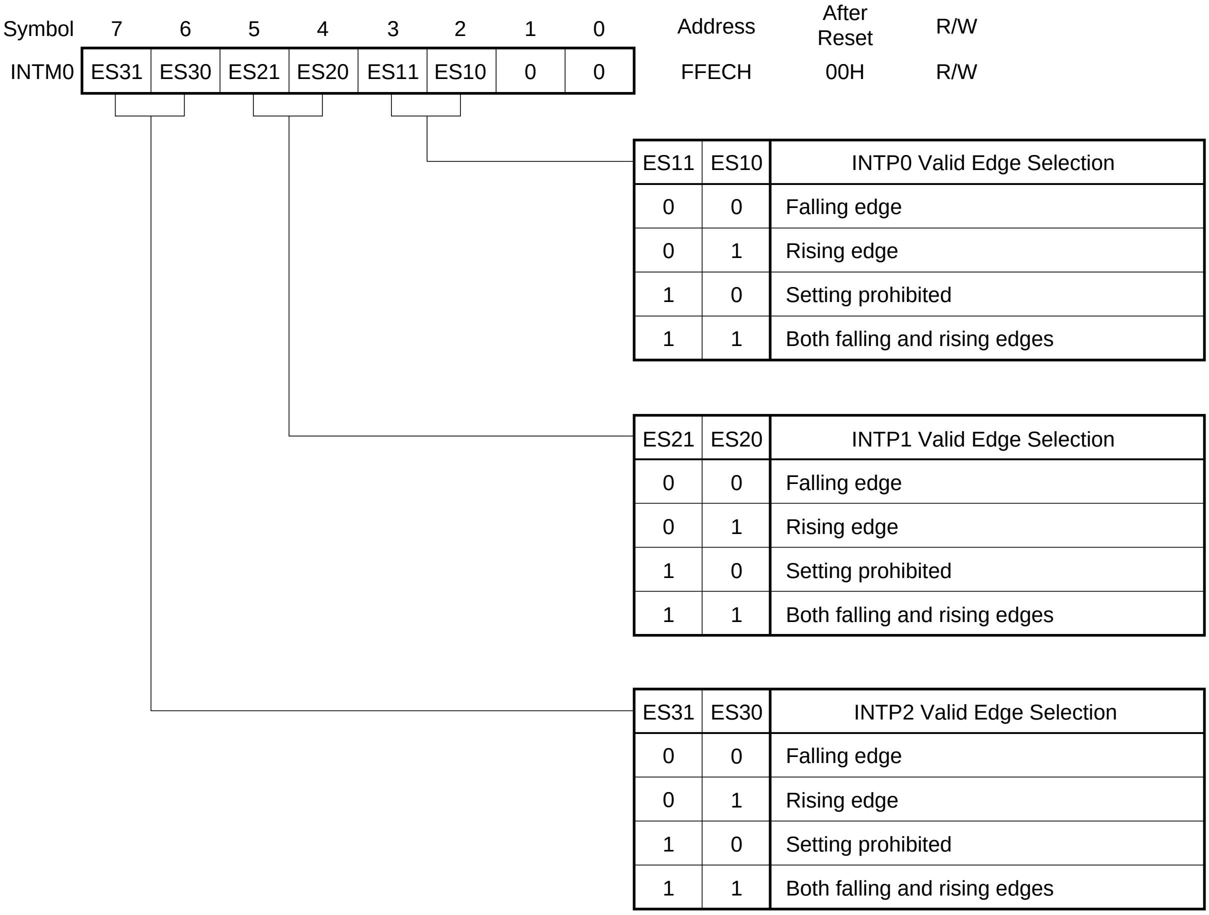
These registers set the valid edge for INTP0 to INTP5.

INTM0 specifies the valid edges of interrupt pins INTP0 to INTP2, and INTM1 specifies the valid edges of INTP3 to INTP5.

INTM0 and INTM1 are set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears these registers to 00H.

Figure 21-5. External Interrupt Mode Register 0 Format



Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
INTM0	ES31	ES30	ES21	ES20	ES11	ES10	0	0	FFECH	00H	R/W

ES11	ES10	INTP0 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES21	ES20	INTP1 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES31	ES30	INTP2 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

Caution Before setting the valid edge of the INTP0/TIO0/P00 pin, stop the timer operation by clearing the bits 1 to 3 (TMC01 to TMC03) of the 16-bit timer mode control register to 0, 0, 0.

Figure 21-6. External Interrupt Mode Register 1 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
INTM1	0	0	ES61	ES60	ES51	ES50	ES41	ES40	FFEDH	00H	R/W

ES41	ES40	INTP3 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES51	ES50	INTP4 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES61	ES60	INTP5 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

(5) Sampling clock select register (SCS)

This register is used to set the valid edge clock sampling clock to be input to INTP0. When remote controlled data reception is carried out using INTP0, digital noise is eliminated with sampling clocks.

SCS is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears SCS to 00H.

Figure 21-7. Sampling Clock Select Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
SCS	0	0	0	0	0	0	SCS1	SCS0	FF47H	00H	R/W

SCS1	SCS0	INTP0 Sampling Clock Selection		
			MCS = 1	MCS = 0
0	0	$f_{XX}/2^{N}$		
0	1	$f_{XX}/2^{7}$	$f_X/2^{7}$ (39.1 kHz)	$f_X/2^{8}$ (19.5 kHz)
1	0	$f_{XX}/2^{5}$	$f_X/2^{5}$ (156.3 kHz)	$f_X/2^{6}$ (78.1 kHz)
1	1	$f_{XX}/2^{6}$	$f_X/2^{6}$ (78.1 kHz)	$f_X/2^{7}$ (39.1 kHz)

Caution **$f_{XX}/2^{N}$ is a clock to be supplied to the CPU and $f_{XX}/2^{5}$, $f_{XX}/2^{6}$, and $f_{XX}/2^{7}$ are clocks to be supplied to the peripheral hardware. $f_{XX}/2^{N}$ stops in the HALT mode.**

Remarks **1.** N : Value (N = 0 to 4) at bits 0 to 2 (PCC0 to PCC2) of processor clock control register (PCC)

2. f_{XX} : Main system clock frequency (f_X or $f_X/2$)

3. f_X : Main system clock oscillation frequency

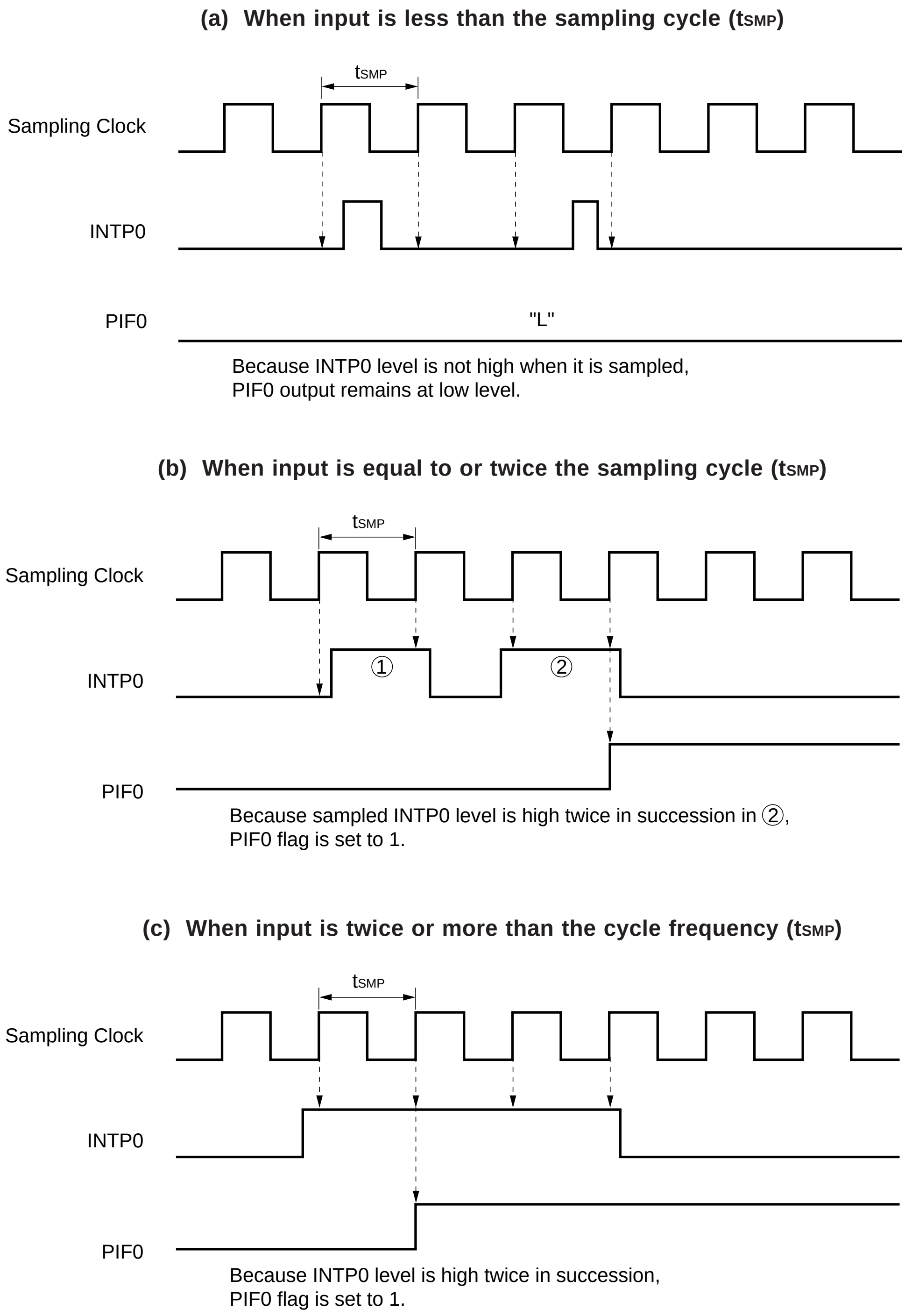
4. MCS : Bit 0 of oscillation mode selection register (OSMS)

5. Values in parentheses apply to operation with f_X = 5.0 MHz.

When the sampled INTP0 input level is active twice in succession, the noise eliminator sets the interrupt request flag (PIF0) to 1.

Figure 21-8 shows the input/output timing of the noise eliminator.

Figure 21-8. Noise Eliminator Input/Output Timing (During Rising Edge Detection)



(a) When input is less than the sampling cycle (t_{SMP})

Because INTP0 level is not high when it is sampled, PIF0 output remains at low level.

(b) When input is equal to or twice the sampling cycle (t_{SMP})

Because sampled INTP0 level is high twice in succession in ②, PIF0 flag is set to 1.

(c) When input is twice or more than the cycle frequency (t_{SMP})

Because INTP0 level is high twice in succession, PIF0 flag is set to 1.

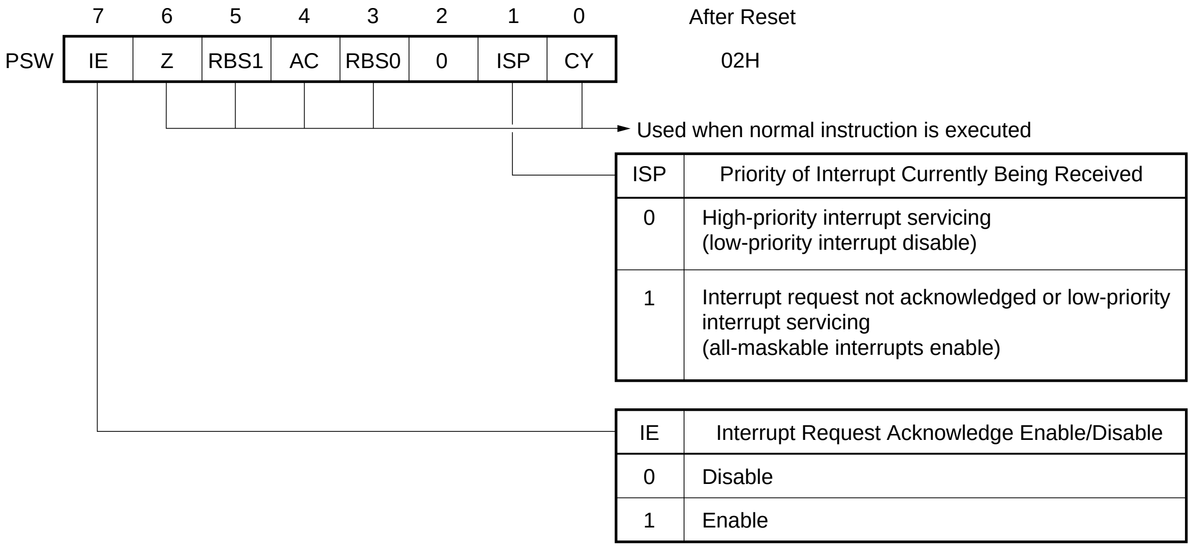
(6) Program status word (PSW)

The program status word is a register to hold the instruction execution result and the current status for interrupt request. The IE flag to set maskable interrupt enable/disable and the ISP flag to control multiple interrupt processing are mapped.

Besides 8-bit unit read/write, this register can carry out operations with a bit manipulation instruction and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged or when the BRK instruction is executed, the contents of PSW are automatically saved to the stack and the IE flag is reset to 0. If a maskable interrupt request is acknowledged contents of the priority specify flag of the acknowledged interrupt are transferred to the ISP flag. The contents of PSW can also be saved into the stack with the PUSH PSW instruction. It is reset from the stack with the RETI, RETB, and POP PSW instructions.

$\overline{\text{RESET}}$ input sets PSW to 02H.

Figure 21-9. Program Status Word Format



	7	6	5	4	3	2	1	0	After Reset
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY	02H

ISP	Priority of Interrupt Currently Being Received
0	High-priority interrupt servicing (low-priority interrupt disable)
1	Interrupt request not acknowledged or low-priority interrupt servicing (all-maskable interrupts enable)

IE	Interrupt Request Acknowledge Enable/Disable
0	Disable
1	Enable

21.4 Interrupt Servicing Operations

21.4.1 Non-maskable interrupt request acknowledge operation

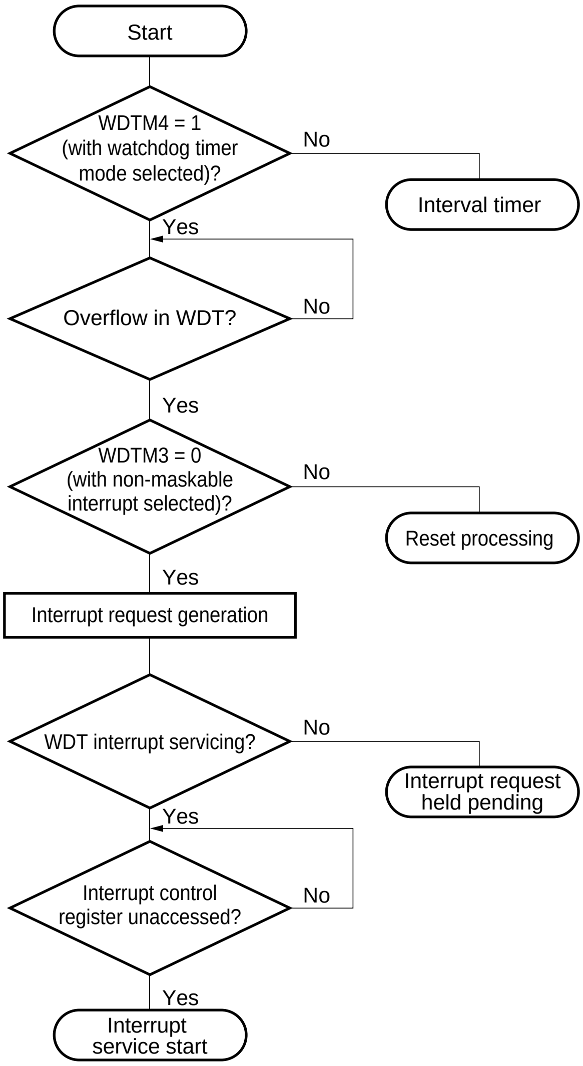
A non-maskable interrupt request is unconditionally acknowledged even if in an interrupt request acknowledge disable state. It does not undergo interrupt priority control and has highest priority over all other interrupts.

If a non-maskable interrupt request is acknowledged, the contents of acknowledged interrupt are saved in the stacks, PSW and PC, in that order, the IE and ISP flags are reset to 0, and the vector table contents are loaded into PC and branched.

A new non-maskable interrupt request generated during execution of a non-maskable interrupt servicing program is acknowledged after the current execution of the non-maskable interrupt servicing program is terminated (following RETI instruction execution) and one main routine instruction is executed. If a new non-maskable interrupt request is generated twice or more during non-maskable interrupt service program execution, only one non-maskable interrupt request is acknowledged after termination of the non-maskable interrupt service program execution.

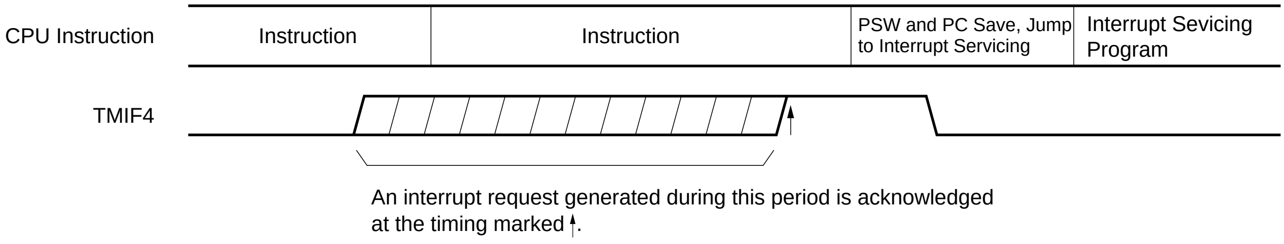
Figure 21-10 shows the flow chart illustrating how the non-maskable interrupt request occurs and is acknowledged. Figure 21-11 shows the acknowledge timing of the non-maskable interrupt request. Figure 21-12 shows acknowledge operation of multiple non-maskable interrupt requests.

Figure 21-10. Non-Maskable Interrupt Request Occurrence and Acknowledge Flowchart



WDTM : Watchdog timer mode register
WDT : Watchdog timer

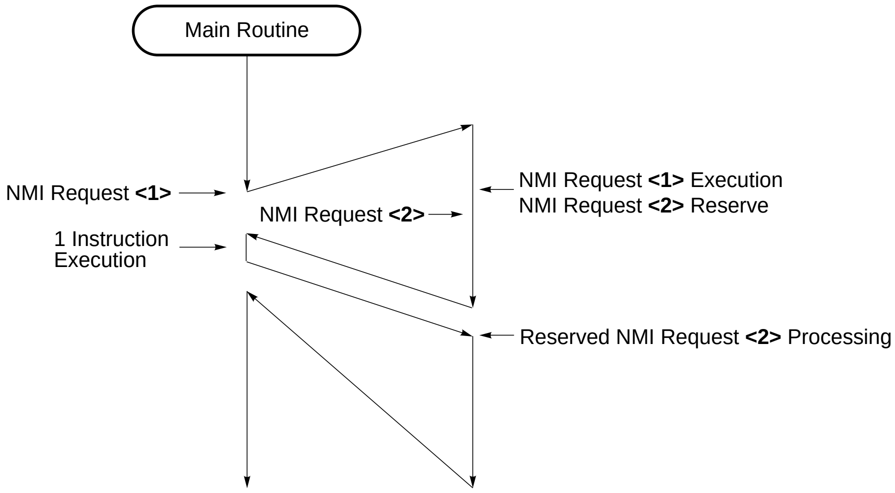
Figure 21-11. Non-Maskable Interrupt Request Acknowledge Timing



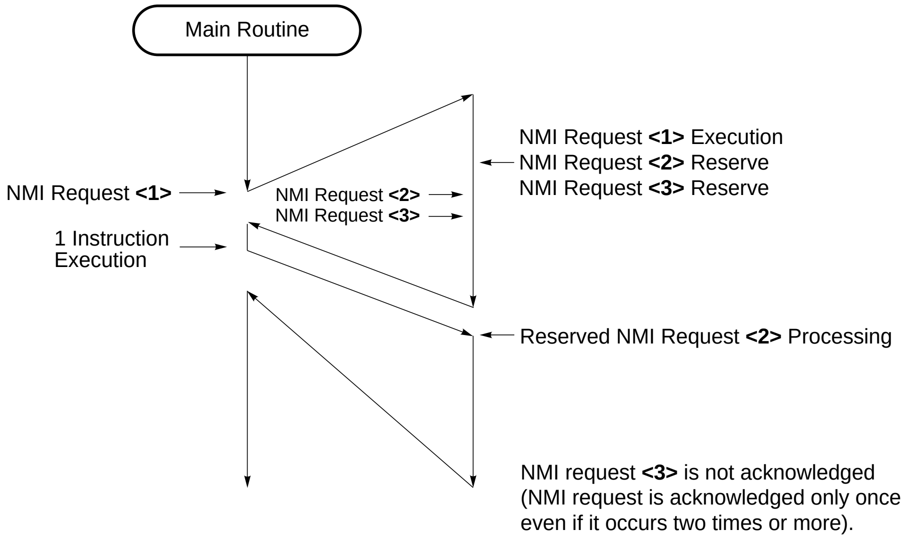
TMIF4 : Watchdog timer interrupt request flag

Figure 21-12. Non-Maskable Interrupt Request Acknowledge Operation

(a) If a new non-maskable interrupt request is generated during non-maskable interrupt servicing program execution



(b) If two non-maskable interrupt requests are generated during non-maskable interrupt servicing program execution


Main Routine
NMI Request <1>
NMI Request <2>
NMI Request <3>
1 Instruction Execution
NMI Request <1> Execution
NMI Request <2> Reserve
NMI Request <3> Reserve
Reserved NMI Request <2> Processing
NMI request <3> is not acknowledged (NMI request is acknowledged only once even if it occurs two times or more).

21.4.2 Maskable interrupt request acknowledge operation

A maskable interrupt request becomes acknowledgeable when an interrupt request flag is set to 1 and the interrupt mask (MK) flag is cleared to 0. A vectored interrupt request is acknowledged in an interrupt enable state (with IE flag set to 1). However, a low-priority interrupt is not acknowledged during high-priority interrupt service (with ISP flag reset to 0).

Wait times from maskable interrupt request generation to interrupt servicing are shown in Table 21-3.

For the timing to acknowledge an interrupt request, refer to **Figures 21-14 and 21-15**.

Table 21-3. Times from Maskable Interrupt Request Generation to Interrupt Service

	Minimum Time	Maximum Time Note
When ××PR = 0	7 clocks	32 clocks
When ××PR = 1	8 clocks	33 clocks

Note If an interrupt request is generated just before a divide instruction, the wait time is maximized.

Remark 1 clock : $\frac{1}{f_{CPU}}$ (f_{CPU}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request specified for higher priority with the priority specify flag is acknowledged first. If two or more requests specified for the same priority by the interrupt priority specify flag, the one with the higher default priority is acknowledged first.

