To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

User's Manual

μPD78064B Subseries

8-Bit Single-chip Microcontroller

μPD78064B
μPD78P064B
μPD78064B(A)

Document No. U10785EJ2V0UM00 (2nd edition)
Date Published August 1997 N

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 this product from Japan is regulated by the Japanese government. To export this product may be prohibited without governmental license, the need for which must be judged by the customer. The export or re-export of this product from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

The application circuits and their parameters are for references only and are not intended for use in actual design-in's.

Regional Information

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

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

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

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

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

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

NEC Electronics Italiana s.r.1.
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

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

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

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

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

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

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

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

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

NEC do Brasil S.A.
Sao Paulo-SP, Brasil
Tel: 011-889-1680
Fax: 011-889-1689

Major Revisions in This Edition

Page	Description
Throughout	• Addition of μPD78064B(A) • Addition of 100-pin plastic LQFP (fine pitch) (14 × 14 mm) package to μPD78064B and 78P064B
p.2	Addition of **1.4 Quality Grade**
p.3, 17	Addition of notes on differences between AV_{DD} and AV_{SS} pins to **1.5 Pin Configuration (Top View)** and **2.1.1 Normal operating mode pins**
p.8	Addition of following subseries to **1.6 Development of 78K/0 Series**: μPD78075B, 78075BY, 780018, 780018Y, 780058, 780058Y, 780034, 780034Y, 780024, 780024Y, 78014H, 780964, 780924, 780228, 78044H, 78044F, 78098B, 780973, and 780805 subseries
p.12	Addition of **1.10 Differences between μPD78064B and μPD78064B(A)**
p.155, 160	Addition of **Figures 7-10** and **7-13 Timing of Square Wave Output Operation**
p.204, 211	Addition of notes on changing operation mode to **13.1 Serial Interface Channel 0 Functions** and (2) Serial operation mode register (CSIM0) in **13.3 Serial Interface Channel 0 Control Registers**
p.229	Addition of notes to (f) Busy signal ($\overline{BUSY}$), ready signal (READY) in (2) Definition of SBI in **13.4.3 SBI mode operation**
p.225, 247	Addition of notes to (11) Notes on SBI mode in **13.4.3 (2) (a) Bus release signal (REL), (b) command signal (CMD)**
p.281	Correction of **Figure 14-10 Receive Error Timing**
p.289	Addition of (3) Changing MSB/LSB first to **14.4.3 3-wire serial I/O mode**
p.291	Addition of **14.4.4 Limitations when using UART mode**
p.391	Addition of **APPENDIX A DIFFERENCES BETWEEN μPD78064 AND 78064B SUBSERIES**
p.393	**APPENDIX B DEVELOPMENT TOOLS** • Addition of PA-78P0308GC, PA-78P0308GF, and IE-780308-R-EM • Change of name of conversion adapter EV-9500GC-100 to TGC-100SDW • Deletion of Windows-compatible 5” FD
p.417	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 μPD78064B subseries and design and develop its application systems and programs.

★ • μPD78064B subseries: μPD78064B, 78P064B, 78064B(A)

Purpose

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

Organization

The μPD78064B subseries manual is separated into two parts: this manual and the instruction edition (common to the 78K/0 series).

μPD78064B subseries User's Manual (This manual)	78K/0 series User's Manual Instruction
• Pin functions • Internal block functions • Interrupt • Other on-chip peripheral functions	• CPU functions • Instruction set • Explanation of each instruction

How to Read This Manual

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

★ ○ When you use this manual as that of the μPD78064B(A):
→ Unless otherwise specified, the μPD78064B is treated as the representative models in this manual. If you use the μPD78064B(A), take the μPD78064B as 78064B(A).

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

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

○ When you know a register name and want to confirm its details:
→ Read **APPENDIX D REGISTER INDEX**.

○ To learn the μPD78064B subseries instruction function in detail:
→ Refer to the **78K/0 series User's Manual: Instructions (U12326E)**

○ To learn the electrical specifications of the μPD78064B subseries:
→ Refer to separately available **Data Sheet μPD78064B (U11590E), μPD78P064B Data Sheet (U11598E)**

○ To learn the application examples of the respective functions of the μPD78064B subseries:
→ Refer to the Application Note separately available.