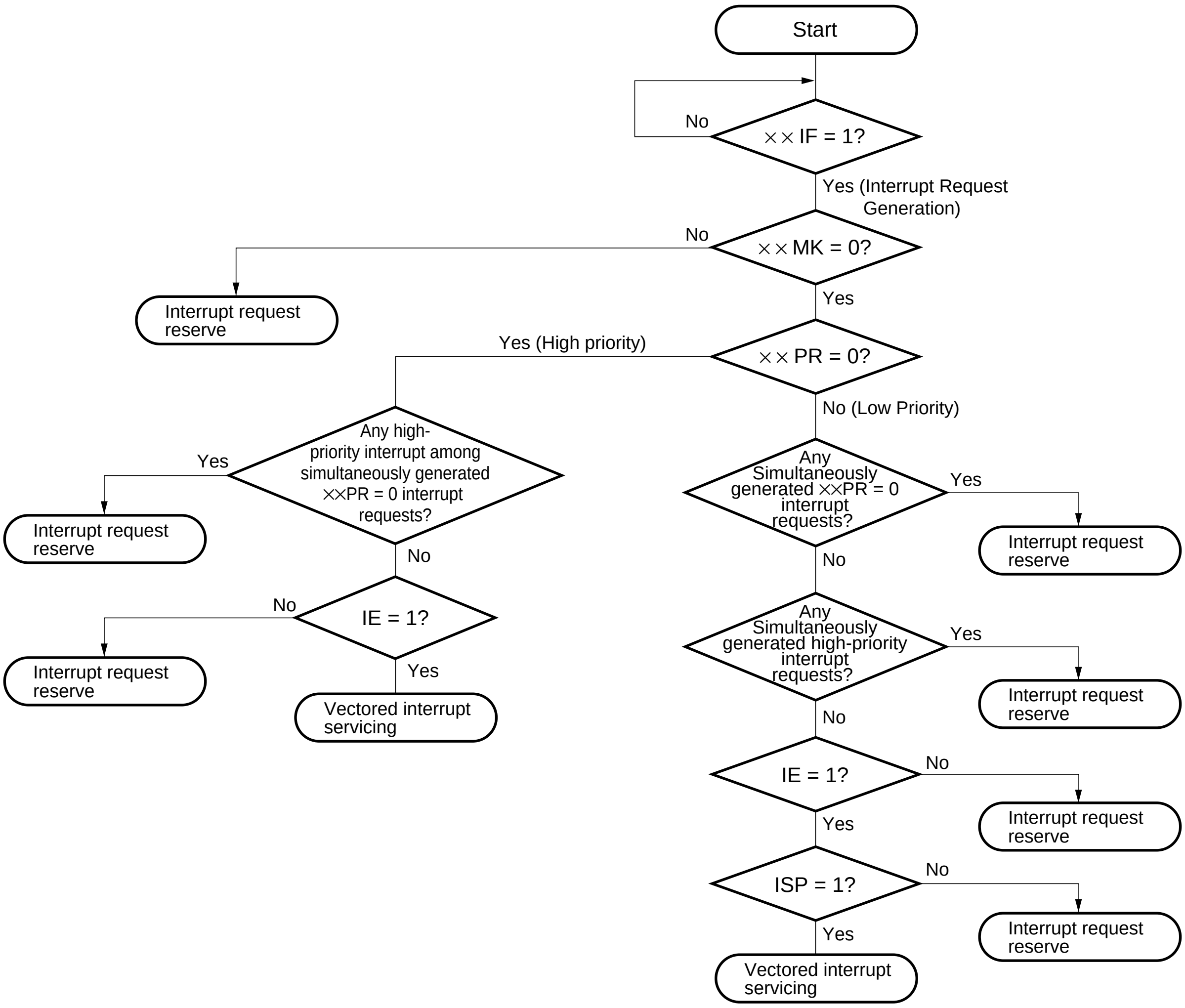
Any reserved interrupt requests are acknowledged when they become acknowledgeable.

Figure 21-13 shows interrupt request acknowledge algorithms.

If a maskable interrupt request is acknowledged, the contents of acknowledged interrupt are saved in the stacks, program status word (PSW) and program counter (PC), in that order, the IE flag is reset to 0, and the acknowledged interrupt priority specify flag contents are transferred to the ISP flag. Further, the vector table data determined for each interrupt request is loaded into PC and branched.

Return from the interrupt is possible with the RETI instruction.

Figure 21-13. Interrupt Request Acknowledge Processing Algorithm


Start
No
×× IF = 1?
Yes (Interrupt Request Generation)
No
×× MK = 0?
Interrupt request reserve
Yes
Yes (High priority)
×× PR = 0?
No (Low Priority)
Any high-priority interrupt among simultaneously generated ××PR = 0 interrupt requests?
Yes
Interrupt request reserve
No
No
IE = 1?
Interrupt request reserve
Yes
Vectored interrupt servicing
Any Simultaneously generated ××PR = 0 interrupt requests?
Yes
Interrupt request reserve
No
Any Simultaneously generated high-priority interrupt requests?
Yes
Interrupt request reserve
No
IE = 1?
No
Interrupt request reserve
Yes
ISP = 1?
No
Interrupt request reserve
Yes
Vectored interrupt servicing

××IF: Interrupt request flag

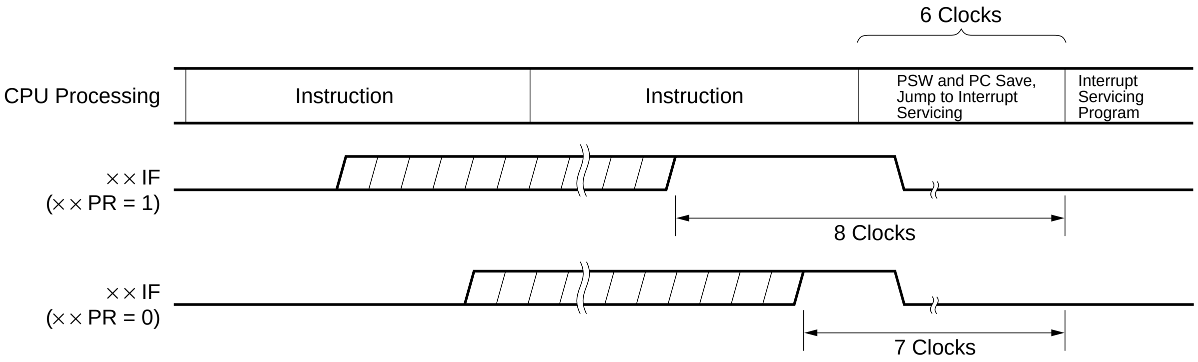
××MK: Interrupt mask flag

××PR: Priority specify flag

IE: Flag that controls maskable interrupt request acknowledge (1 = enable, 0 = disable)

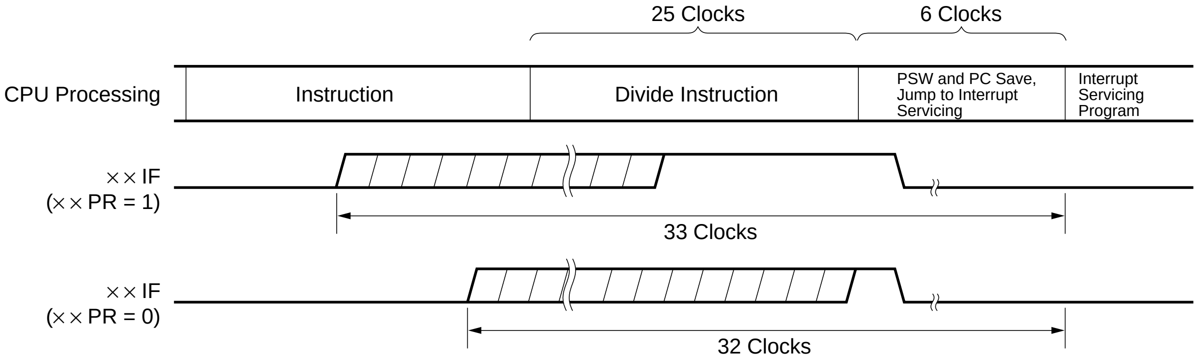
ISP: Flag indicating priority of interrupt currently processed (0 = interrupt with high priority is serviced. 1 = interrupt request is not acknowledged or interrupt with low priority is serviced).

Figure 21-14. Interrupt Request Acknowledge Timing (Minimum Time)



Remark 1 clock : $\frac{1}{f_{CPU}}$ (f_{CPU}: CPU clock)

Figure 21-15. Interrupt Request Acknowledge Timing (Maximum Time)



Remark 1 clock : $\frac{1}{f_{CPU}}$ (f_{CPU}: CPU clock)

21.4.3 Software interrupt request acknowledge operation

A software interrupt request is acknowledged by BRK instruction execution. Software interrupt cannot be disabled.

If a software interrupt request is acknowledged, the contents of acknowledged interrupts are saved in the stacks, program status word (PSW) and program counter (PC), in that order, the IE flag is reset to 0 and the contents of the vector tables (003EH and 003FH) are loaded into PC and branched.

Return from the software interrupt is possible with the RETB instruction.

Caution Do not use the RETI instruction for returning from the software interrupt.

21.4.4 Multiple interrupt servicing

Acknowledging another interrupt request while one interrupt is processed is called multiple interrupts.

A multiple interrupt is not generated unless the interrupt request is enabled (IE = 1) (except the non-maskable interrupt). When an interrupt request is acknowledged, the other interrupts are disabled (IE = 0). To enable a multiple interrupt, therefore, the IE flag must be set to 1 by executing the EI instruction during interrupt servicing and the interrupt must be enabled. Even if the interrupt request is enabled, some multiple instructions are not acknowledged. However, the multiple instructions are controlled by the programmable priority.

An interrupt has two types of priorities: default priority and programmable priority. The multiple interrupt is controlled by the programmable priority.

In the EI status, if an interrupt request having the same as or higher priority than that of the interrupt currently serviced is generated, the interrupt is acknowledged as multiple interrupt. If an interrupt request with a priority lower than that of the interrupt currently serviced is generated, the interrupt is not acknowledged as multiple interrupt.

If an interrupt is disabled, or if a multiple interrupt is not acknowledged because it has a low priority, the interrupt is kept pending. After the servicing of the current interrupt has been completed, and after one instruction of the main processing has been executed, the pending interrupt is acknowledged.

Multiple interrupts are not acknowledged while the non-maskable interrupt is serviced.

Table 21-4 shows interrupt requests enabled for multiple interrupts. Figure 21-16 shows multiple interrupt examples.

Table 21-4. Interrupt Request Enabled for Multiple Interrupt During Interrupt Servicing

Multiple Interrupt Request Servicing Interrupt		Non-maskable Interrupt Request	Maskable Interrupt Request			
			PR = 0		PR = 1	
			IE = 1	IE = 0	IE = 1	IE = 0
Non-maskable interrupt		D	D	D	D	D
Maskable interrupt	ISP = 0	E	E	D	D	D
	ISP = 1	E	E	D	E	D
Software interrupt		E	E	D	E	D

Remarks 1. E : Multiple interrupt enable

2. D : Multiple interrupt disable

3. ISP and IE are the flags contained in PSW

ISP = 0 : An interrupt with higher priority is being serviced

ISP = 1 : An interrupt request is not accepted or an interrupt with lower priority is being serviced

IE = 0 : Interrupt request acknowledge is disabled

IE = 1 : Interrupt request acknowledge is enabled

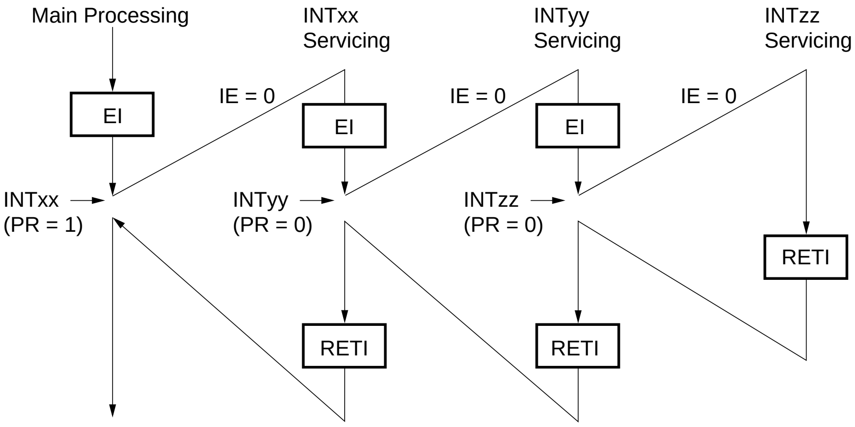
4. PR is a flag contained in PR0L, PR0H, and PR1L

PR = 0 : Higher priority level

PR = 1 : Lower priority level

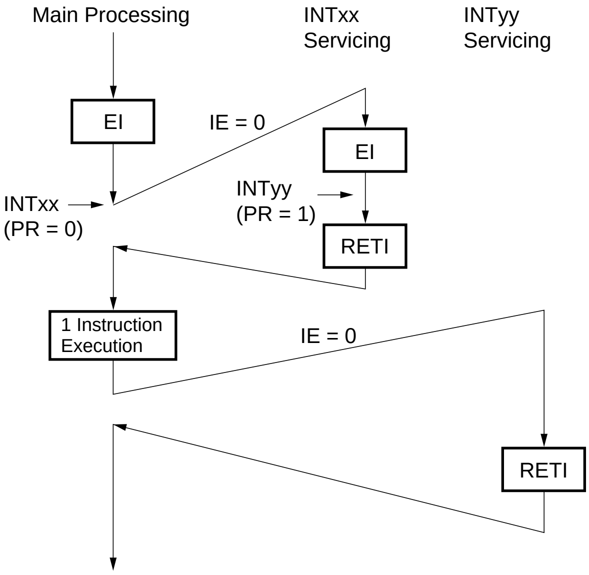
Figure 21-16. Multiple Interrupt Example (1/2)

Example 1. Two multiple interrupts are acknowledged.



Two multiple interrupt requests INTyy and INTzz are acknowledged while interrupt INTxx is serviced. Before each interrupt request is acknowledged, the EI instruction is always issued and the interrupt request is enabled.

Example 2. Multiple interrupt is not acknowledged because of its priority.

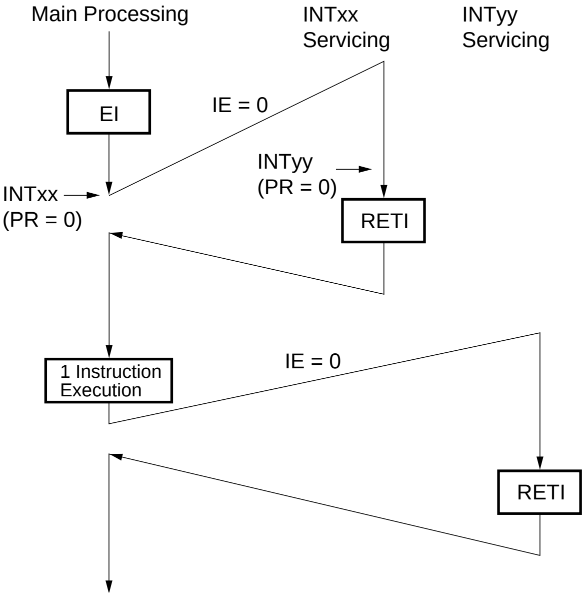


INTyy that occurs while INTxx is serviced is not acknowledged as a multiple interrupt because the priority of INTyy is lower than that of INTxx. INTyy is reserved and is acknowledged after one instruction of the main processing has been executed.

PR = 0:	High-priority interrupt
PR = 1:	Low-priority interrupt
IE = 0:	Interrupt acknowledge disabled

Figure 21-16. Multiple Interrupt Example (2/2)

Example 3. Multiple interrupt is not acknowledged because an interrupt is not enabled.



In the servicing of INTxx, other interrupts are not enabled (the EI instruction is not executed). Therefore, INTyy is not acknowledged as a multiple interrupt. This interrupt is reserved and acknowledged after one instruction of the main processing has been executed.

PR = 0: High-priority interrupt
IE = 0: Interrupt acknowledge disabled

21.4.5 Interrupt request reserve

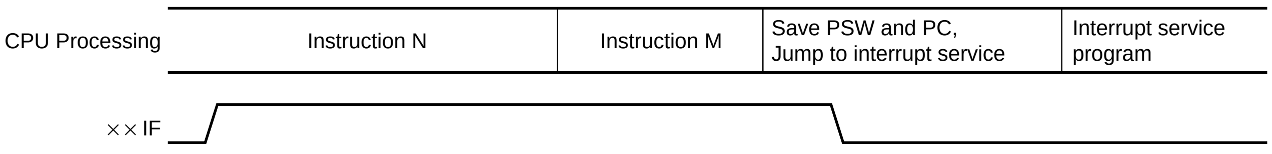
Even if an interrupt request is generated, some instructions reserve interrupt acknowledge while the current instruction is executed and until execution of the next instruction is completed. The instruction that reserve interrupt requests (interrupt request reserve) are shown below.

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW.bit, CY
- MOV1 CY, PSW.bit
- AND1 CY, PSW.bit
- OR1 CY, PSW.bit
- XOR1 CY, PSW.bit
- SET1 PSW.bit
- CLR1 PSW.bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW.bit, $addr16
- BF PSW.bit, $addr16
- BTCLR PSW.bit, $addr16
- EI
- DI
- Manipulate instructions for IF0L, IF0H, IF1L, MK0L, MK0H, MK1L, PR0L, PR0H, PR1L, INTM0, INTM1 registers

Caution The BRK instruction is not an interrupt request reserve instruction. However, the IE flag is cleared to 0 with a software interrupt that is started by BRK instruction execution. Thus, even if a maskable interrupt request is generated during BRK instruction, interrupt requests are not acknowledged. However, non-maskable interrupt requests are acknowledged.

Figure 21-17 shows the timing at which an interrupt request is reserved.

Figure 21-17. Interrupt Request Hold



Remarks 1. Instruction N: Instruction that holds interrupts requests

2. Instruction M: Instructions other than instruction N

3. The ××PR (priority level) values do not affect the operation of ××IF (interrupt request).

21.5 Test Functions

When the watch timer overflows, port 4 falling edge is detected, an internal test input flag is set to 1, and the standby release signal is generated.

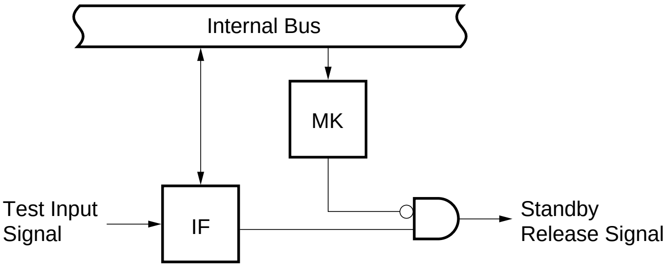
Unlike the interrupt function, vectored processing is not performed.

There are two test input factors as shown in Table 21-5. The basic configuration is shown in Figure 21-18.

Table 21-5. Test Input Factors

Test Input Factors		Internal/ External
Name	Trigger	
INTWT	Watch timer overflow	Internal
INTPT4	Falling edge detection at port 4	External

Figure 21-18. Basic Configuration of Test Function



Remark IF: Test input flag
MK: Test mask flag

21.5.1 Registers controlling the test function

The test function is controlled by the following three registers.

- Interrupt request flag register 1L (IF1L)
- Interrupt mask flag register 1L (MK1L)
- Key return mode register (KRM)

The names of the test input flags and test mask flags corresponding to the test input signals are listed in Table 21-6.

Table 21-6. Flags Corresponding to Test Input Signals

Test Input Signal Name	Test Input Flag	Test Mask Flag
INTWT	WTIF	WTMK
INTPT4	KRIF	KRMK

(1) Interrupt request flag register 1L (IF1L)

It indicates whether a watch timer overflow is detected or not.

IF1L is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears IF1L to 00H.

Figure 21-19. Format of Interrupt Request Flag Register 1L

Symbol	⑦	6	5	4	3	②	①	⓪	Address	After Reset	R/W
IF1L	WTIF	0	0	0	0	ADIF	TMIF2	TMIF1	FFE2H	00H	R/W

WTIF	Watch Timer Overflow Detection Flag
0	Not detected
1	Detected

Caution Be sure to set bits 3 to 6 to 0.

(2) Interrupt mask flag register 1L (MK1L)

It is used to set the standby mode enable/disable at the time the standby mode is released by the watch timer.

MK1L is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets MK1L to FFH.

Figure 21-20. Format of Interrupt Mask Flag Register 1L

Symbol	⑦	6	5	4	3	②	①	⓪	Address	After Reset	R/W
MK1L	WTMK	1	1	0	0	ADMK	TMMK2	TMMK1	FFE6H	FFH	R/W

WTMK	Standby Mode Control by Watch Timer
0	Enables releasing the standby mode.
1	Disables releasing the standby mode.

Caution Be sure to set bits 3 to 6 to 1.

(3) Key return mode register (KRM)

This register is used to set enable/disable of standby function clear by key return signal (port 4 falling edge detection).

KRM is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets KRM to 02H.

Figure 21-21. Key Return Mode Register Format

Symbol	7	6	5	4	3	2	①	⓪	Address	After Reset	R/W
KRM	0	0	0	0	0	0	KRMK	KRIF	FFF6H	02H	R/W

KRIF	Key Return Signal
0	Not detected
1	Detected (port 4 falling edge detection)

KRMK	Standby Mode Control by Key Return Signal
0	Standby mode release enabled
1	Standby mode release disabled

Caution When port 4 falling edge detection is used, be sure to clear KRIF to 0 (not cleared to 0 automatically)

21.5.2 Test input signal acknowledge operation

(1) Internal test signal (INTWT)

INTWT is generated when the watch timer overflows, and sets the WTIF flag. Unless interrupts are masked by the interrupt mask flag (WTMK) at this time, the standby release signal is generated.

The watch function is available by checking the WTIF flag at a shorter cycle than the watch timer overflow cycle.

(2) External test input signal (INTPT4)

INTPT4 is generated when a falling edge is input to the port 4 (P40 to P47) pins, and KRIF is set. Unless interrupts are masked by the interrupt mask flag (KRMK) at this time, the standby release signal is generated.

If port 4 is used as key matrix return signal input, whether or not a key input has been applied can be checked from the KRIF status.