★ **Caution The application examples presented in this manual are for the "standard" quality models in general-purpose electronic systems. If you wish to use the applications presented in this manual for electronic systems that require "special" quality models, thoroughly study the parts and circuits to be actually used, and their quality grade.**

Legend

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

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

- **Related documents for μPD78064B subseries**

	Document Name	Document No.	
		Japanese	English
	μPD78064B Data Sheet	U11590J	U11590E
	μPD78P064B Data Sheet	U11598J	U11598E
★	78K/0 Series Application Note —Fundamental (III)	U10182J	U10182E
	μPD78064B Subseries User's Manual	U10785J	This manual
	78K/0 Series User's Manual —Instruction	U12326J	U12326E
	78K/0 Series Instruction Table	U10903J	—
	78K/0 Series Instruction Set	U10904J	—
★	μPD78064B(A) Data Sheet	U11597J	U11597E

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

	Document Name		Document No.	
			Japanese	English
	RA78K Series Assembler Package	Operation	EEU-809	EEU-1399
		Language	EEU-815	EEU-1404
	RA78K Series Structured Assembler Preprocessor		EEU-817	EEU-1402
★	RA78K0 Assembler Package	Operation	U11802J	U11802E
★		Assembly Language	U11801J	U11801E
★		Structured Assembly Language	U11789J	U11789E
	CC78K Series C Compiler	Operation	EEU-656	EEU-1280
		Language	EEU-655	EEU-1284
★	CC78K0 C Compiler	Operation	U11517J	U11517E
★		Language	U11518J	U11518E
	CC78K C Compiler Application Note	Programming Know-How	EEA-618	EEA-1208
	CC78K Series Library Source File		U12322J	—
	PG-1500 PROM Programmer		U11940J	EEU-1355
	PG-1500 Controller PC-9800 Series (MS-DOS™) Base		EEU-704	EEU-1291
	PG-1500 Controller IBM PC Series (PC DOS™) Base		EEU-5008	U10540E
	IE-78000-R		U11376J	U11376E
★	IE-78000-R-A		U10057J	U10057E
	IE-78000-R-BK		EEU-867	EEU-1427
	IE-78064-R-EM		EEU-905	EEU-1443
★	IE-780308-R-EM		U11362J	U11362E
	EP-78064		EEU-934	EEU-1469
	SM78K0 System Simulator Windows™ Base	Reference	U10181J	U10181E
	SM78K0 Series System Simulator	External Parts User Open Interface Specification	U10092J	U10092E
	ID78K0 Integrated Debugger EWS Base	Reference	U11151J	—
★	ID78K0 Integrated Debugger PC Base	Reference	U11539J	U11539E
★	ID78K0 Integrated Debugger Windows Base	Guide	U11649J	U11649E
	SD78K/0 Screen Debugger	Introduction	EEU-852	U10539E
	PC-9800 Series (MS-DOS) Base	Reference	U10952J	—
	SD78K/0 Screen Debugger	Introduction	EEU-5024	EEU-1414
	IBM PC/AT™(PC DOS) Base	Reference	U11279J	U11279E

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

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

Document Name		Document No.	
		Japanese	English
78K/0 Series Real-Time OS	Fundamental	U11537J	—
	Installation	U11536J	—
78K/0 Series OS MX78K0	Fundamental	U12257J	—
Fuzzy Knowledge Data Creation Tool		EEU-829	EEU-1438
78K/0, 78K/II, 87AD Series Fuzzy Inference Development Support System—Translator		EEU-862	EEU-1444
78K/0 Series Fuzzy Inference Development Support System—Fuzzy Inference Module		EEU-858	EEU-1441
78K/0 Series Fuzzy Inference Development Support System—Fuzzy Inference Debugger		EEU-921	EEU-1458

- **Other Documents**

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

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

CONTENTS

LIST OF FIGURES (1/6)

LIST OF FIGURES (2/6)

LIST OF FIGURES (3/6)

LIST OF FIGURES (4/6)

LIST OF FIGURES (5/6)

LIST OF FIGURES (6/6)

LIST OF TABLES (1/3)

LIST OF FIGURES (2/3)

LIST OF FIGURES (3/3)