[MEMO]

CHAPTER 22 EXTERNAL DEVICE EXPANSION FUNCTION

22.1 External Device Expansion Functions

The external device expansion functions connect external devices to are as 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 22-1. 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
$\overline{\text{RD}}$	Read strobe signal	P64
$\overline{\text{WR}}$	Write strobe signal	P65
$\overline{\text{WAIT}}$	Wait signal	P66
ASTB	Address strobe signal	P67

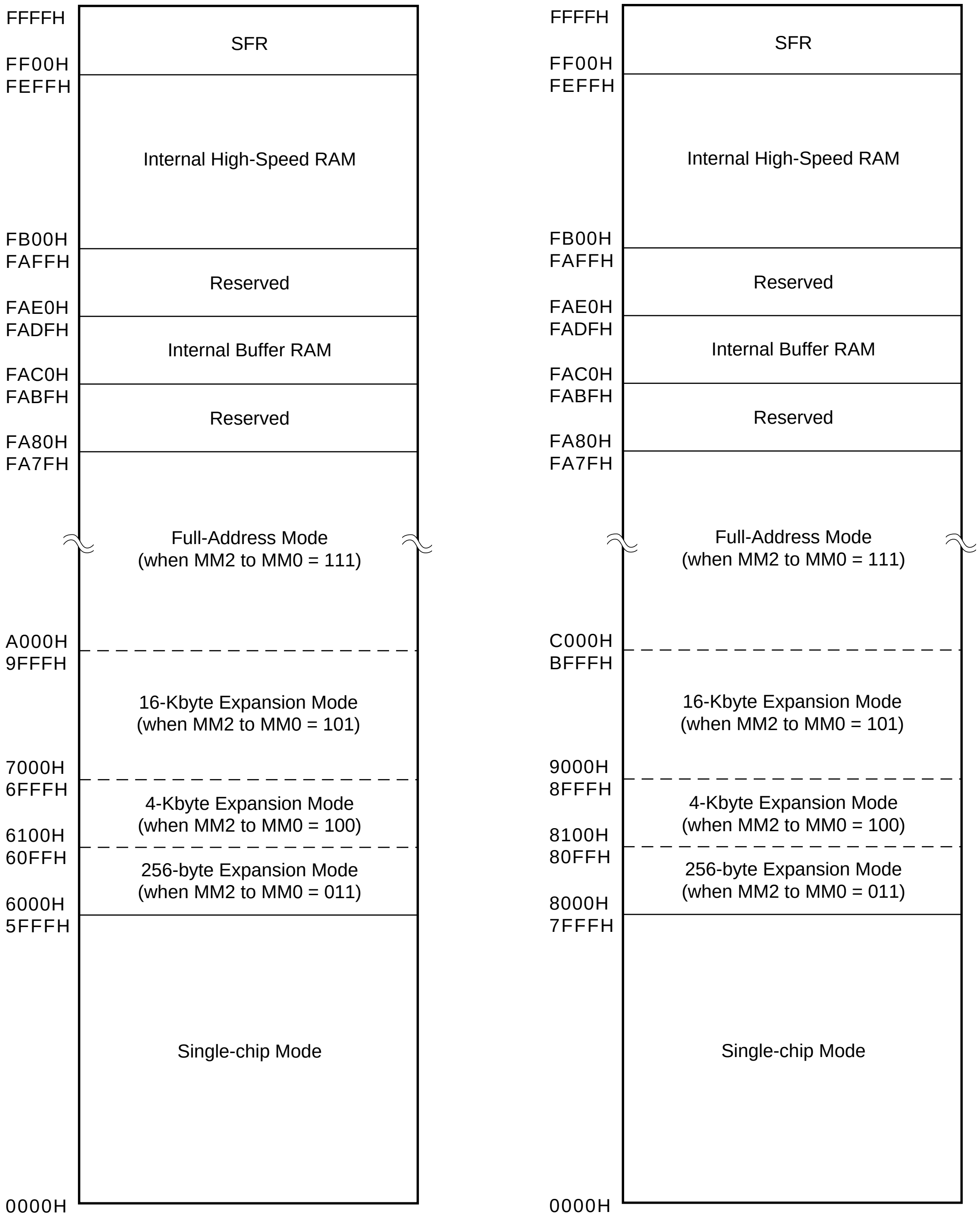
Table 22-2. State of Ports 4 to 6 Pins in External Memory Expansion Mode

Ports and bits External Expansion Modes	Port 4	Port 5								Port 6	
	0 to 7	0	1	2	3	4	5	6	7	0 to 3	4 to 7
Single-chip mode	Port	Port								Port	Port
256-byte expansion mode	Address/data	Port								Port	$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB
4-Kbyte expansion mode	Address/data	Address				Port				Port	$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB
16-Kbyte expansion mode	Address/data	Address						Port		Port	$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB
Full-address mode	Address/data	Address								Port	$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB

Caution When the external wait function is not used, the $\overline{\text{WAIT}}$ pin can be used as a port in all modes.

Memory maps when using the external device expansion function are as follows.

Figure 22-1. Memory Map when Using External Device Expansion Function (1/3)

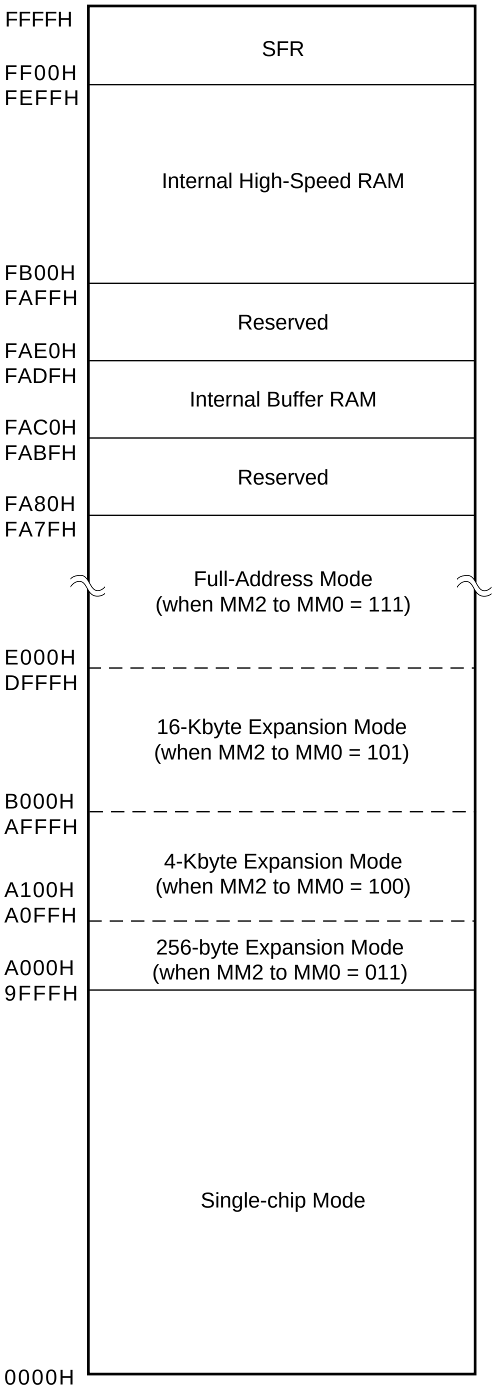


(a) Memory map of μPD780053 and 780053Y, and μPD780058, 780058Y, 78F0058, and 78F0058Y with internal ROM (flash memory) set to 24 KB

(b) Memory map of μPD780054 and 780054Y, and μPD780058, 780058Y, 78F0058, and 78F0058Y with internal ROM (flash memory) set to 32 KB

Figure 22-1. Memory Map when Using External Device Expansion Function (2/3)

(c) Memory map of μPD780055 and 780055Y, and μPD780058, 780058Y, 78F0058, and 78F0058Y with internal ROM (flash memory) set to 40 KB



(d) Memory map of μPD780056 and 780056Y, and μPD780058, 780058Y, 78F0058, and 78F0058Y with internal ROM (flash memory) set to 48 KB

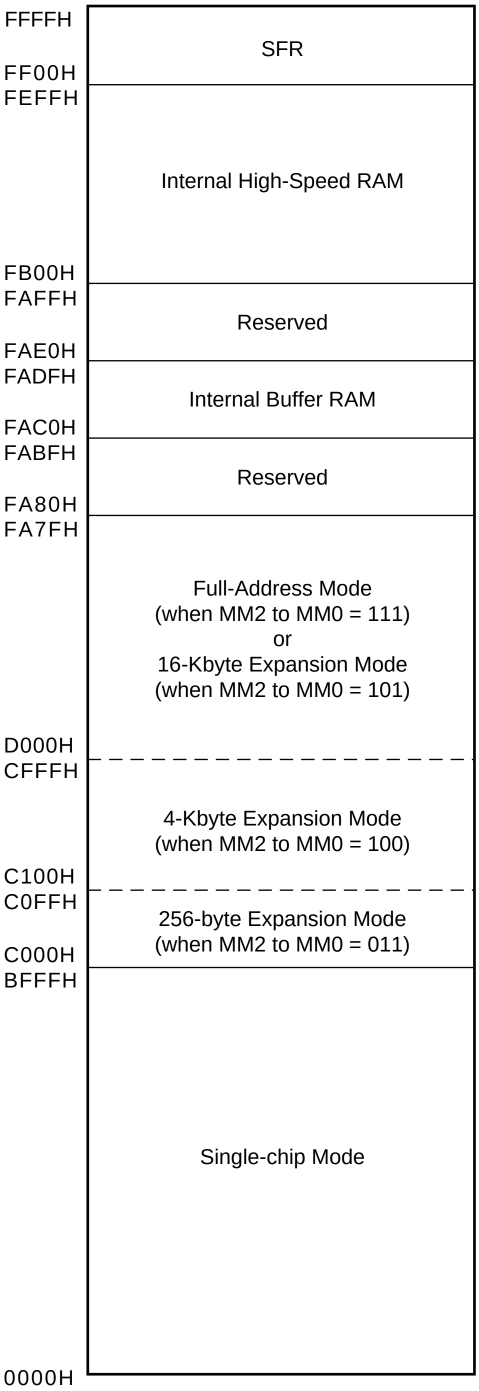
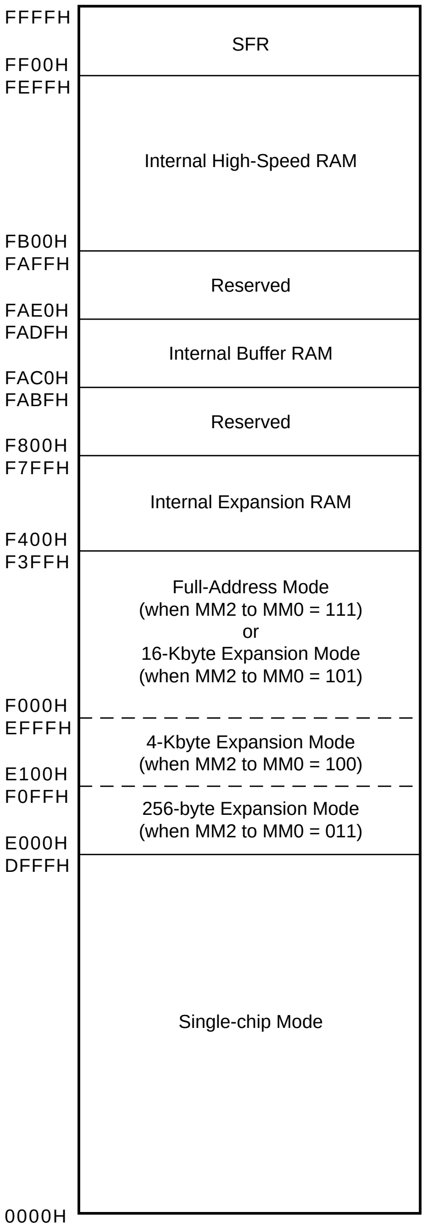
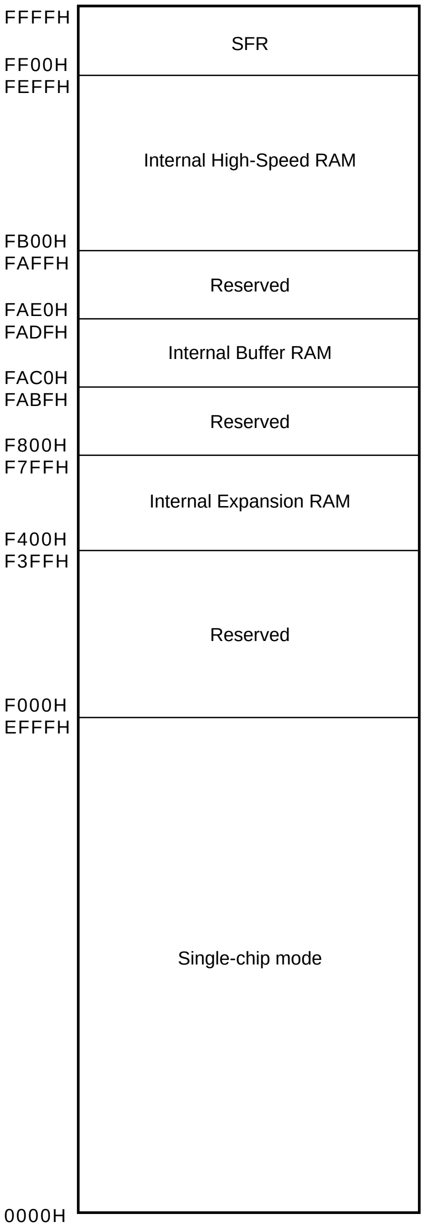


Figure 22-1. Memory Map when Using External Device Expansion Function (3/3)

(e) μPD780058, 780058Y, 78F0058, 78F0058Y
Memory map when internal ROM (flash memory) size is 56 Kbytes



(f) μPD780058, 780058Y, 78F0058, 78F0058Y
Memory map when internal ROM (flash memory) size is 60 Kbytes



Caution When the internal ROM (flash memory) size is 60 Kbytes, the area from F000H to F3FFH cannot be used. F000H to F3FFH can be used as external memory by setting the internal ROM (flash memory) size to less than 56 Kbytes by the memory size switching register (IMS).

22.2 External Device Expansion Function Control Register

The external device expansion function is controlled by the memory expansion mode register (MM) and memory size switching register (IMS).

(1) Memory expansion mode register (MM)

MM sets the wait count and external expansion area, and also sets the input/output of port 4.
MM is set with a 1-bit memory or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets MM to 10H.

Figure 22-2. Memory Expansion Mode Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
MM	0	0	PW1	PW0	0	MM2	MM1	MM0	FFF8H	10H	R/W

MM2	MM1	MM0	Single-chip/ Memory Expansion Mode Selection		P40 to P47, P50 to P57, P64 to P67 Pin State					
					P40 to P47		P50 to P53	P54, P55	P56, P57	P64 to P67
0	0	0	Single-chip mode		Port mode	Input	Port mode			
0	0	1				Output				
0	1	1	Memory expansion mode	256-byte mode	AD0 to AD7		Port mode			
1	0	0		4-Kbyte mode			A8 to A11	Port mode		P64 = $\overline{\text{RD}}$ P65 = $\overline{\text{WR}}$ P66 = 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							

PW1	PW0	Wait Control
0	0	No wait
0	1	Wait (one wait state insertion)
1	0	Setting prohibited
1	1	Wait control by external wait pin

Note The full-address mode allows external expansion to the entire 64-Kbyte address space except for the internal ROM, RAM, and SFR areas and the reserved areas.

Remark P60 to P63 are used as port pins without regard to the mode (single-chip mode or memory expansion mode).

(2) Memory size switching register (IMS)

This register specifies the internal memory size. In principle, use IMS in a default status. However, when using the external device expansion function with the μPD780058, 780058Y, set the IMS so that the internal ROM capacity is 56 Kbytes or lower.

IMS is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IMS to the value indicated in Table 22-3.

Figure 22-3. Memory Size Switching Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0	FFF0H	**Note**	R/W

ROM3	ROM2	ROM1	ROM0	Internal ROM Size Selection
0	1	1	0	24 Kbytes
1	0	0	0	32 Kbytes
1	0	1	0	40 Kbytes
1	1	0	0	48 Kbytes
1	1	1	0	56 Kbytes
1	1	1	1	60 Kbytes
Other than above				Setting prohibited

RAM2	RAM1	RAM0	Internal High-Speed RAM Size Selection
1	1	0	1,024 bytes
Other than above			Setting prohibited

Note The values after reset depend on the product. (see **Table 22-3**)

Table 22-3. Values After the Memory Size Switching Register Is Reset

Part Number	Reset Value
μPD780053, 780053Y	C6H
μPD780054, 780054Y	C8H
μPD780055, 780055Y	CAH
μPD780056, 780056Y	CCH
μPD780058, 780058Y	CFH

22.3 External Device Expansion Function Timing

Timing control signal output pins in the external memory expansion mode are as follows.

(1) $\overline{\text{RD}}$ pin (Alternate function: P64)

Read strobe signal output pin. The read strobe signal is output in data accesses and instruction fetches from external memory.

During internal memory access, the read strobe signal is not output (maintains high level).

(2) $\overline{\text{WR}}$ pin (Alternate function: P65)

Write strobe signal output pin. The write strobe signal is output in data access to external memory.

During internal memory access, the write strobe signal is not output (maintains high level).

(3) $\overline{\text{WAIT}}$ pin (Alternate function: P66)

External wait signal input pin. When the external wait is not used, the $\overline{\text{WAIT}}$ pin can be used as an input/output port.

During internal memory access, the external wait signal is ignored.

(4) ASTB pin (Alternate function: P67)

Address strobe signal output pin. Timing signal is output without regard to the data accesses and instruction fetches from external memory. The ASTB signal is also output when the internal memory is accessed.

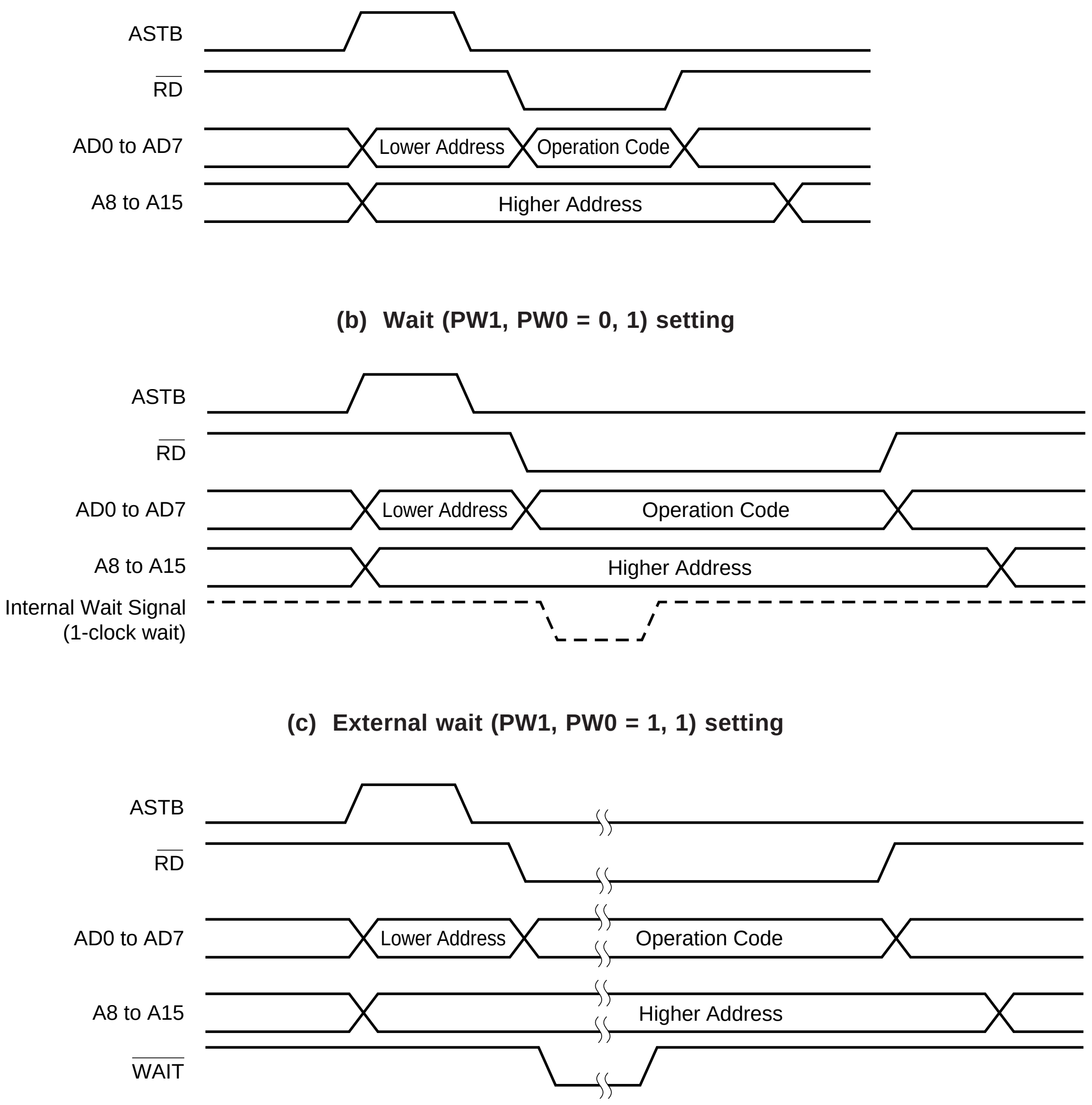
(5) AD0 to AD7, A8 to A15 pins (Alternate function: P40 to P47, P50 to P57)

Address/data signal output pin. Valid signal is output or input during data accesses and instruction fetches from external memory.

These signals change when the internal memory is accessed (output values are undefined).

Timing charts are shown in Figures 22-4 to 22-7.

Figure 22-4. Instruction Fetch from External Memory



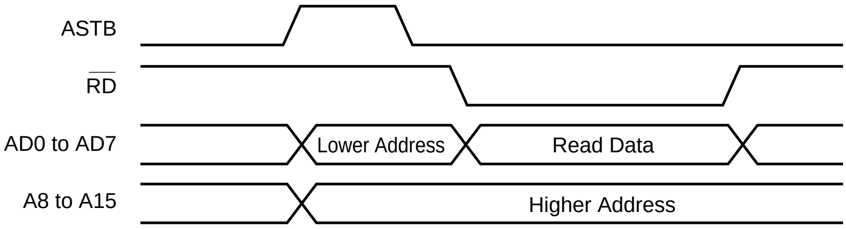
(a) No wait (PW1, PW0 = 0, 0) setting

(b) Wait (PW1, PW0 = 0, 1) setting

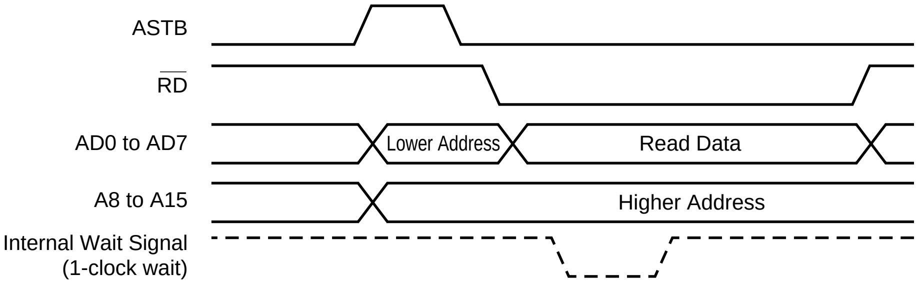
(c) External wait (PW1, PW0 = 1, 1) setting

Figure 22-5. 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

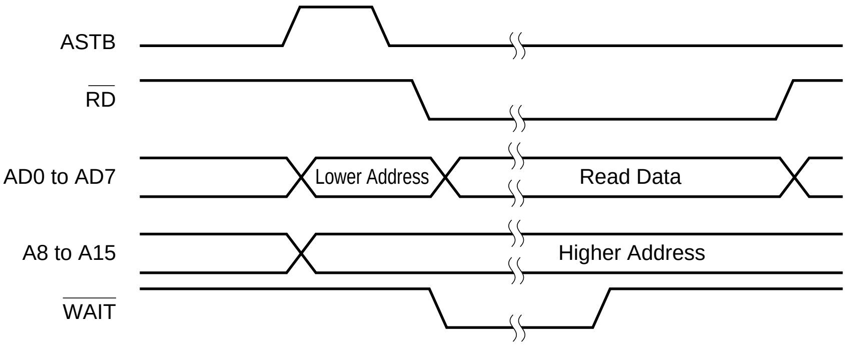
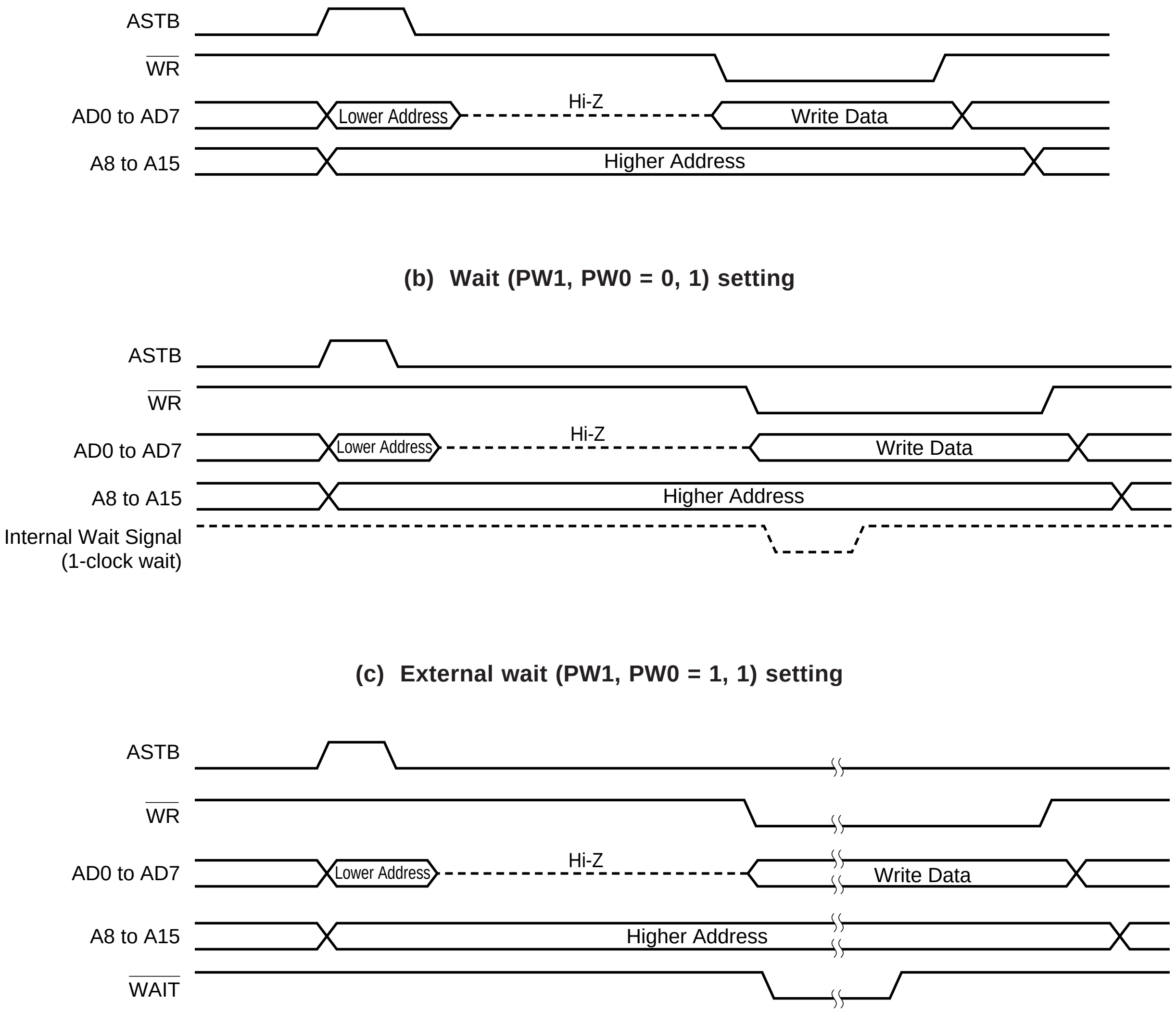


Figure 22-6. External Memory Write Timing



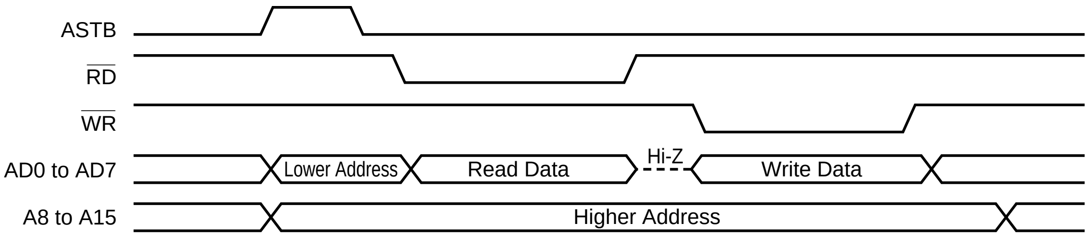
(a) No wait (PW1, PW0 = 0, 0) setting

(b) Wait (PW1, PW0 = 0, 1) setting

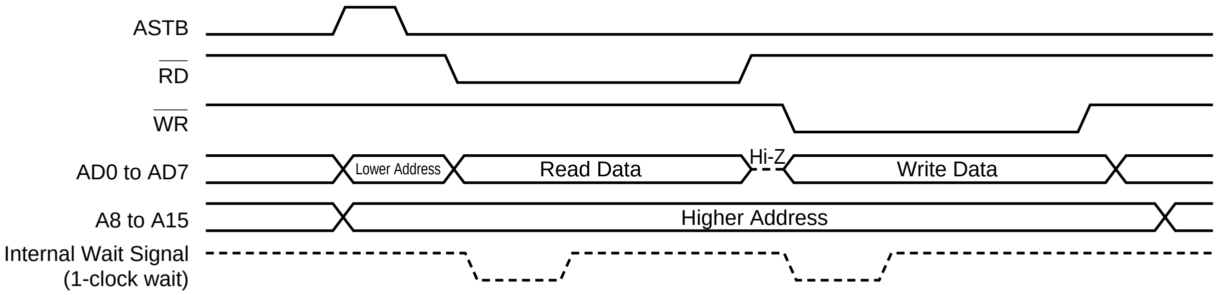
(c) External wait (PW1, PW0 = 1, 1) setting

Figure 22-7. External Memory Read Modify Write Timing

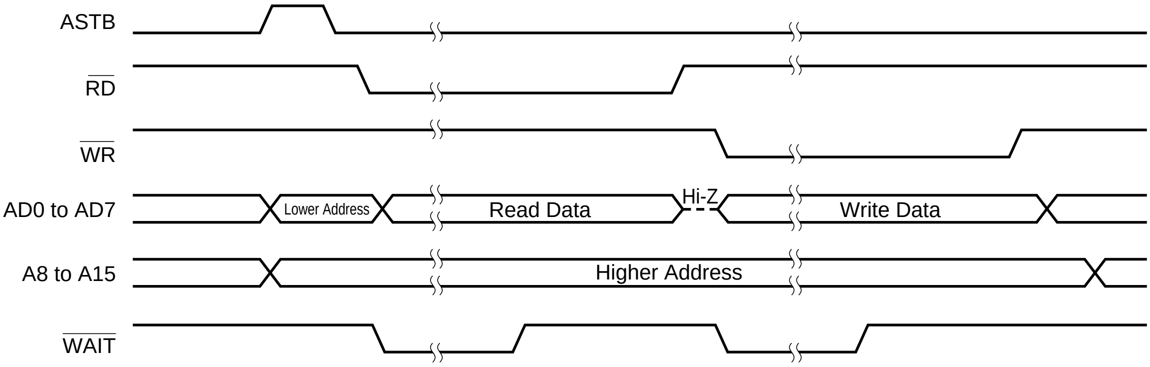
(a) No wait (PW1, PW0 = 0, 0) setting



(b) Wait (PW1, PW0 = 0, 1) setting



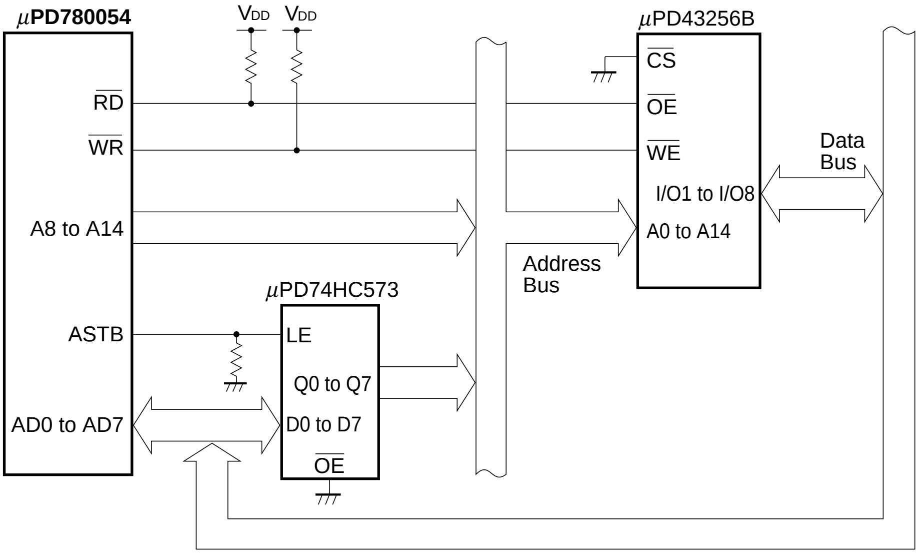
(c) External wait (PW1, PW0 = 1, 1) setting



22.4 Example of Connection with Memory

This section provides μPD780054 and external memory connection examples in Figure 22-8. SRAMs are used as the external memory in these diagrams. In addition, the external device expansion function is used in the full-address mode, and the address from 0000H to 7FFFH (32 Kbytes) are allocated for internal ROM, and the addresses after 8000H for SRAM.

Figure 22-8. Connection Example of μPD780054 and Memory



CHAPTER 23 STANDBY FUNCTION

23.1 Standby Function and Configuration

23.1.1 Standby function

The standby function is designed to decrease power consumption of the system. The following two modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. The HALT mode is intended to stop the CPU operation clock. System clock oscillator continues oscillation. In this mode, current consumption cannot be decreased as in the STOP mode. The HALT mode is valid to restart immediately upon interrupt request and to carry out intermittent operations such as in watch applications.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the main system clock oscillator stops and the whole system stops. CPU current consumption can be considerably decreased.

Data memory low-voltage hold (down to V_{DD} = 1.8 V) is possible. Thus, the STOP mode is effective to hold data memory contents with ultra-low current consumption. Because this mode can be cleared upon interrupt request, it enables intermittent operations to be carried out.

However, because a wait time is necessary to secure an oscillation stabilization time after the STOP mode is cleared, select the HALT mode if it is necessary to start processing immediately upon interrupt request.

In any mode, all the contents of the register, flag and data memory just before standby mode setting are held. The input/output port output latch and output buffer statuses are also held.

Cautions 1. The STOP mode can be used only when the system operates with the main system clock (subsystem clock oscillation cannot be stopped). The HALT mode can be used with either the main system clock or the subsystem clock.

2. When proceeding to the STOP mode, be sure to stop the peripheral hardware operation and execute the STOP instruction.

3. The following sequence is recommended for power consumption reduction of the A/D converter when the standby function is used: first clear the bit 7 (CS) of A/D converter mode register (ADM) to 0 to stop the A/D conversion operation, and then execute the HALT or STOP instruction.

23.1.2 Standby function control register

A wait time after the STOP mode is cleared upon interrupt request till the oscillation stabilizes is controlled with the oscillation stabilization time select register (OSTS).

OSTS is set with an 8-bit memory manipulation instruction.

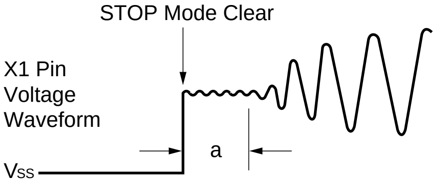
$\overline{\text{RESET}}$ input sets OSTS to 04H. However, it takes $2^{17}/f_X$, not $2^{18}/f_X$, until the STOP mode is cleared by $\overline{\text{RESET}}$ input.

Figure 23-1. Oscillation Stabilization Time Selection Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFAH	04H	R/W

OSTS2	OSTS1	OSTS0	Selection of Oscillation Stabilization Time when STOP Mode is Released		
				MCS = 1	MCS = 0
0	0	0	$2^{12}/f_{XX}$	$2^{12}/f_X$ (819 μs)	$2^{13}/f_X$ (1.64 ms)
0	0	1	$2^{14}/f_{XX}$	$2^{14}/f_X$ (3.28 ms)	$2^{15}/f_X$ (6.55 ms)
0	1	0	$2^{15}/f_{XX}$	$2^{15}/f_X$ (6.55 ms)	$2^{16}/f_X$ (13.1 ms)
0	1	1	$2^{16}/f_{XX}$	$2^{16}/f_X$ (13.1 ms)	$2^{17}/f_X$ (26.2 ms)
1	0	0	$2^{17}/f_{XX}$	$2^{17}/f_X$ (26.2 ms)	$2^{18}/f_X$ (52.4 ms)
Other than above			Setting prohibited		

Caution The wait time that elapses when the STOP mode has been released does not include the time required for the clock to start oscillation (see "a" in the illustration below) after the STOP mode has been released. The same applies when the STOP mode has been released by $\overline{\text{RESET}}$ input and by generation of an interrupt request.



Remarks 1. f_{XX} : Main system clock frequency (f_X or $f_X/2$)

2. f_X : Main system clock oscillation frequency

3. MCS : Bit 0 of oscillation mode select register (OSMS)

4. Figures in parentheses apply to operation with f_X = 5.0 MHz

23.2 Standby Function Operations

23.2.1 HALT mode

(1) HALT mode set and operating status

The HALT mode is set by executing the HALT instruction. It can be set with the main system clock or the subsystem clock.

The operating status in the HALT mode is described below.

Table 23-1. HALT Mode Operating Status

Setting of HALT Mode Item		On Execution of HALT Instruction during Main System Clock Operation		On Execution of HALT Instruction during Subsystem Clock Operation	
		Without subsystem clock Note 1	With subsystem clock Note 2	When main system clock continues oscillation	When main system clock stops oscillation
Clock generator		Both main system and subsystem clocks can be oscillated. Clock supply to the CPU stops			
CPU		Operation stops			
Port (output latch)		Status before HALT mode setting is held			
16-bit timer/event counter		Operable			Operable when watch timer output is selected as count clock (f_{XT} is selected as count clock of watch timer) or when TI00 is selected
8-bit timer/event counter		Operable			Operable when TI1 or TI2 is selected as count clock
Watch timer		Operable when $f_{XX}/2^7$ is selected as count clock	Operable		Operable when f_{XT} is selected as count clock
Watchdog timer		Operable		Operation stops	
A/D converter		Operable			Operation stops
D/A converter		Operable			
Real-time output port		Operable			
Serial interface	Other than automatic transmit/receive function	Operable			Operable when external $\overline{SCK}$ is used
	Automatic transmit/receive function	Operation stops			
External interrupt request	INTP0	INTP0 is operable when clock supplied for peripheral hardware is selected as sampling clock ($f_{XX}/2^5$, $f_{XX}/2^6$, $f_{XX}/2^7$)			Operation stops
	INTP1 to INTP5	Operable			
Bus line for external expansion	AD0 to AD7	High impedance			
	A0 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 when external clock is not supplied

2. Including when external clock is supplied

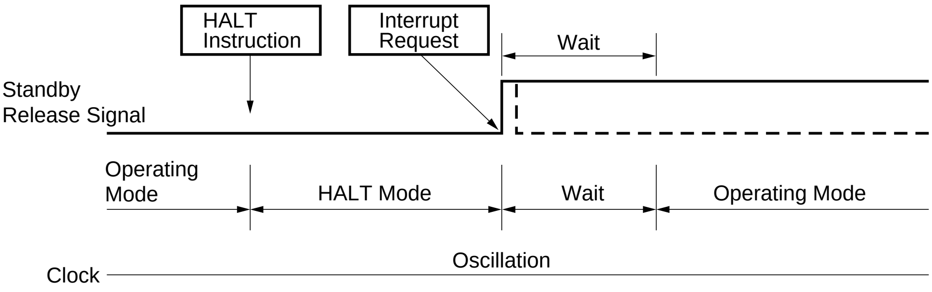
(2) HALT mode release

The HALT mode can be released with the following four types of sources.

(a) Release by unmasked interrupt request

If an unmasked interrupt request is generated, the HALT mode is released. If interrupt request acknowledgement is enabled, vectored interrupt service is carried out. If disabled, the next address instruction is executed.

Figure 23-2. HALT Mode Release by Interrupt Request Generation



Remarks 1. The broken line indicates the case when the interrupt request which has released the standby status is acknowledged.

2. Wait time will be as follows:

- When the program branches to vector table : 8 to 9 clocks
- When the program does not branch to vector table : 2 to 3 clocks

(b) Release by non-maskable interrupt request generation

If a non-maskable interrupt request is generated, the HALT mode is released and vectored interrupt service is carried out irrespective of whether interrupt request acknowledgement is enabled or disabled.

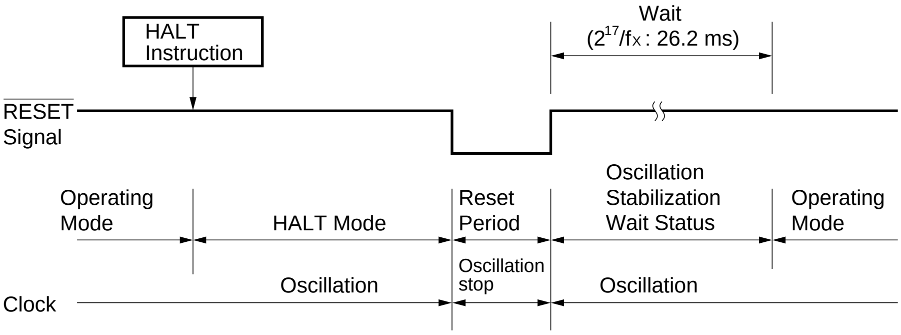
(c) Release by unmasked test input

If an unmasked test signal is input, the HALT mode is released, and the next address instruction of the HALT instruction is executed.

(d) Release by $\overline{\text{RESET}}$ input

If the $\overline{\text{RESET}}$ signal is input, the HALT mode is released. As is the case with normal reset operation, a program is executed after branch to the reset vector address.

Figure 23-3. HALT Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses apply to operation with f_X = 5.0 MHz.

Table 23-2. Operation After HALT Mode Release

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt service execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt service execution
	1	×	×	×	HALT mode hold
Non-maskable interrupt request	–	–	×	×	Interrupt service execution
Test input	0	–	×	×	Next address instruction execution
	1	–	×	×	HALT mode hold
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

Remark ×: don't care

23.2.2 STOP mode

(1) STOP mode set and operating status

The STOP mode is set by executing the STOP instruction. It can be set only with the main system clock.

Cautions 1. When the STOP mode is set, the X2 pin is internally connected to V_{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 below.

Table 23-3. STOP Mode Operating Status

Setting of STOP Mode Item		With Subsystem Clock	Without Subsystem Clock
Clock generator		Only main system clock stops oscillation	
CPU		Operation stops	
Port (output latch)		Status before STOP mode setting is held	
16-bit timer/event counter		Operable when watch timer output is selected as count clock (f_{XT} is selected as count clock of watch timer)	Operation stops
8-bit timer/event counter		Operable when TI1 and TI2 are selected for the count clock	
Watch timer		Operable when f_{XT} is selected for the count clock	Operation stops
Watchdog timer		Operation stops	
A/D converter			
D/A converter		Operable	
Real-time output port		Operable when external trigger is used or TI1 and TI2 are selected for the 8-bit timer/event counter count clock	
Serial interface	Other than automatic transmit/receive function and UART	Operable when externally supplied clock is specified as the serial clock	
	Automatic transmit/receive function and UART	Operation stops	
External interrupt request	INTP0	Not operable	
	INTP1 to INTP5	Operable	
Bus line for external expansion	AD0 to AD7	High impedance	
	A0 to A15	Status before STOP mode setting is held	
	ASTB	Low level	
	$\overline{WR}$, $\overline{RD}$	High level	
	$\overline{WAIT}$	High impedance	

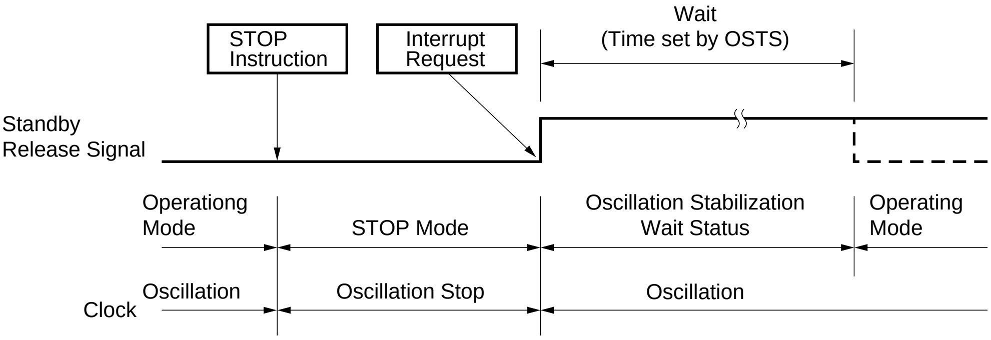
(2) STOP mode release

The STOP mode can be released with the following three types of sources.

(a) Release by unmasked interrupt request

If an unmasked interrupt request is generated, the STOP mode is released. If interrupt request acknowledgement is enabled after the lapse of oscillation stabilization time, vectored interrupt service is carried out. If interrupt request acknowledgement is disabled, the next address instruction is executed.

Figure 23-4. STOP Mode Release by Interrupt Request Generation



Remark The broken line indicates the case when the interrupt request which has released the standby status is acknowledged.