CHAPTER 1 OUTLINE

1.1 Features

○ Reduced EMI (Electro Magnetic Interference) noise compared to the existing μPD78064 subseries
○ On-chip high-capacity ROM and RAM

Type / Part Number	Program Memory (ROM)	Data Memory	
		Internal High-Speed RAM	LCD RAM
μPD78064B	32 Kbytes	1024 bytes	40 × 4 bytes
μPD78P064B			

○ 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
  • Multiply and divide instructions
○ Fifty-seven I/O ports (including alternative function pins for segment signal output)
○ LCD Controller / Driver
  • Segment signal output: Max. 40
  • Common signal output: Max. 4
  • Bias: 1/2, 1/3 bias switching possible
  • Power supply voltage: V_{DD} = 2.0 to 6.0 V (Static display mode)
    V_{DD} = 2.5 to 6.0 V (1/3 bias method)
    V_{DD} = 2.7 to 6.0 V (1/2 bias method)
○ 8-bit resolution A/D converter: 8 channels
○ Serial interface: 2 channels
  • 3-wire serial I/O/SBI/2-wire serial I/O mode: 1 channel
  • 3-wire serial I/O/UART mode: 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
○ Twenty vectored interrupt sources
○ Two test inputs
○ Two types of on-chip clock oscillators (main system clock and subsystem clock)
○ Power supply voltage: V_{DD} = 2.0 to 6.0 V

1.2 Application Fields

- **μPD78064B, 78P064B**
  Audio equipments containing tuner, communication devices such as radio equipment or cellular phones, and pagers, meters, etc.

★ • **μPD78064B(A)**
  Control unit of automotive appliances, gas leak breaker, safety devices, etc.

1.3 Ordering Information

Part number	Package	Internal ROM
μPD78064BGC-×××-7EA	100-pin plastic QFP (Fine pitch) (14 × 14 mm)	Mask ROM
★ μPD78064BGC-×××-8EU	100-pin plastic LQFP (Fine pitch) (14 × 14 mm)	Mask ROM
μPD78064BGF-×××-3BA	100-pin plastic QFP (14 × 20 mm)	Mask ROM
μPD78P064BGC-7EA	100-pin plastic QFP (Fine pitch) (14 × 14 mm)	One-time PROM
★ μPD78P064BGC-8EU	100-pin plastic LQFP (Fine pitch) (14 × 14 mm)	One-time PROM
μPD78P064BGF-3BA	100-pin plastic QFP (14 × 20 mm)	One-time PROM
★ μPD78064BGC(A)-×××-7EA	100-pin plastic QFP (Fine pitch) (14 × 14 mm)	Mask ROM
★ μPD78064BGF(A)-×××-3BA	100-pin plastic QFP (14 × 20 mm)	Mask ROM

Caution Two types of package are available for the μPD78064BGC and 78P064BGC. For the information of the suppliable package, consult NEC sales personnel.

Remark ××× indicates a ROM code suffix.

★ 1.4 Quality Grade

Part number	Package	Internal ROM
μPD78064BGC-×××-7EA	100-pin plastic QFP (Fine pitch) (14 × 14 mm)	Standard
μPD78064BGC-×××-8EU	100-pin plastic LQFP (Fine pitch) (14 × 14 mm)	Standard
μPD78064BGF-×××-3BA	100-pin plastic QFP (14 × 20 mm)	Standard
μPD78P064BGC-7EA	100-pin plastic QFP (Fine pitch) (14 × 14 mm)	Special
μPD78P064BGC-8EU	100-pin plastic LQFP (Fine pitch) (14 × 14 mm)	Special
μPD78P064BGF-3BA	100-pin plastic QFP (14 × 20 mm)	Special
μPD78064BGC(A)-×××-7EA	100-pin plastic QFP (Fine pitch) (14 × 14 mm)	Special
μPD78064BGF(A)-×××-3BA	100-pin plastic QFP (14 × 20 mm)	Special