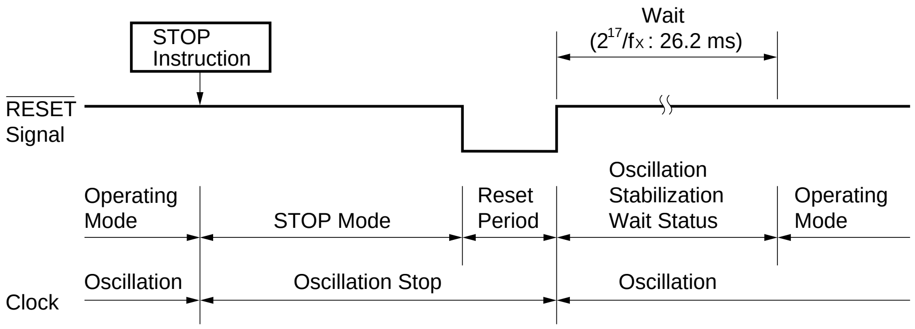
(b) Release by unmasked test input

If an unmasked test signal is input, the STOP mode is released. After the lapse of oscillation stabilization time, the instruction at the next address of the STOP instruction is executed.

(c) Release by $\overline{\text{RESET}}$ input

If the $\overline{\text{RESET}}$ signal is input, the STOP mode is released and after the lapse of oscillation stabilization time, reset operation is carried out.

Figure 23-5. STOP Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses apply to operation with f_X = 5.0 MHz.

Table 23-4. Operation After STOP Mode Release

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt service execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt service execution
	1	×	×	×	STOP mode hold
Test input	0	–	×	×	Next address instruction execution
	1	–	×	×	STOP mode hold
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

Remark ×: don't care

CHAPTER 24 RESET FUNCTION

24.1 Reset Function

The following two operations are available to generate the reset signal.

(1) External reset input by $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer overrun time detection

External reset and internal reset have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H by $\overline{\text{RESET}}$ input.

When a low level is input to the $\overline{\text{RESET}}$ pin or the watchdog timer overflows, a reset is applied and each hardware is set to the status as shown in Table 24-1. Each pin has high impedance during reset input or during oscillation stabilization time just after reset clear.

When a high level is input to the $\overline{\text{RESET}}$ input, the reset is cleared and program execution starts after the lapse of oscillation stabilization time ($2^{17}/f_X$). The reset applied by watchdog timer overflow is automatically cleared after a reset and program execution starts after the lapse of oscillation stabilization time ($2^{17}/f_X$) (see **Figures 24-2** to **24-4**).

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 24-1. Reset Function Block Diagram

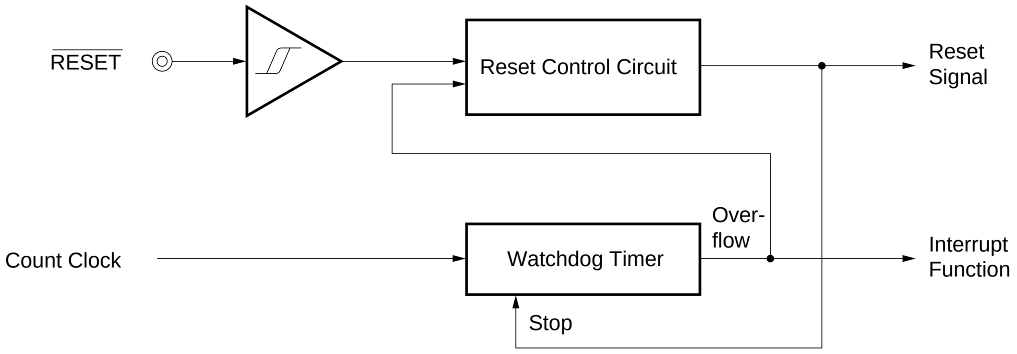


Figure 24-2. Reset Timing by $\overline{\text{RESET}}$ Input

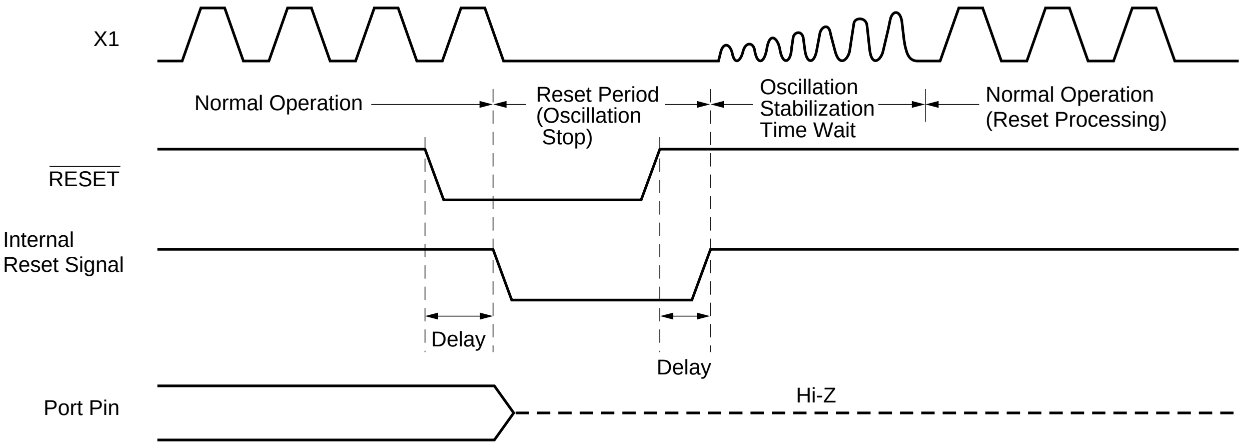

X1
Normal Operation
Reset Period (Oscillation Stop)
Oscillation Stabilization Time Wait
Normal Operation (Reset Processing)
RESET
Internal Reset Signal
Delay
Delay
Port Pin
Hi-Z

Figure 24-3. Reset Timing due to Watchdog Timer Overflow

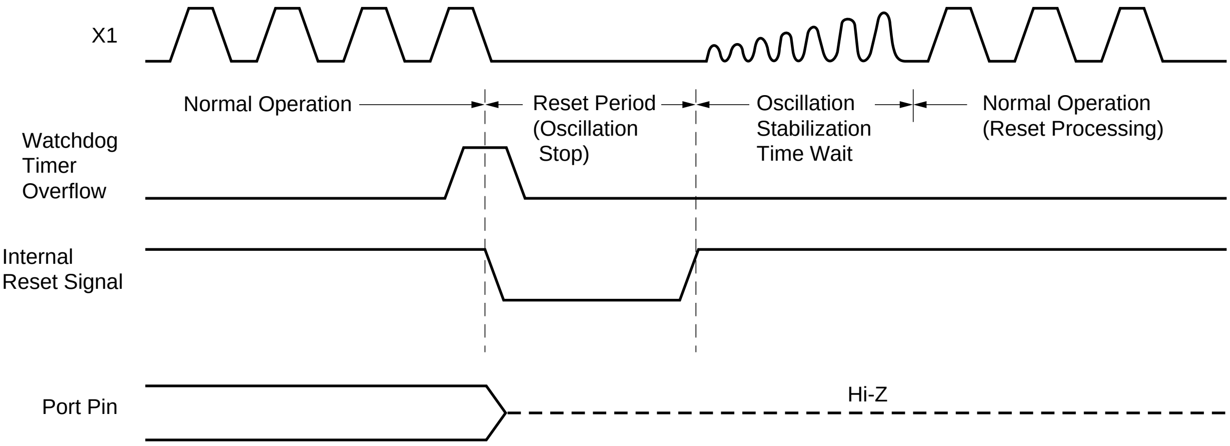

X1
Normal Operation
Reset Period (Oscillation Stop)
Oscillation Stabilization Time Wait
Normal Operation (Reset Processing)
Watchdog Timer Overflow
Internal Reset Signal
Port Pin
Hi-Z

Figure 24-4. Reset Timing by $\overline{\text{RESET}}$ Input in STOP Mode

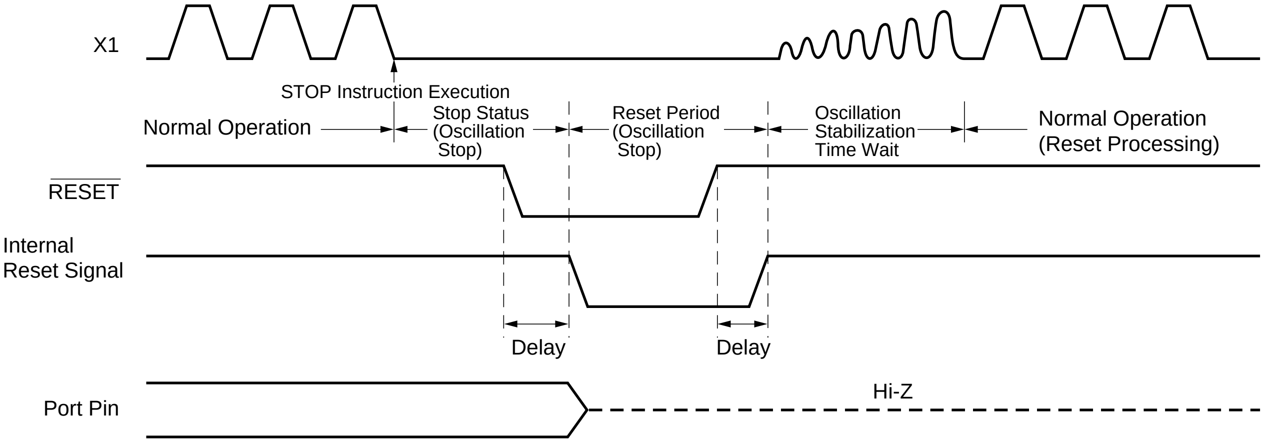

X1
STOP Instruction Execution
Normal Operation
Stop Status (Oscillation Stop)
Reset Period (Oscillation Stop)
Oscillation Stabilization Time Wait
Normal Operation (Reset Processing)
RESET
Internal Reset Signal
Delay
Delay
Port Pin
Hi-Z

Table 24-1. Hardware Status After Reset (1/2)

Hardware		Status after Reset
Program counter (PC) Note 1		The contents of reset vector tables (0000H and 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined Note 2
	General register	Undefined Note 2
Port (Output latch)	Ports 0 to 3, 7, 12, 13 (P0 to P3, P7, P12, P13)	00H
	Ports 4 to 6 (P4 to P6)	Undefined
Port mode register (PM0 to PM3, PM5 to PM7, PM12, PM13)		FFH
Pull-up resistor option register (PUOH, PUOL)		00H
Processor clock control register (PCC)		04H
Oscillation mode selection register (OSMS)		00H
Memory size switching register (IMS)		Note 3
Internal expansion RAM size switching register (IXS) Note 4		0AH
Memory expansion mode register (MM)		10H
Oscillation stabilization time select register (OSTS)		04H
16-bit timer/event counter	Timer register (TM0)	00H
	Capture/compare register (CR00, CR01)	Undefined
	Clock selection register (TCL0)	00H
	Mode control register (TMC0)	00H
	Capture/compare control register 0 (CRC0)	04H
	Output control register (TOC0)	00H
8-bit timer/event counter 1 and 2	Timer register (TM1, TM2)	00H
	Compare registers (CR10, CR20)	Undefined
	Clock selection register (TCL1)	00H
	Mode control registers (TMC1)	00H
	Output control register (TOC1)	00H

Notes 1. During reset input or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remains unchanged after reset.

2. If the reset signal is input in the standby mode, the status before reset is retained even after reset.

3. The values after reset depend on the product.
μPD780053, 780053Y : C6H, μPD780054, 780054Y : C8H, μPD780055, 780055Y : CAH,
μPD780056, 780056Y : CCH, μPD780058, 780058Y : CFH, μPD78F0058, 78F0058Y: CFH

4. Provided only in the μPD780058, 780058Y, 78F0058, and 78F0058Y.

Table 24-1. Hardware Status after Reset (2/2)

Hardware		Status after Reset
Watch timer	Mode control register (TMC2)	00H
Watchdog timer	Clock select register (TCL2)	00H
	Mode register (WDTM)	00H
Serial interface	Clock select register (TCL3)	88H
	Shift registers (SIO0, SIO1)	Undefined
	Mode registers (CSIM0, CSIM1, CSIM2)	00H
	Serial bus interface control register (SBIC)	00H
	Slave address register (SVA)	Undefined
	Automatic data transmit/receive control register (ADTC)	00H
	Automatic data transmit/receive address pointer (ADTP)	00H
	Automatic data transmit/receive interval specify register (ADTI)	00H
	Asynchronous serial interface mode register (ASIM)	00H
	Asynchronous serial interface status register (ASIS)	00H
	Baud rate generator control register (BRGC)	00H
	Serial interface pin select register (SIPS)	00H
	Transmit shift register (TXS)	FFH
	Receive buffer register (RXB)	
	Interrupt timing specify register (SINT)	00H
A/D converter	Mode register (ADM)	01H
	Conversion result register (ADCR)	Undefined
	Input select register (ADIS)	00H
D/A converter	Mode register (DAM)	00H
	Conversion value setting register (DACS0, DACS1)	00H
Real-time output port	Mode register (RTPM)	00H
	Control register (RTPC)	00H
	Buffer register (RTBL, RTBH)	00H
ROM correction [Note]	Correction address register (CORAD0, CORAD1)	0000H
	Correction control register (CORCN)	00H
Interrupt	Request flag register (IF0L, IF0H, IF1L)	00H
	Mask flag register (MK0L, MK0H, MK1L)	FFH
	Priority specify flag register (PR0L, PR0H, PR1L)	FFH
	External interrupt mode register (INTM0, INTM1)	00H
	Key return mode register (KRM)	02H
	Sampling clock select register (SCS)	00H

Note Provided only in the μPD780058, 780058Y, 78F0058, 78F0058Y.

CHAPTER 25 ROM CORRECTION

25.1 ROM Correction Functions

The μPD780058, 780058Y Subseries can replace part of a program in the mask ROM with a program in the internal expansion RAM.

Instruction bugs found in the mask ROM can be avoided, and program flow can be changed by using the ROM correction.

The ROM correction can correct two places (max.) of the internal ROM (program).

★ **Caution The ROM correction cannot be emulated by the in-circuit emulator (IE-78000-R, IE-78000-R-A, IE-78K0-NS, IE-78001-R-A).**

25.2 ROM Correction Configuration

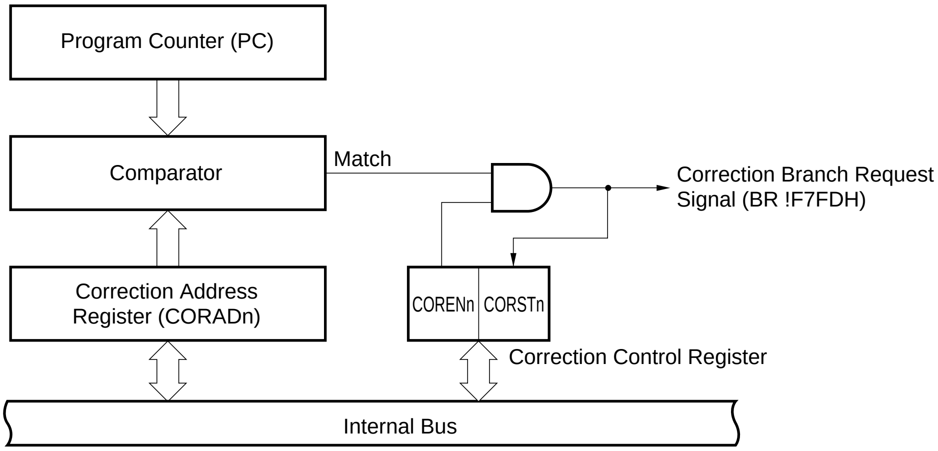
The ROM correction is executed by the following hardware.

Table 25-1. ROM Correction Configuration

Item	Configuration
Register	Correction address registers 0 and 1 (CORAD0, CORAD1)
Control register	Correction control register (CORCN)

Figure 25-1 shows a block diagram of the ROM correction.

Figure 25-1. ROM Correction Block Diagram


Program Counter (PC)
Comparator
Match
Correction Branch Request Signal (BR !F7FDH)
Correction Address Register (CORADn)
CORENn
CORSTn
Correction Control Register
Internal Bus

Remark n = 0, 1

(1) Correction address registers 0 and 1 (CORAD0, CORAD1)

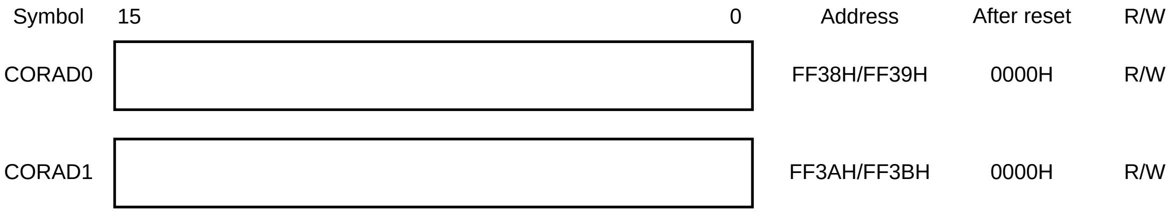
These registers set the start address (correction address) of the instruction(s) to be corrected in the mask ROM.

The ROM correction corrects two places (max.) of the program. Addresses are set to two registers, CORAD0 and CORAD1. If only one place needs to be corrected, set the address to either of the registers.

CORAD0 and CORAD1 are set with a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CORAD0 and CORAD1 to 0000H.

Figure 25-2. Correction Address Registers 0 and 1 Format



Symbol	15 … 0	Address	After reset	R/W
CORAD0		FF38H/FF39H	0000H	R/W
CORAD1		FF3AH/FF3BH	0000H	R/W

Cautions 1. Set the CORAD0 and CORAD1 when bit 1 (COREN0) and bit 3 (COREN1) of the correction control register (CORCN : see Figure 25-3) are 0.

2. Only addresses where operation codes are stored can be set in CORAD0 and CORAD1.

3. Do not set the following addresses to CORAD0 and CORAD1.

- **Address value in table area of table reference instruction (CALLT instruction) : 0040H to 007FH**
- **Address value in vector table area : 0000H to 003FH**

(2) Comparator

The comparator always compares the correction address value set in correction address registers 0 and 1 (CORAD0, CORAD1) with the fetch address value. When bit 1 (COREN0) or bit 3 (COREN1) of the correction control register (CORCN) is 1 and the correction address matches the fetch address value, the correction branch request signal (BR !F7FDH) is generated from the ROM correction circuit.

25.3 ROM Correction Control Registers

The ROM correction is controlled with the correction control register (CORCN).

(1) Correction control register (CORCN)

This register controls whether or not the correction branch request signal is generated when the fetch address matches the correction address set in correction address registers 0 and 1. The correction control register consists of correction enable flags (COREN0, COREN1) and correction status flags (CORST0, CORST1). The correction enable flags enable or disable the comparator match detection signal, and correction status flags show the values are matched.

CORCN is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CORCN to 00H.

Figure 25-3. Correction Control Register Format

Symbol	7	6	5	4	③	②	①	⓪	Address	After reset	R/W
CORCN	0	0	0	0	COREN1	CORST1	COREN0	CORST0	FF8AH	00H	R/W Note

CORST0	Correction Address Register 0 and Fetch Address Match Detection
0	Not detected
1	Detected

COREN0	Correction Address Register 0 and Fetch Address Match Detection Control
0	Disabled
1	Enabled

CORST1	Correction Address Register 1 and Fetch Address Match Detection
0	Not detected
1	Detected

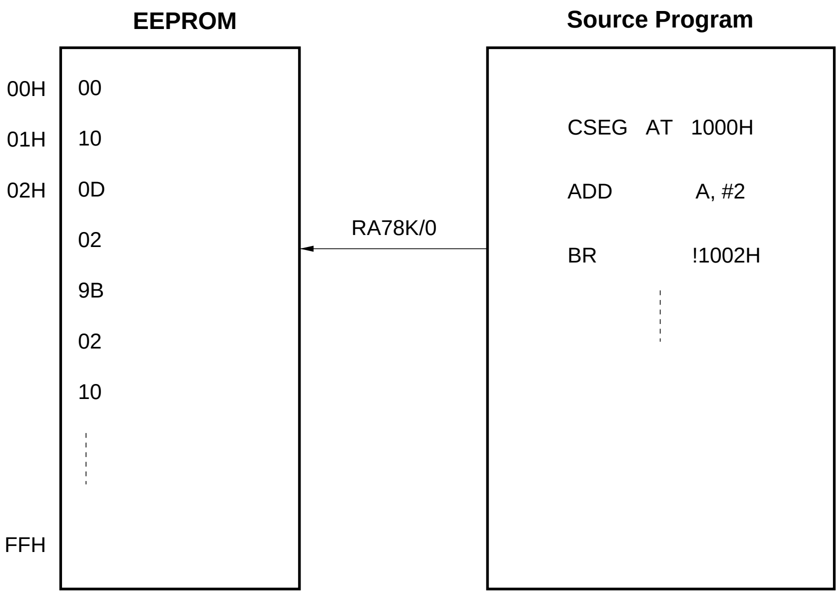
COREN1	Correction Address Register 1 and Fetch Address Match Detection Control
0	Disabled
1	Enabled

Note Bits 0 and 2 are read-only bits.

25.4 ROM Correction Application

(1) Store the correction address and instruction after correction (patch program) to nonvolatile memory (such as EEPROM^{TM}) outside the microcontroller.
When two places should be corrected, store the branch destination judgment program as well. The branch destination judgment program checks which one of the addresses set to correction address registers 0 and 1 (CORAD0 or CORAD1) generates the correction branch.

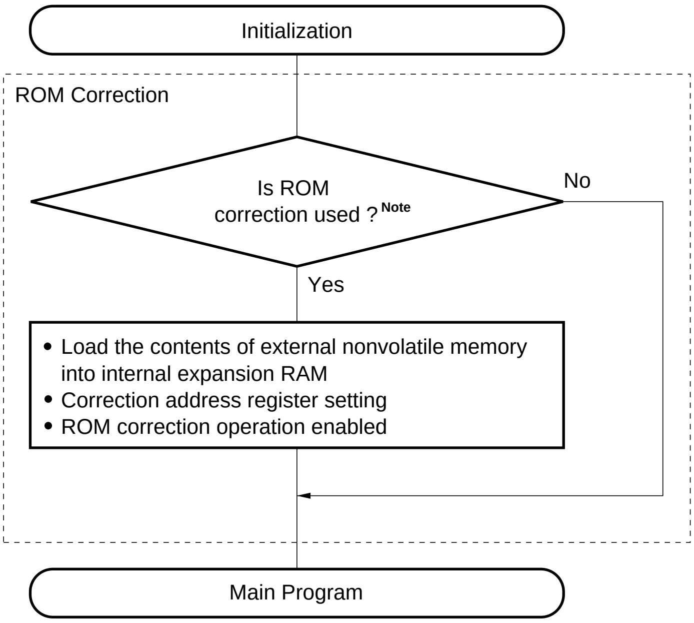
Figure 25-4. Storing Example to EEPROM (When One Place Is Corrected)



```
CSEG   AT   1000H
ADD          A, #2
BR           !1002H
```

(2) Assemble in advance the initialization routine as shown in Figure 25-5 to correct the program.

Figure 25-5. Initialization Routine


Initialization
ROM Correction
Is ROM correction used ? Note
No
Yes
• Load the contents of external nonvolatile memory into internal expansion RAM
• Correction address register setting
• ROM correction operation enabled
Main Program

Note Whether the ROM correction is used or not should be judged by the port input level. For example, when the P20 input level is high, the ROM correction is used, otherwise, it is not used.

(3) After reset, store the contents that have been previously stored in the external nonvolatile memory with initialization routine for ROM correction of the user to internal expansion RAM (see **Figure 25-5**).
Set the start address of the instruction to be corrected to CORAD0 and CORAD1, and set bits 1 and 3 (COREN0, COREN1) of the correction control register (CORCN) to 1.

(4) Set the entire-space branch instruction (BR !addr16) to the specified address (F7FDH) of the internal expansion RAM with the main program.

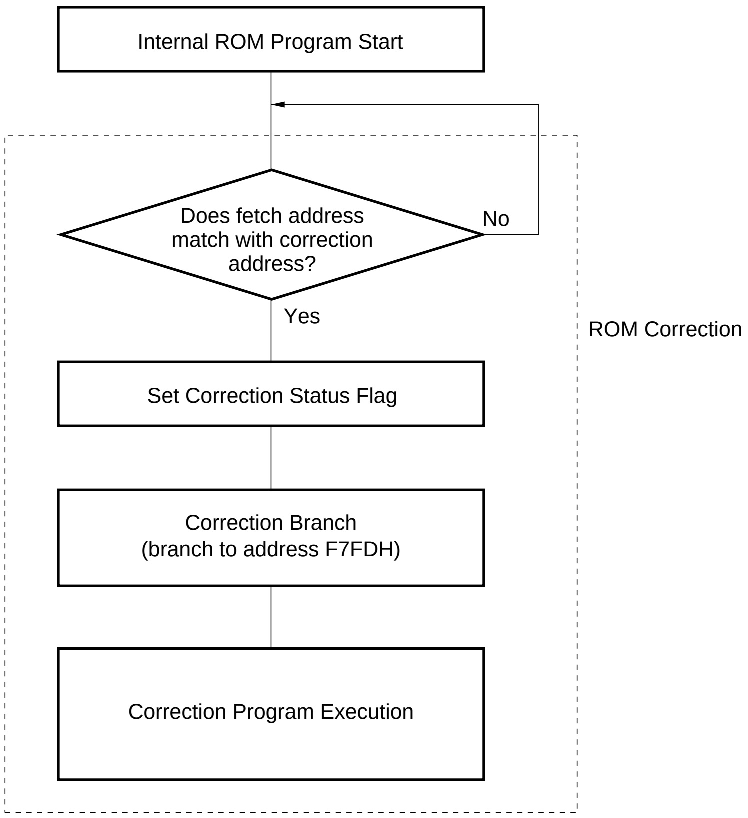
(5) After the main program is started, the fetch address value and the values set in CORAD0 and CORAD1 are always compared by the comparator in the ROM correction circuit. When these values match, the correction branch request signal is generated. Simultaneously the corresponding correction status flag (CORST0 or CORST1) is set to 1.

(6) Branch to the address F7FDH by the correction branch request signal.

(7) Branch to the internal expansion RAM address set with the main program by the entire-space branch instruction of the address F7FDH.

(8) When one place is corrected, the correction program is executed. When two places are corrected, the correction status flag is checked with the branch destination judgment program, and branches to the correction program.

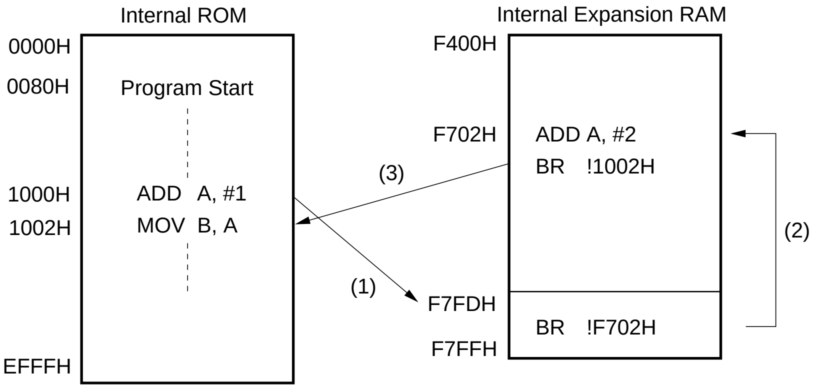
Figure 25-6. ROM Correction Operation


Internal ROM Program Start
Does fetch address match with correction address?
No
Yes
ROM Correction
Set Correction Status Flag
Correction Branch
(branch to address F7FDH)
Correction Program Execution

25.5 ROM Correction Usage Example

The example of ROM correction when the instruction at address 1000H “ADD A, #1” is changed to “ADD A, #2” is as follows.

Figure 25-7. ROM Correction Usage Example



(1) Branches to address F7FDH when the preset value 1000H in the correction address register matches the fetch address value after the main program is started.

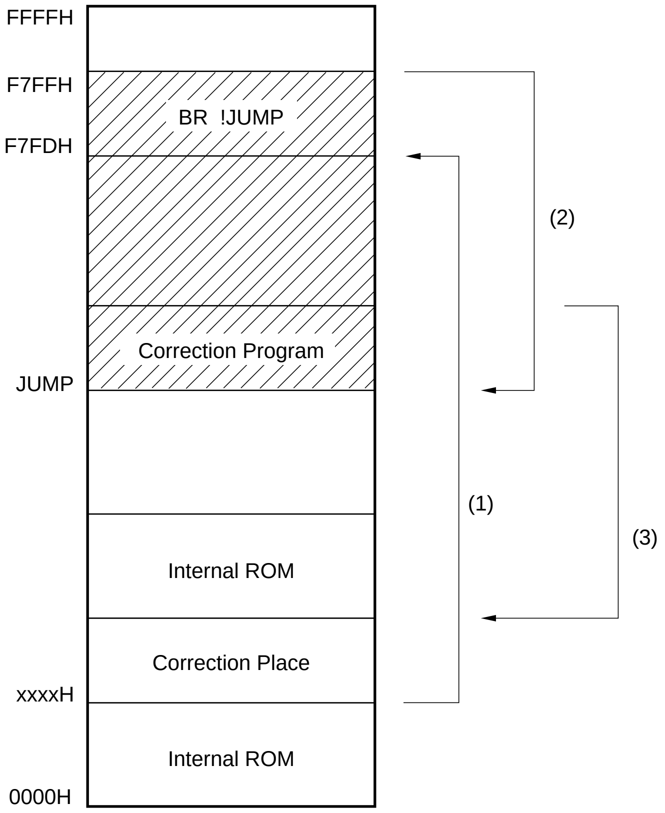
(2) Branches to any address (address F702H in this example) by setting the entire-space branch instruction (BR !addr16) to address F7FDH with the main program.

(3) Returns to the internal ROM program after executing the substitute instruction ADD A, #2.

25.6 Program Execution Flow

Figures 25-8 and 25-9 show the program transition diagrams when the ROM correction is used.

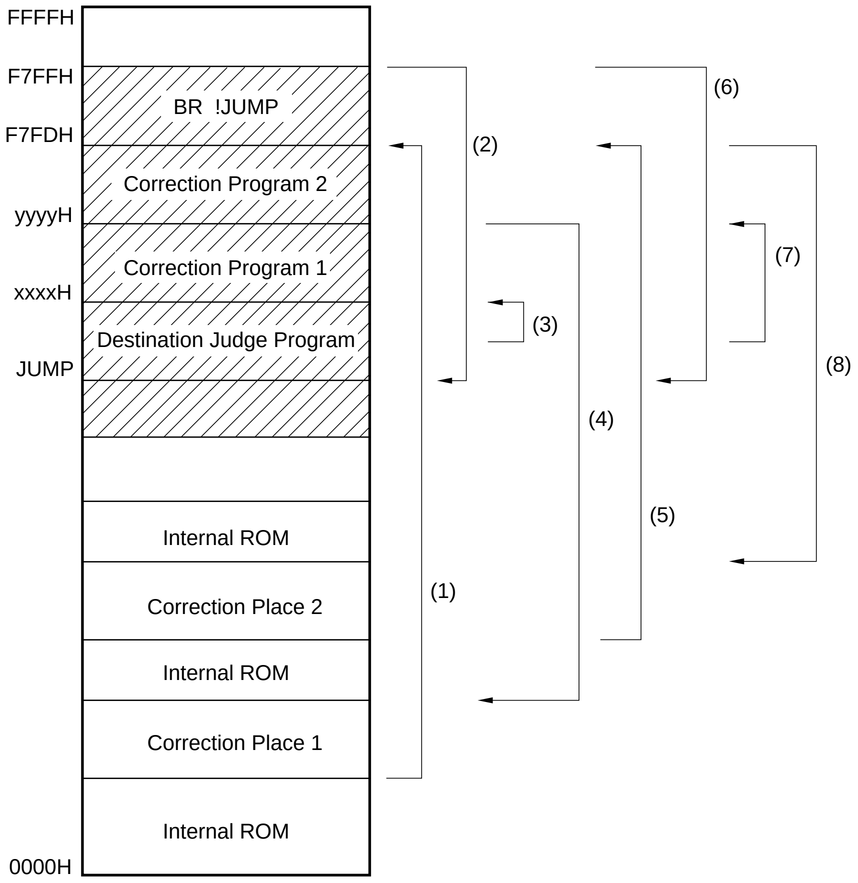
Figure 25-8. Program Transition Diagram (When One Place Is Corrected)



(1) Branches to address F7FDH when fetch address matches correction address
(2) Branches to correction program
(3) Returns to internal ROM program

Remark Area filled with diagonal lines : Internal expansion RAM
JUMP : Correction program start address

Figure 25-9. Program Transition Diagram (When Two Places Are Corrected)



(1) Branches to address F7FDH when fetch address matches correction address
(2) Branches to branch destination judgment program
(3) Branches to correction program 1 by branch destination judgment program (BTCLR !CORST0, $xxxxH)
(4) Returns to internal ROM program
(5) Branches to address F7FDH when fetch address matches correction address
(6) Branches to branch destination judgment program
(7) Branches to correction program 2 by branch destination judgment program (BTCLR !CORST1, $yyyyH)
(8) Returns to internal ROM program

Remark Area filled with diagonal lines : Internal expansion RAM
JUMP : Destination judge program start address

25.7 ROM Correction Cautions

(1) Address values set in correction address registers 0 and 1 (CORAD0, CORAD1) must be addresses where instruction codes are stored.

(2) Correction address registers 0 and 1 (CORAD0, CORAD1) should be set when the correction enable flag (COREN0, COREN1) is 0 (when the correction branch is in disabled state). If address is set to CORAD0 or CORAD1 when COREN0 or COREN1 is 1 (when the correction branch is in enabled state), the correction branch may start with the different address from the set address value.

(3) Do not set the address value of instruction immediately after the instruction that sets the correction enable flag (COREN0, COREN1) to 1, to correction address register 0 or 1 (CORAD0, CORAD1) ; the correction branch may not start.

(4) Do not set the address value in table area of table reference instruction (CALLT instruction) (0040H to 007FH), and the address value in vector table area (0000H to 003FH) to correction address registers 0 and 1 (CORAD0, CORAD1).

(5) Do not set two addresses immediately after the instructions shown below to correction address registers 0 and 1 (CORAD0, CORAD1). (that is, when the mapped terminal address of these instructions is N, do not set the address values of N + 1 and N + 2.)

- RET
- RETI
- RETB
- BR $addr16
- STOP
- HALT

CHAPTER 26 μPD78F0058, 78F0058Y

The μPD78F0058 and 78F0058Y have a flash memory whose contents can be written, erased, rewritten with the device mounted on a PC board. Table 26-1 lists the differences between the flash memory versions (μPD78F0058 and 78F0058Y) and the mask ROM versions (μPD780053, 780054, 780055, 780056, 780053Y, 780054Y, 780055Y, 780056Y, and 780058Y).

Table 26-1. Differences Between μPD78F0058, 78F0058Y and Mask ROM Versions

Item	μPD78F0058, 78F0058Y	Mask ROM Version
Internal ROM structure	Flash memory	Mask ROM
Internal ROM capacity	60 Kbytes	μPD780053, 780053Y: 24 Kbytes μPD780054, 780054Y: 32 Kbytes μPD780055, 780055Y: 40 Kbytes μPD780056, 780056Y: 48 Kbytes μPD780058, 780058Y: 60 Kbytes
Internal expansion RAM capacity	1,024 bytes	μPD780053, 780053Y: None μPD780054, 780054Y: None μPD780055, 780055Y: None μPD780056, 780056Y: None μPD780058, 780058Y: 1,024 bytes
Changing internal ROM capacities with memory size switching register (IMS)	Possible Note 1	Impossible
Changing of internal expansion RAM capacity by internal expansion RAM size switching register (IXS)	Possible Note 2	Impossible
IC pin	None	Available
V_{PP} pin	Available	None
Mask option for connecting internal pull-up resistor to P60 to P63 pins	No mask option available that connects pull-up resistor	Available

Notes 1. $\overline{\text{RESET}}$ input makes the flash memory 60 Kbytes.

2. $\overline{\text{RESET}}$ input makes the internal expansion RAM 1,024 bytes.

Caution There are differences in noise immunity and noise radiation between the flash memory and mask ROM versions. When pre-producing an application set with the flash memory version and then mass-producing it with the mask ROM version, be sure to conduct sufficient evaluations for the commercial samples (not engineering samples) of the mask ROM version.

Remark Only the μPD780058, 78F0058, 780058Y, 780058Y, and 78F0058Y are provided with an internal expansion RAM size switching register.

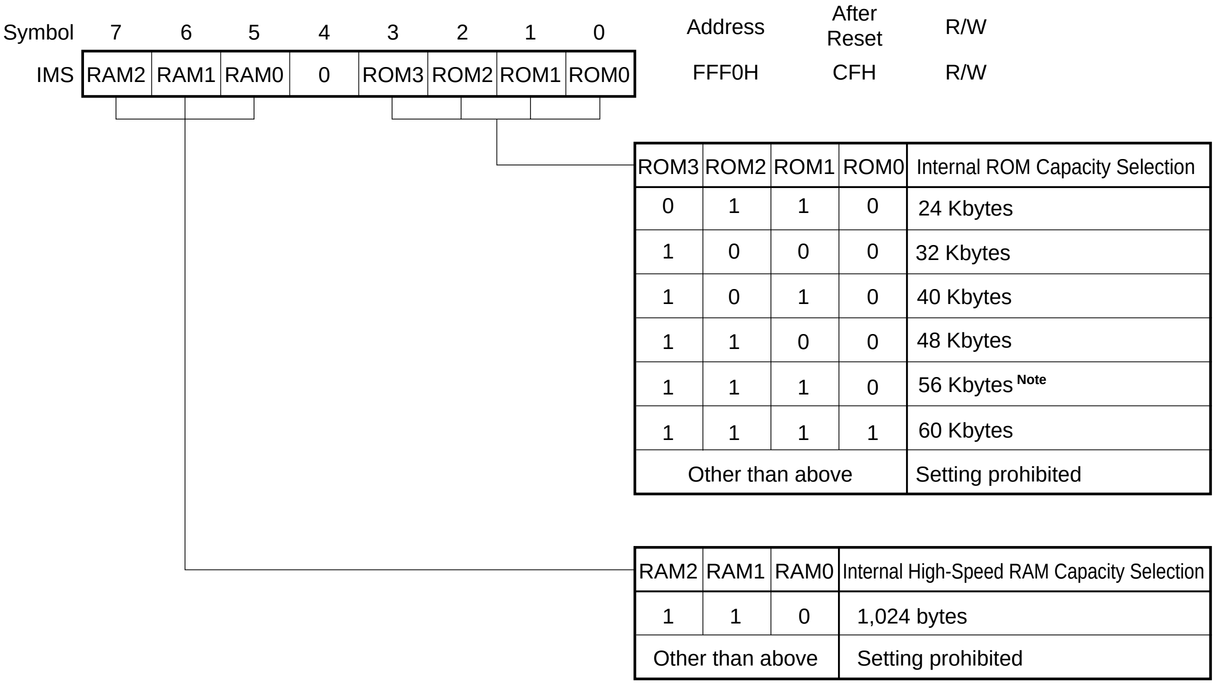
26.1 Memory Size Switching Register

The μPD78F0058 and 78F0058Y allow users to define its internal ROM size using the memory size switching register (IMS), so that the same memory mapping as that of a mask ROM version with a different-size internal ROM is possible.

IMS is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IMS to CFH.

Figure 26-1. Memory Size Switching Register Format



Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0	FFF0H	CFH	R/W

ROM3	ROM2	ROM1	ROM0	Internal ROM Capacity Selection
0	1	1	0	24 Kbytes
1	0	0	0	32 Kbytes
1	0	1	0	40 Kbytes
1	1	0	0	48 Kbytes
1	1	1	0	56 Kbytes Note
1	1	1	1	60 Kbytes
Other than above				Setting prohibited

RAM2	RAM1	RAM0	Internal High-Speed RAM Capacity Selection
1	1	0	1,024 bytes
Other than above			Setting prohibited

★ **Note** When using the external device expansion function of the μPD780058, 780058Y, 78F0058, and 78F0058Y, set the internal ROM capacity to 56 Kbytes or less.

The IMS settings to give the same memory map as mask ROM versions are shown in Table 26-2.

Table 26-2. Memory Size Switching Register Setting Value

Target Mask ROM Version	IMS Setting
μPD780053, 780053Y	C6H
μPD780054, 780054Y	C8H
μPD780055, 780055Y	CAH
μPD780056, 780056Y	CCH
μPD780058, 780058Y	CFH

26.2 Internal Expansion RAM Size Switching Register

The μPD78F0058 and 78F0058Y allow users to define its internal expansion RAM size by using the internal expansion RAM size switching register (IXS), so that the same memory mapping as that of a mask ROM version with a different-size internal expansion RAM is possible.

IXS is set with an 8-bit memory manipulation instruction.

RESET input sets IXS to 0AH.

Figure 26-2. Internal Expansion RAM Size Switching Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
IXS	0	0	0	0	IXRAM3	IXRAM2	IXRAM1	IXRAM0	FFF4H	0AH	W

IXRAM3	IXRAM2	IXRAM1	IXRAM0	Internal Extension RAM Capacity Selection
1	1	0	0	0 bytes
1	0	1	0	1,024 bytes
Other than above				Setting prohibited

The value in the IXS that has the identical memory map to the mask ROM versions is given in Table 26-3.

Table 26-3. Internal Expansion RAM Size Switching Register Setting Value

Target Mask ROM Version	IXS Setting
μPD780053, 780053Y	0CH
μPD780054, 780054Y	
μPD780055, 780055Y	
μPD780056, 780056Y	
μPD780058, 780058Y	0AH

Remark If a program for the μPD78F0058 or 78F0058Y which includes "MOV IXS, #0CH" is implemented with the μPD780055, 780055Y, 780056, or 780056Y, this instruction is ignored and causes no malfunction.

26.3 Flash Memory Programming

Data can be written to the flash memory with the device mounted on the target system (on-board). Write data ★ to the flash memory by connecting the dedicated flash programmer (Flashpro II) to the host machine and target system.

Remark Flashpro II is a product made by Naitou Densei Machidaseisakusho Co., Ltd.

26.3.1 Selecting communication mode

Data are written to the flash memory by using the Flashpro II and by means of serial communication. Select a communication mode from those listed in Table 26-4. To select a communication mode, use the format shown in Figure 26-3. Each communication mode is selected by the number of V$_{PP}$ pulses shown in Table 26-4.

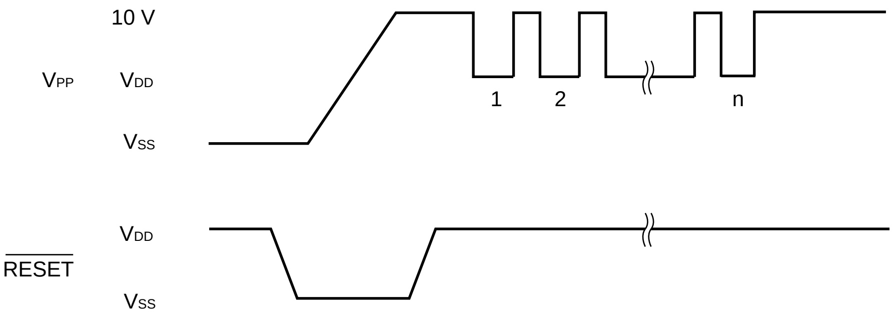
Table 26-4. Communication Modes

Communication Mode	Number of Channels	Pins Used	Number of VPP Pulses
3-wire serial I/O	3	P27/$\overline{\text{SCK0}}$/SCL P26/SO0/SB1/SDA1 P25/SI0/SB0/SDA0	0
		P22/$\overline{\text{SCK1}}$ P21/SO1 P20/SI1	1
		P72/$\overline{\text{SCK2}}$/ASCK P71/SO2/TxD0 P70/SI2/RxD0	2
UART	2	P71/SO2/TxD0 P70/SI2/RxD0	8
		P23/TxD1 P24/RxD1	9
Pseudo 3-wire serial I/O Note	1	P32/TO2 (serial clock input/output) P31/TO1 (serial data output) P30/TO0 (serial data input)	12

Note Serial transfer is executed by controlling the port by software.

Caution Be sure to select a communication mode by specifying the number of V$_{PP}$ pulses shown in Table 26-4.

Figure 26-3. Communication Mode Selecting Format



26.3.2 Flash memory programming function

Data is written to the flash memory by transmitting or receiving commands and data in the selected communication mode. The major functions are listed in Table 26-5.

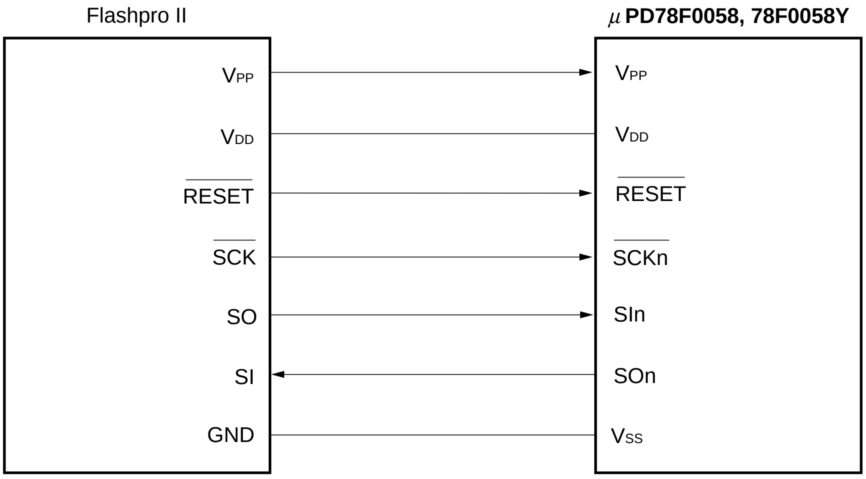
Table 26-5. Major Functions of Flash Memory Programming

Function	Description
Reset	Used to stop writing or detect communication synchronization
Batch verify	Compares all contents of memory with input data
Block verify	Compares contents of specified memory block with input data
Batch erase	Erase all contents of memory
Block erase	Erase contents of specified memory block with one memory block consisting of 16 KB
Convergence	Prevents excessive erasure
Batch blank check	Checks erased status of entire memory
Block blank check	Checks erased status of specified block
High-speed write	Writes flash memory based on write start address and number of written data (number of bytes)
Successive write	Successively writes based on information input in high-speed write mode
Status	Used to check current operation mode and end of operation
Oscillation frequency setting	Inputs frequency information on oscillator
Erase time setting	Inputs erase time of memory
Baud rate setting	Sets communication rate during communication in UART mode
Convergence time setting	Sets compensation time during convergence
Silicon signature read	Outputs device name, memory capacity, and block information of device

26.3.3 Connecting Flashpro II

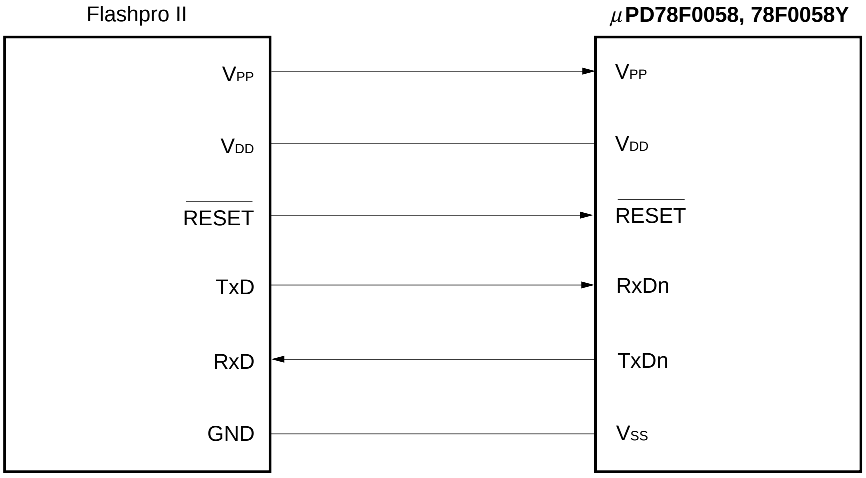
Connection between the Flashpro II and μPD78F0058 (or μPD78F0058Y) differs depending on the communication mode. Figures 26-4 to 26-6 show the connections in the respective communication modes.

Figure 26-4. Connection of Flashpro II in 3-wire Serial I/O Mode



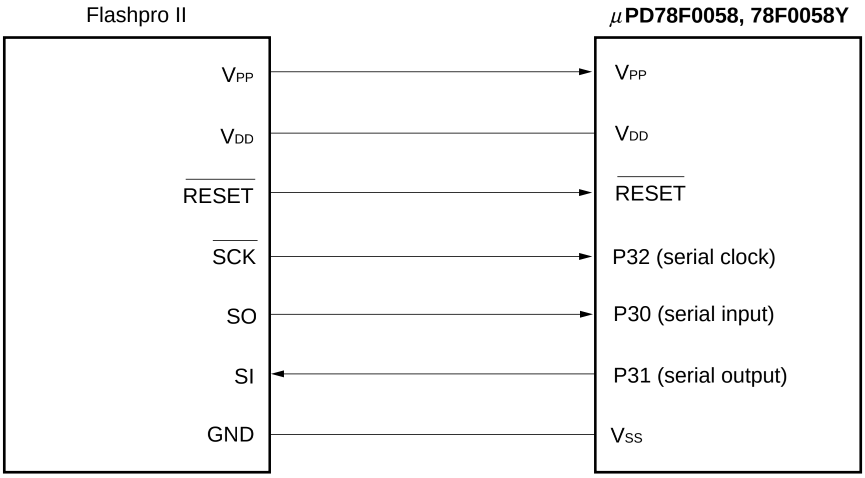
Note n = 0 to 2

Figure 26-5. Connection of Flashpro II in UART Mode



Note n = 0, 1

Figure 26-6. Connection of Flashpro II in Pseudo 3-wire Serial I/O Mode



[MEMO]

CHAPTER 27 INSTRUCTION SET OUTLINE

This chapter describes each instruction set of the μPD780058 and 780058Y Subseries as list table. For details of its operation and operation code, refer to the separate document **78K/0 Series User's Manual — Instructions (U12326E)**.

27.1 Legends Used in Operation List

27.1.1 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 27-1. Operand Identifiers and Description Methods

Identifier	Description Method
r rp sfr sfrp	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7), AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special-function register symbol Note Special-function register symbol (16-bit manipulatable register even addresses only) Note
saddr saddrp	FE20H to FF1FH Immediate data or labels FE20H to FF1FH Immediate data or labels (even address only)
addr16 addr11 addr5	0000H to FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions) 0800H to 0FFFH Immediate data or labels 0040H to 007FH Immediate data or labels (even address only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special-function register symbols, refer to **Table 5-2 Special-Function Register List**.

27.1.2 Description of operation column

A	:	A register; 8-bit accumulator
X	:	X register
B	:	B register
C	:	C register
D	:	D register
E	:	E register
H	:	H register
L	:	L register
AX	:	AX register pair; 16-bit accumulator
BC	:	BC register pair
DE	:	DE register pair
HL	:	HL register pair
PC	:	Program counter
SP	:	Stack pointer
PSW	:	Program status word
CY	:	Carry flag
AC	:	Auxiliary carry flag
Z	:	Zero flag
RBS	:	Register bank select flag
IE	:	Interrupt request enable flag
NMIS	:	Non-maskable interrupt servicing flag
( )	:	Memory contents indicated by address or register contents in parentheses
$\times_H$, $\times_L$	:	Higher 8 bits and lower 8 bits of 16-bit register
$\wedge$	:	Logical product (AND)
$\vee$	:	Logical sum (OR)
$\forall$	:	Exclusive logical sum (exclusive OR)
——	:	Inverted data
addr16	:	16-bit immediate data or label
jdisp8	:	Signed 8-bit data (displacement value)

27.1.3 Description of flag operation column

(Blank)	:	Nt affected
0	:	Cleared to 0
1	:	Set to 1
×	:	Set/cleared according to the result
R	:	Previously saved value is restored

27.2 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 ← A_0, A_{m-1} ← A_m) × 1 time			×
	ROL	A, 1	1	2	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1 time			×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1 time			×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1 time			×
	ROR4	[HL]	2	10	12 + n + m	A_{3-0} ← $(HL)_{3-0}$, $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$			
	ROL4	[HL]	2	10	12 + n + m	A_{3-0} ← $(HL)_{7-4}$, $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$			
BCD adjust	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipu- late	**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 manipu-late	**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 ← $\overline{\text{CY}}$			×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

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) ← $(PC + 3)_H$, (SP – 2) ← $(PC + 3)_L$, PC ← addr16, SP ← SP – 2			
	CALLF	!addr11	2	5	–	(SP – 1) ← $(PC + 2)_H$, (SP – 2) ← $(PC + 2)_L$, PC_{15-11} ← 00001, PC_{10-0} ← addr11, SP ← SP – 2			
	CALLT	[addr5]	1	6	–	(SP – 1) ← $(PC + 1)_H$, (SP – 2) ← $(PC + 1)_L$, PC_H ← (00000000, addr5 + 1), PC_L ← (00000000, addr5), SP ← SP – 2			
	BRK		1	6	–	(SP – 1) ← PSW, (SP – 2) ← $(PC + 1)_H$, (SP – 3) ← $(PC + 1)_L$, PC_H ← (003FH), PC_L ← (003EH), SP ← SP – 3, IE ← 0			
	RET		1	6	–	PC_H ← (SP + 1), PC_L ← (SP), SP ← SP + 2			
	RETI		1	6	–	PC_H ← (SP + 1), PC_L ← (SP), PSW ← (SP + 2), SP ← SP + 3, NMIS ← 0	R	R	R
	RETB		1	6	–	PC_H ← (SP + 1), PC_L ← (SP), PSW ← (SP + 2), SP ← SP + 3	R	R	R
Stack manipulate	**PUSH**	PSW	1	2	–	(SP – 1) ← PSW, SP ← SP – 1			
		rp	1	4	–	(SP – 1) ← rp_H, (SP – 2) ← rp_L, SP ← SP – 2			
	POP	PSW	1	2	–	PSW ← (SP), SP ← SP + 1	R	R	R
		rp	1	4	–	rp_H ← (SP + 1), rp_L ← (SP), SP ← SP + 2			
	MOVW	SP, #word	4	–	10	SP ← word			
		SP, AX	2	–	8	SP ← AX			
		AX, SP	2	–	8	AX ← SP			
Unconditional branch	**BR**	!addr16	3	6	–	PC ← addr16			
		$addr16	2	6	–	PC ← PC + 2 + jdisp8			
		AX	2	8	–	PC_H ← A, PC_L ← X			
Conditional branch	**BC**	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 0			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

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.

27.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

Second Operand First Operand	#byte	A	r Note	sfr	saddr	!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

Second Operand / 1st Operand	#word	AX	rp Note	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW Note						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

Second Operand / First Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call/instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

Second Operand / First Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

[MEMO]

★ APPENDIX A DIFFERENCES AMONG μPD78054, 78058F, AND 780058 SUBSERIES

Table A-1 shows the major differences among the μPD78054, 78058F, and 780058 Subseries.

Table A-1. Major Differences Among μPD78054, 78058F, and 780058 Subseries (1/2)

Product Name Item	μPD78054 Subseries	μPD78058F Subseries	μPD780058 Subseries
EMI noise measures	None	Provided	Provided
Supply voltage	V_{DD} = 2.0 to 6.0 V	V_{DD} = 2.7 to 6.0 V	V_{DD} = 1.8 to 5.5 V
PROM version	μPD78P054, 78P058	μPD78P058F	None
Flash memory version	None	None	μPD78F0058
Internal ROM size	μPD78052: 16 Kbytes μPD78053: 24 Kbytes μPD78054: 32 Kbytes μPD78P054: 32 Kbytes μPD78056: 48 Kbytes μPD78058: 60 Kbytes μPD78P058: 60 Kbytes	μPD78056F: 48 Kbytes μPD78058F: 60 Kbytes μPD78P058F: 60 Kbytes	μPD780053: 24 Kbytes μPD780054: 32 Kbytes μPD780055: 40 Kbytes μPD780056: 48 Kbytes μPD780058: 60 Kbytes μPD78F0058: 60 Kbytes
Internal high-speed RAM size	μPD78052: 512 bytes μPD78053, 78054, 78P054, 78056, 78058, 78P058: 1,024 bytes	1,024 bytes	1,024 bytes
I/O port	Total: 69 pins • CMOS Input: 2 pins • CMOS I/O: 63 pins • N-ch open-drain I/O: 4 pins		Total: 68 pins • CMOS Input: 2 pins • CMOS I/O: 62 pins • N-ch open-drain I/O: 4 pins
AV_{DD} pin	Power supply for A/D converter	Power supply for A/D converter and port output buffer	None (Power supplied to port output buffer is V_{DD0})
AV_{REF0} pin	Reference voltage input to A/D converter		Reference voltage input and analog power supply to A/D converter
Serial interface channel 2	3-wire serial I/O/UART mode		3-wire serial I/O/UART mode with time division function
External maskable interrupt	7 sources		6 sources
Emulation probe	EP-78230GC-R, EP-78054GK-R		EP-780058GC-R, EP-780058GK-R
Device file	DF78054		DF780058

Table A-1. Major Differences among μPD78054, 78058F, and 780058 Subseries (2/2)

Product Name Item	μPD78054 Subseries	μPD78058F Subseries	μPD780058 Subseries
Package	• 80-pin plastic QFP (14 × 14 mm, resin thickness: 2.7 mm) • 80-pin plastic QFP (14 × 14 mm, resin thickness: 1.4 mm) • 80-pin ceramic WQFN (14 × 14 mm) (μPD78P054, 78P058 only)	• 80-pin plastic QFP (14 × 14 mm, resin thickness: 2.7 mm) • 80-pin plastic QFP (14 × 14 mm, resin thickness: 1.4 mm) • 80-pin plastic TQFP (Fine pitch) (12 × 12 mm) (μPD78058F only)	• 80-pin plastic QFP (14 × 14 mm, resin thickness: 2.7 mm) • 80-pin plastic QFP (14 × 14 mm, resin thickness: 1.4 mm) • 80-pin plastic TQFP (Fine pitch) (12 × 12 mm)
Electrical specifications and recommended soldering conditions	Refer to Data Sheet of individual product.		

★

APPENDIX B DEVELOPMENT TOOLS

The following development tools are available for the development of systems which employ the μPD780058 and 780058Y Subseries.

Figure B-1 shows the configuration of the development tools.

Figure B-1. Development Tool Configuration (1/2)

(1) When using the in-circuit emulator IE-78K0-NS

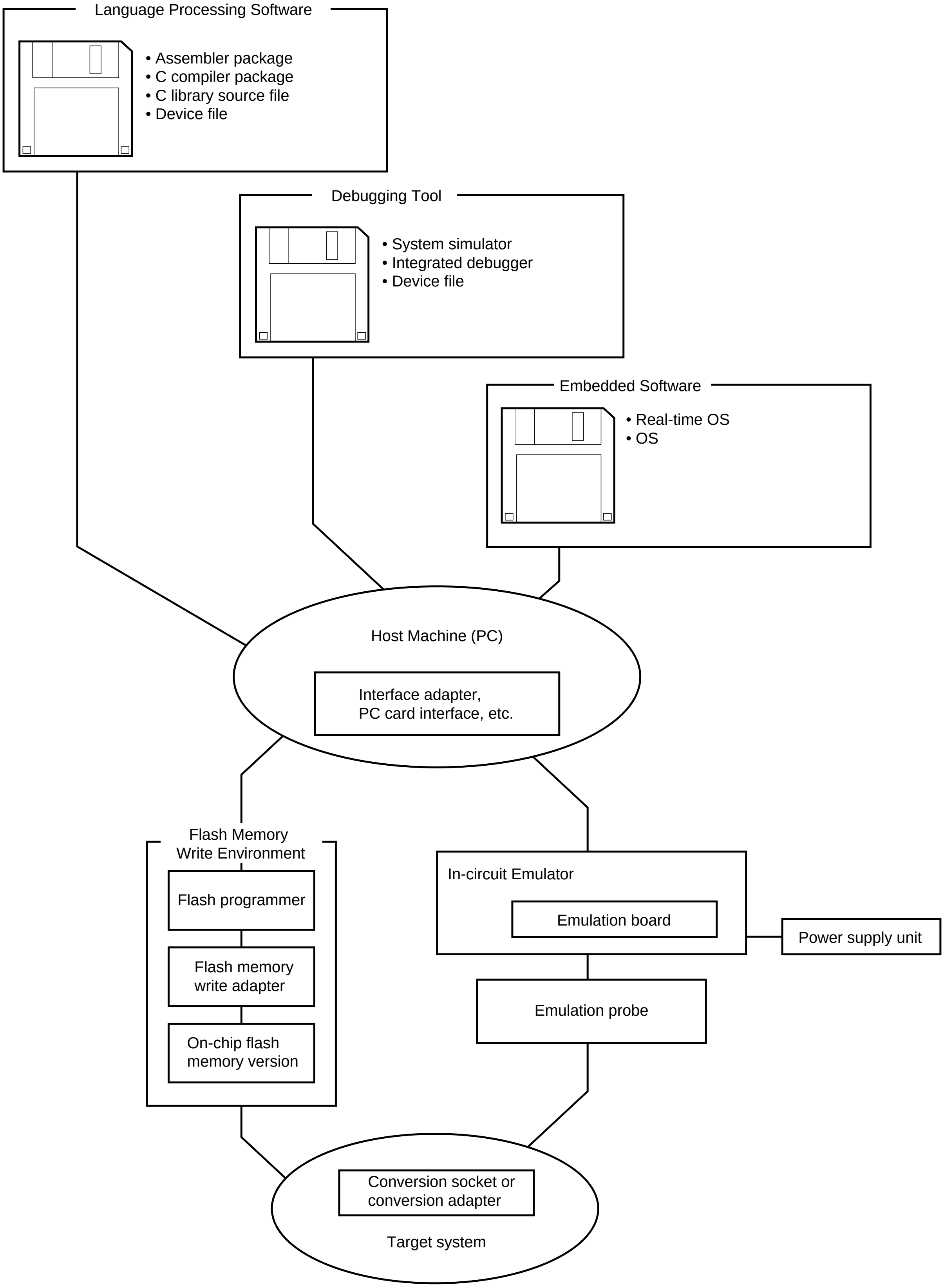
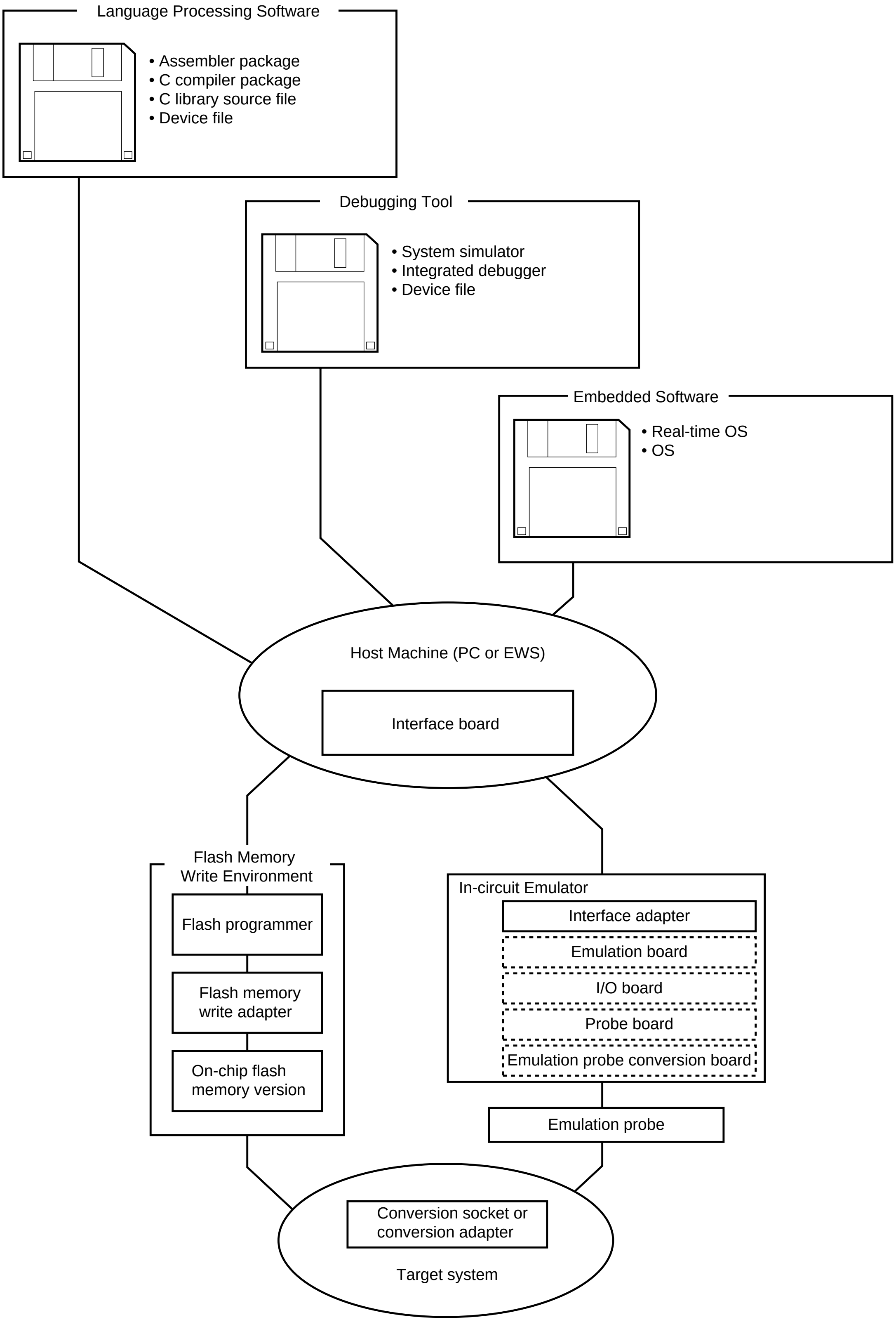


Figure B-1. Development Tool Configuration (2/2)

(1) When using the in-circuit emulator IE-78001-R-A



Remark Items in broken line boxes differ according to the development environment. Refer to **B.3.1 Hardware**.

B.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 (DF780058). <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
DF780058 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××××DF780024, μS××××DF780034
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 DF780058 can be used in common with the RA78K/0, CC78K/0, SM78K0, ID78K0-NS, and ID78K0.

Remark ×××× in the part number differs depending on the host machine and OS used.

µS××××RA78K0
µS××××CC78K0
µS××××DF780058
µS××××CC78K0-L

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	Windows (Japanese version)[Notes 1, 2]	3.5-inch 2HD FD
AB13	IBM PC/AT^{TM} and its compatibles	Windows (Japanese version)[Notes 1, 2]	3.5-inch 2HC FD
BB13		Windows (English version)[Notes 1, 2]	
3P16	HP9000 series 700^{TM}	HP-UX^{TM} (Rel. 9.05)	DAT (DDS)
3K13	SPARCstation^{TM}	SunOS^{TM} (Rel. 4.1.4)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS^{TM} (RISC)	NEWS-OS^{TM} (Rel. 6.1)	3.5-inch 2HC FD

Notes 1. Can be operated in DOS environment.
2. Not support WindowsNT^{TM}

B.2 Flash Memory Writing Tools

Flashpro II (type FL-PR2) Flash Programmer	Flash programmer dedicated to microcontrollers with on-chip flash memory.
FA-80GC FA-80GK[Note] Flash Memory Writing Adapter	Flash memory writing adapter used connected to the Flashpro II. • FA-80GC : 80-pin plastic QFP (GC-3B9, GC-8BT type) • FA-80GK : 80-pin plastic TQFP (GK-BE9 type)

Note Under development.

Remark Flashpro II, FA-80GC, and FA-80GK are products of Naito Densei Machidaseisakusho Co., Ltd.
Phone: (044) 822-3813 Naitou Densei Machidaseisakusho Co., Ltd.

B.3 Debugging Tools

B.3.1 Hardware (1/2)

(1) When using the in-circuit emulator IE-78K0-NS

IE-78K0-NS **Note** In-circuit Emulator		This is an in-circuit emulator for debugging the hardware and software when an application system using the 78K/0 Series is developed. It supports the integrated debugger (ID78K0-NS). This emulator is used with a power supply unit, emulation probe, and interface adapter for connecting a host machine.
IE-70000-MC-PS-B Power Supply Unit		This adapter is used to supply power from an AC 100 to 240-V outlet.
IE-70000-98-IF-C **Note** Interface Adapter		This adapter is necessary when a PC-9800 series PC (except notebook type) is used as the host machine for the IE-78K0-NS.
IE-70000-CD-IF **Note** PC Card Interface		There are a PC card and an interface cable necessary when the PC-9800 series notebook type computer is used as the host machine for the IE-78K0-NS.
IE-70000-PC-IF-C **Note** Interface Adapter		This adapter is necessary when IBM PC/AT or a compatible machine is used as the host machine for the IE-78K0-NS.
IE-780308-NS-EM1 **Note** Emulation Board		This board is used with an in-circuit emulator to emulate device-specific peripheral hardware.
NP-80GC Emulation Probe		This probe is for 80-pin plastic QFP (GC-3B9 and GC-8BT types) and connects an in-circuit emulator and the target system.
	EV-9200GC-80 Conversion Socket (Refer to **Figure B-2**)	This conversion socket connects the board of the target system created to mount 80-pin plastic QFP (GC-3B9 and GC-8BT types) and NP-80GC.
NP-80GK Emulation Probe		This probe is for 80-pin plastic TQFP (GK-BE9 type) and connects an in-circuit emulator and the target system.
	TGK-080SDW Conversion Adapter (Refer to **Figure B-3**)	This conversion adapter connects the board of the target system created to mount 80-pin plastic TQFP (GK-BE9 type) and TGK-080SDW.

Note Under development

Remarks **1.** NP-80GC and NP-80GK are products of Naitou Densei Machidaseisakusho Co., Ltd.
For further information, consult: Naitou Densei Machidaseisakusho Co., Ltd. (TEL (044) 822-3813).

2. TGK-080SDW is a product of TOKYO ELETECH Corporation.
For further information, consult:
Tokyo Electronic Components Div. (TEL (03) 3820-7112), or
Osaka Electronic Components Div. (TEL (06) 244-6672)
Daimaru Kogyo Corporation.

3. EV-9200GC-80s are sold in sets of five units.

4. TGK-080SDW is sold in single units.

B.3.1 Hardware (2/2)

(2) When using the in-circuit emulator IE-78001-R-A

IE-78001-R-A Note In-circuit Emulator		This is an in-circuit emulator for debugging the hardware and software when an application system using the 78K/0 Series is developed. It supports the integrated debugger (ID78K0). This emulator is used with an emulation probe and interface adapter for connecting a host machine.
IE-70000-98-IF-B or IE-70000-98-IF-C Note Interface Adapter		This adapter is necessary when a PC-9800 series PC (except notebook type) is used as the host machine for the IE-78001-R-A.
IE-70000-PC-IF-B or IE-70000-PC-IF-C Note Interface Adapter		This adapter is necessary when an IBM PC/AT or a compatible machine is used as the host machine for the IE-78001-R-A.
IE-78000-R-SV3 Interface Adapter		These are an adapter and a cable necessary when an EWS is used as the host machine for the IE-78001-R-A. They are connected to the board in the IE-78001-R-A. As Ethernet^{TM}, 10Base-5 is supported. A commercially available adapter is necessary if any other method is used.
IE-780308-NS-EM1 Note Emulation Board		This board is used with an in-circuit emulator and emulation probe conversion board to emulate device-specific peripheral hardware.
	IE-78K0-R-EX1 Note Emulation Prove Conversion Board	This is a board necessary when using the IE-780308-NS-EM1 with the IE-78001-R-A.
IE-780058GC-R Note Emulation Probe		This probe is for 80-pin plastic QFP (GC-3B9 and GC-8BT types) and connects an in-circuit emulator and the target system.
	EV-9200GC-80 Conversion Socket (Refer to **Figure B-2**)	This conversion socket connects the board of the target system created to mount 80-pin plastic QFP (GC-3B9 and GC-8BT types) and EP-780058GC-R.
EP-780058GK-R Emulation Probe		This probe is for 80-pin plastic TQFP (GK-BE9 type) and connects an in-circuit emulator and the target system.
	TGK-080SDW Conversion Adapter (Refer to **Figure B-3**)	This conversion adapter connects the board of the target system created to mount 80-pin plastic TQFP (GK-BE9 type) and EP-780058GK-R.

Note Under development

Remarks 1. TGK-080SDW is a product of TOKYO ELETECH Corporation.
For further information, consult:
Tokyo Electronic Components Div. (TEL (03) 3820-7112), or
Osaka Electronic Components Div. (TEL (06) 244-6672)
Daimaru Kogyo Corporation.

2. EV-9200GC-80s are sold in sets of five units.

3. TGK-080SDW is sold in single units.

B.3.2 Software (1/2)

SM78K0 System Simulator	This simulator can debug target system at C source level or assembler level while simulating operation of target system on host machine. SM78K0 runs on Windows. By using SM78K0, logic and performance of application can be verified without in-circuit emulator independently of hardware development, so that development efficiency and software quality can be improved. This simulator is used with optional device file (DF780058).
	Part Number: μS××××SM78K0

Remark ×××× of the part number differs depending on the host machine and OS used. Refer to the table below.

μS××××SM78K0

××××	Host Machine	OS	Medium
AA13	PC-9800 series	Windows (Japanese version) Note	3.5-inch 2HD FD
AB13	IBM PC/AT and its compatibles	Windows (Japanese version) Note	3.5-inch 2HC FD
BB13		Windows (English version) Note	

Note Does not support WindowsNT

B.3.2 Software (2/2)

ID78K0-NS [Note] Integrated Debugger (supporting in-circuit emulator IE-78K0-NS)	This debugger is a control program to debug the 78K/0 Series microcontrollers. It adopts a graphical user interface, which is equivalent visually and operationally to Windows or OSF/Motif^{TM}. 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 analyser, the efficiency of debugging programs, which run on real-time OSs can be improved. It should be used in combination with the optical device file (DF780058).
ID78K0 Integrated Debugger (supporting in-circuit emulator IE-78001-R-A)	
	Part Number: *μ*S××××ID78K0-NS, *μ*S××××ID78K0

Note Under development

Remark ×××× in the part number differs depending on the host machine and OS used.

*μ*S××××ID78K0-NS

××××	Host Machine	OS	Medium
AA13	PC-9800 series	Windows (Japanese version) [Note]	3.5-inch 2HD FD
AB13	IBM PC/AT and its compatibles	Windows (Japanese version) [Note]	3.5-inch 2HC FD
BB13		Windows (English version) [Note]	

Note Does not support WindowsNT

*μ*S××××ID78K0

××××	Host Machine	OS	Medium
AA13	PC-9800 Series	Windows (Japanese version) [Note]	3.5-inch 2HD FD
AB13	IBM PC/AT and its compatibles	Windows (Japanese version) [Note]	3.5-inch 2HC FD
BB13		Windows (English version) [Note]	
3P16	HP9000 series 700	HP-UX (Rel. 9.05)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel. 4.1.4)	3.5-inch 2HC FD
3K15			1/4 inch CGMT
3R13	NEWS^{TM} (RISC)	NEWS-OS (Rel. 6.1)	3.5-inch 2HC FD

Note Does not support WindowsNT

B.4 System-Upgrade Method 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 (under development).

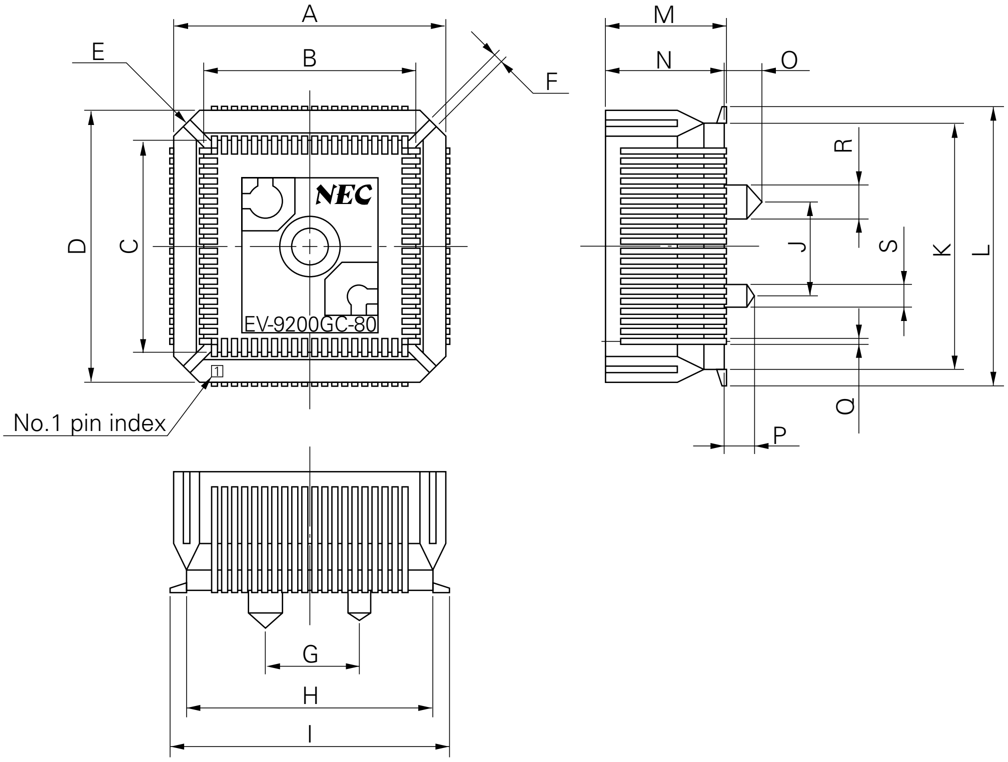
Table B-1. System-up Method from Former In-Circuit Emulator for 78K/0 Series to the IE-78001-R-A

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 upgrading a cabinet, send your in-circuit emulator to NEC.

Drawing and Footprint for Conversion Socket (EV-9200GC-80)

Figure B-2. EV-9200GC-80 Drawing (For Reference Only)

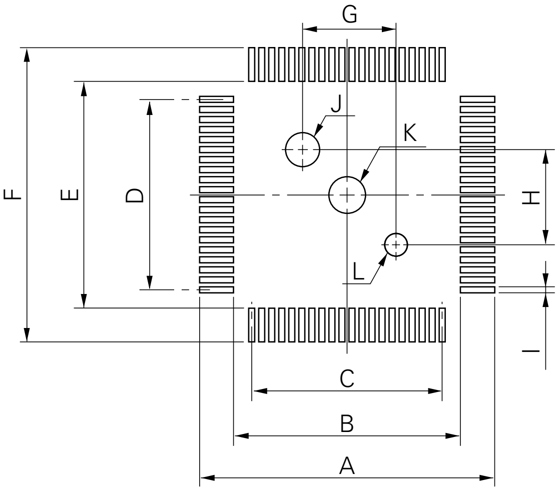


EV-9200GC-80-G0

ITEM	MILLIMETERS	INCHES
A	18.0	0.709
B	14.4	0.567
C	14.4	0.567
D	18.0	0.709
E	4-C 2.0	4-C 0.079
F	0.8	0.031
G	6.0	0.236
H	16.0	0.63
I	18.7	0.736
J	6.0	0.236
K	16.0	0.63
L	18.7	0.736
M	8.2	0.323
O	8.0	0.315
N	2.5	0.098
P	2.0	0.079
Q	0.35	0.014
R	ϕ2.3	ϕ0.091
S	ϕ1.5	ϕ0.059

Figure B-3. EV-9200GC-80 Footprint (For Reference Only)

Based on EV-9200GC-80
(2) Pad drawing (in mm)



EV-9200GC-80-P1E

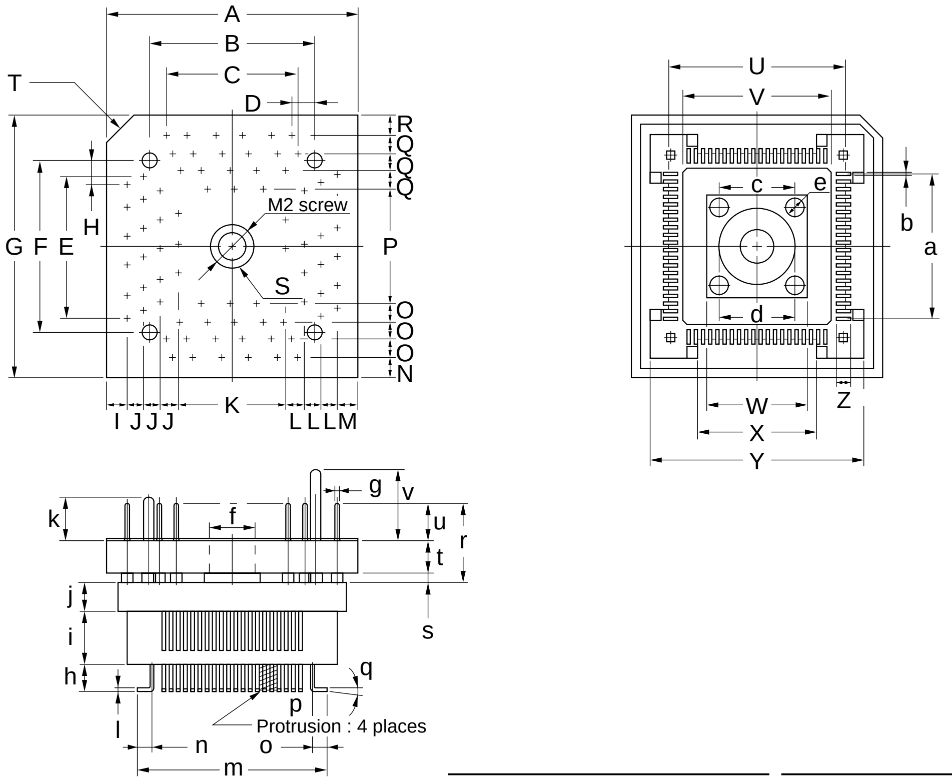
ITEM	MILLIMETERS	INCHES
A	19.7	0.776
B	15.0	0.591
C	0.65±0.02 × 19=12.35±0.05	$0.026^{+0.001}_{-0.002}$ × 0.748=$0.486^{+0.003}_{-0.002}$
D	0.65±0.02 × 19=12.35±0.05	$0.026^{+0.001}_{-0.002}$ × 0.748=$0.486^{+0.003}_{-0.002}$
E	15.0	0.591
F	19.7	0.776
G	6.0±0.05	$0.236^{+0.003}_{-0.002}$
H	6.0±0.05	$0.236^{+0.003}_{-0.002}$
I	0.35±0.02	$0.014^{+0.001}_{-0.001}$
J	ϕ2.36±0.03	$\phi 0.093^{+0.001}_{-0.002}$
K	ϕ2.3	ϕ0.091
L	ϕ1.57±0.03	$\phi 0.062^{+0.001}_{-0.002}$

Caution Dimensions of mount pad for EV-9200 and that for target device (QFP) may be different in some parts. For the recommended mount pad dimensions for QFP, refer to "SEMICONDUCTOR DEVICE MOUNTING TECHNOLOGY MANUAL" (C10535E).

Drawing of Conversion Adapter (TGK-080SDW)

Figure B-4. TGK-080SDW Drawing (For Reference Only) (unit: mm)

TGK-080SDW (TQPACK080SD + TQSOCKET080SDW)
Package dimension (unit: mm)



ITEM	MILLIMETERS	INCHES
A	18.0	0.709
B	11.77	0.463
C	0.5x19=9.5	0.020x0.748=0.374
D	0.5	0.020
E	0.5x19=9.5	0.020x0.748=0.374
F	11.77	0.463
G	18.0	0.709
H	0.5	0.020
I	1.58	0.062
J	1.2	0.047
K	7.64	0.301
L	1.2	0.047
M	1.58	0.062
N	1.58	0.062
O	1.2	0.047
P	7.64	0.301
Q	1.2	0.047
R	1.58	0.062
S	ϕ3.55	ϕ0.140
T	C 2.0	C 0.079
U	12.31	0.485
V	10.17	0.400
W	6.8	0.268
X	8.24	0.324
Y	14.8	0.583
Z	1.4±0.2	0.055±0.008

ITEM	MILLIMETERS	INCHES
a	0.5x19=9.5±0.10	0.020x0.748=0.374±0.004
b	0.25	0.010
c	ϕ5.3	ϕ0.209
d	ϕ5.3	ϕ0.209
e	ϕ1.3	ϕ0.051
f	ϕ3.55	ϕ0.140
g	ϕ0.3	ϕ0.012
h	1.85±0.2	0.073±0.008
i	3.5	0.138
j	2.0	0.079
k	3.0	0.118
l	0.25	0.010
m	14.0	0.551
n	1.4±0.2	0.055±0.008
o	1.4±0.2	0.055±0.008
p	h=1.8 ϕ1.3	h=0.071 ϕ0.051
q	0~5°	0.000~0.197°
r	5.9	0.232
s	0.8	0.031
t	2.4	0.094
u	2.7	0.106
v	3.9	0.154

TGK-080SDW-G1E

note: Product by TOKYO ELETECH CORPORATION.

[MEMO]

APPENDIX C EMBEDDED SOFTWARE

This section describes the embedded software which are provided for the μPD780058 and 780058Y Subseries to allow users to develop and maintain the application program for these subseries.

Real-Time OS (1/2)

RX78K/0 Real-time OS	Real-time OS conforming to μITRON specifications. Tool (configurator) that is used to create nucleus of RX78K/0 and multiple information tables is supplied. Used in combination with optional assembler package (RA78K/0) and device file (DF780058). **<Precaution when using RX78K/0 in PC environment>** Real-time OS is a DOS-based application. When using this application on Windows, use the DOS prompt.
	Part Number: μS××××RX78013-∆∆∆∆

Caution **When purchasing the RX78K/0, fill in the purchase application from in advance, and sign the License Agreement.**

Remark ×××× and ∆∆∆∆ of the part number differs depending on the host machine and operating system used. Refer to the table below.

μS××××RX78013-∆∆∆∆

∆∆∆∆	Product Outline	Max. Number for Use in Mass Production
001	Evaluation object	Do not use for mass production.
100K	Mass-production object	100,000
001M		1,000,000
010M		10,000,000
S01	Source program	Source program for mass-production object

××××	Host Machine	OS	Medium
AA13	PC-9800 series	Windows (Japanese version) Notes 1, 2	3.5-inch 2HD FD
AB13	IBM PC/AT and its compatibles	Windows (Japanese version) Notes 1, 2	3.5-inch 2HC FD
BB13		Windows (English version) Notes 1, 2	
3P16	HP9000 Series 700	HP-UX (Rel. 9.05)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel. 4.1.4)	3.5-inch 2HC FD
3K15			1/4 inch CGMT
3R13	NEWS (RISC)	NEWS-OS (Rel. 6.1)	3.5-inch 2HC FD

Notes 1. Can be operated in DOS environment.

2. Does not support WindowsNT

Real-Time OS (2/2)

MX78K0 OS	μITRON-specification subset OS. Nucleus of MX78K0 is supplied. This OS performs task management, event management, and time management. It controls the task execution sequence for task management and selects the task to be executed next. **<Precaution when using MX78K/0 in PC environment>** MX78K0 is a DOS-based application. When using this application on Windows, use the DOS prompt.
	Part Number: μS××××MX78K0-ΔΔΔ

Remark ×××× and ΔΔΔ of the part number differs depending on the host machine and operating system used. Refer to the table below.

μS××××MX78K0-ΔΔΔ

ΔΔΔ	Product Outline	Note
001	Evaluation object	Use for experimental production.
××	Mass-production object	Use for mass production.
S01	Source program	Can be purchased only when object for mass production has been purchased.

××××	Host Machine	OS	Medium
AA13	PC-9800 series	Windows (Japanese version) Notes 1, 2	3.5-inch 2HD FD
AB13	IBM PC/AT and its compatibles	Windows (Japanese version) Notes 1, 2	3.5-inch 2HC FD
BB13		Windows (English version) Notes 1, 2	
3P16	HP9000 Series 700	HP-UX (Rel. 9.05)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel. 4.1.4)	3.5-inch 2HC FD
3K15			1/4 inch CGMT
3R13	NEWS (RISC)	NEWS-OS (Rel. 6.1)	3.5-inch 2HC FD

Notes 1. Can be operated in DOS environment.

2. Does not support WindowsNT

APPENDIX D REGISTER INDEX

D.1 Register Index (Register Name)

D.2 Register Index (Symbol)

★

APPENDIX E   REVISION HISTORY

The revision history of this edition is listed in the table below. “Chapter” indicates the chapter of the preceding edition where the revision was made.

Edition	Revisions	Chapter
2nd edition	Change of following block diagrams of ports: **Figures 6-5** and **6-7 P20, P21, and P23 to P26 Block Diagram**, **Figures 6-6** and **6-8 P22 and P27 Block Diagram**, **Figure 6-9 P30 to P37 Block Diagram**, and **Figure 6-16 P71 and P72 Block Diagram**	**CHAPTER 6 PORT FUNCTIONS**
	Addition of **Table 7-2 Relationships between CPU Clock and Minimum Instruction Execution Time**	**CHAPTER 7 CLOCK GENERATOR**
	Addition of **Figures 9-10 and 9-13 Square Wave Output Operation Timing**	**CHAPTER 9 8-BIT TIMER/EVENT COUNTER**
	Correction of Note on BSYE in **Figure 16-5 Serial Bus Interface Control Register Format**	**CHAPTER 16 SERIAL INTERFACE CHANNEL 0 (μPD780058 Subseries)**
	Addition of Caution to **16.4.3 (2) (a) Bus release signal (REL)** and **(b) Command signal (CMD)**	
	Addition of **(3) MSB/LSB switching as the start bit** to **18.4.2 3-wire serial I/O mode operation**	**CHAPTER 18 SERIAL INTERFACE CHANNEL 1**
	Change of **18.4.3 (3) (d) Busy control option, (e) Busy & strobe control option**, and **(f) Bit slippage detection function** in old edition to **(4) Synchronization control**, and improvement of explanation	
	Correction of **Figure 19-11 Receive Error Timing**	**CHAPTER 19 SERIAL INTERFACE CHANNEL 2**
	Addition of **(3) MSB/LSB switching as the start bit** to **19.4.3 3-wire serial I/O mode**	
	Addition of **19.4.4 Restrictions in UART mode**	
	Addition of Note to **26.1 Memory Size Switching Register**	**CHAPTER 26 μPD78F0058, 78F0058Y**
	26.3 Flash Memory Programming Change of product name of flash programmer from Flashpro to Flashpro II	
	Addition of **APPENDIX A DIFFERENCES AMONG μPD78054, 78058F, AND 780058 SUBSERIES**	**APPENDIX A DIFFERENCES AMONG μPD78054, 78058F, AND 780058 SUBSERIES**
	Total revision: Support of in-circuit emulators IE-78K0-NS and IE-78001-R-A	**APPENDIX B DEVELOPMENT TOOLS**
	Total revision: Deletion of fuzzy inference development support system	**APPENDIX C EMBEDDED SOFTWARE**

[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: 02-719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Corporation
Semiconductor Solution Engineering Division
Technical Information Support Dept.
Fax: 044-548-7900

I would like to report the following error/make the following suggestion:

Document title: ____________________

Document number: ____________________ Page number: __________

____

____

____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	❏	❏	❏	❏
Technical Accuracy	❏	❏	❏	❏
Organization	❏	❏	❏	❏