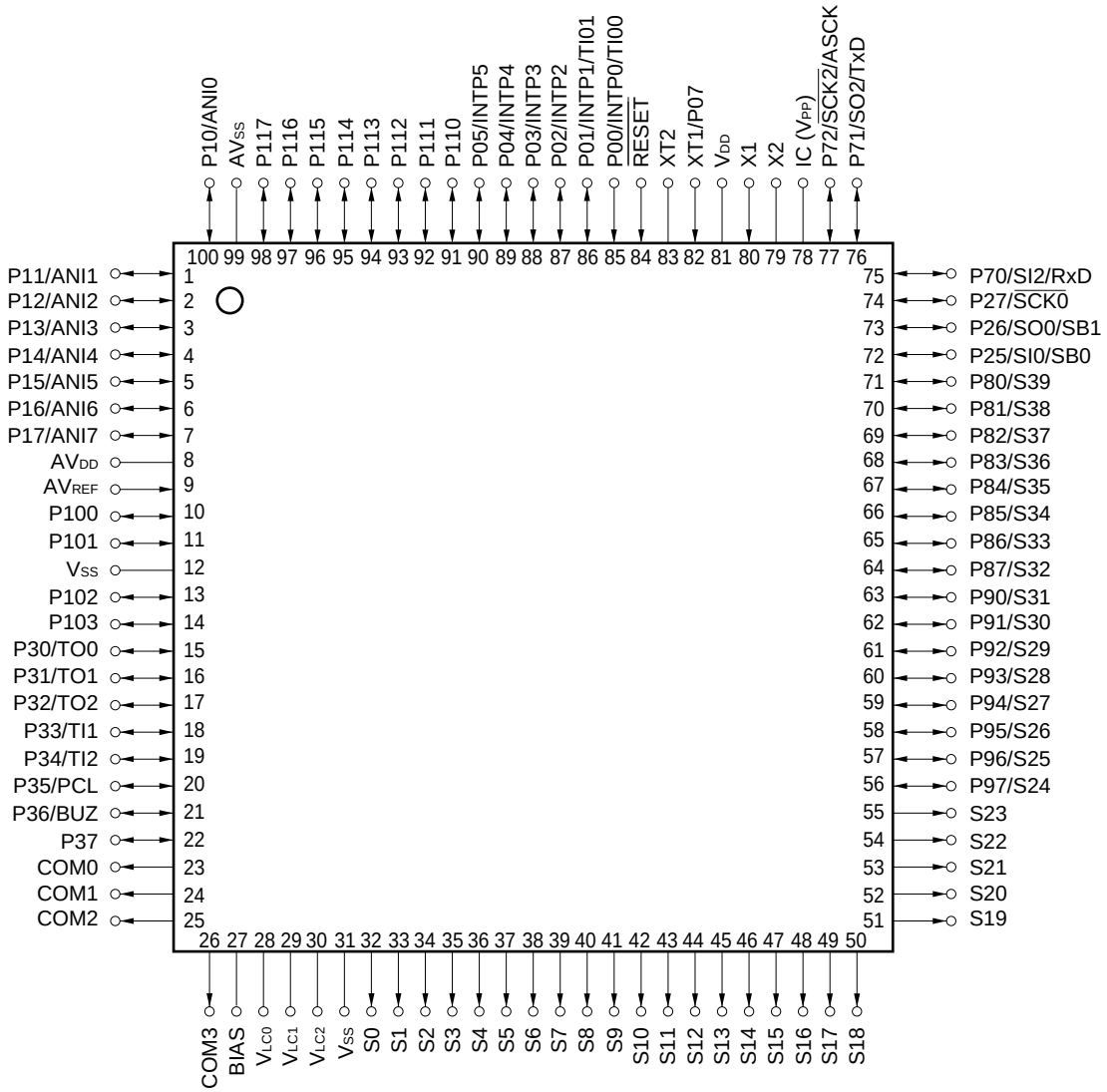
Remark ××× indicates a ROM code suffix.

Please refer to "Quality Grades on NEC Semiconductor Devices" (Document No. C11531E) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

1.5 Pin Configuration (Top View)

(1) Normal operating mode

- **100-pin plastic QFP (Fine pitch) (14 × 14 mm)**
★ μPD78064BGC-×××-7EA, 78P064BGC-7EA, 78064BGC(A)-×××-7EA
★ - **100-pin plastic LQFP (Fine pitch) (14 × 14 mm)**
μPD78064BGC-×××-8EU, 78P064BGC-8EU



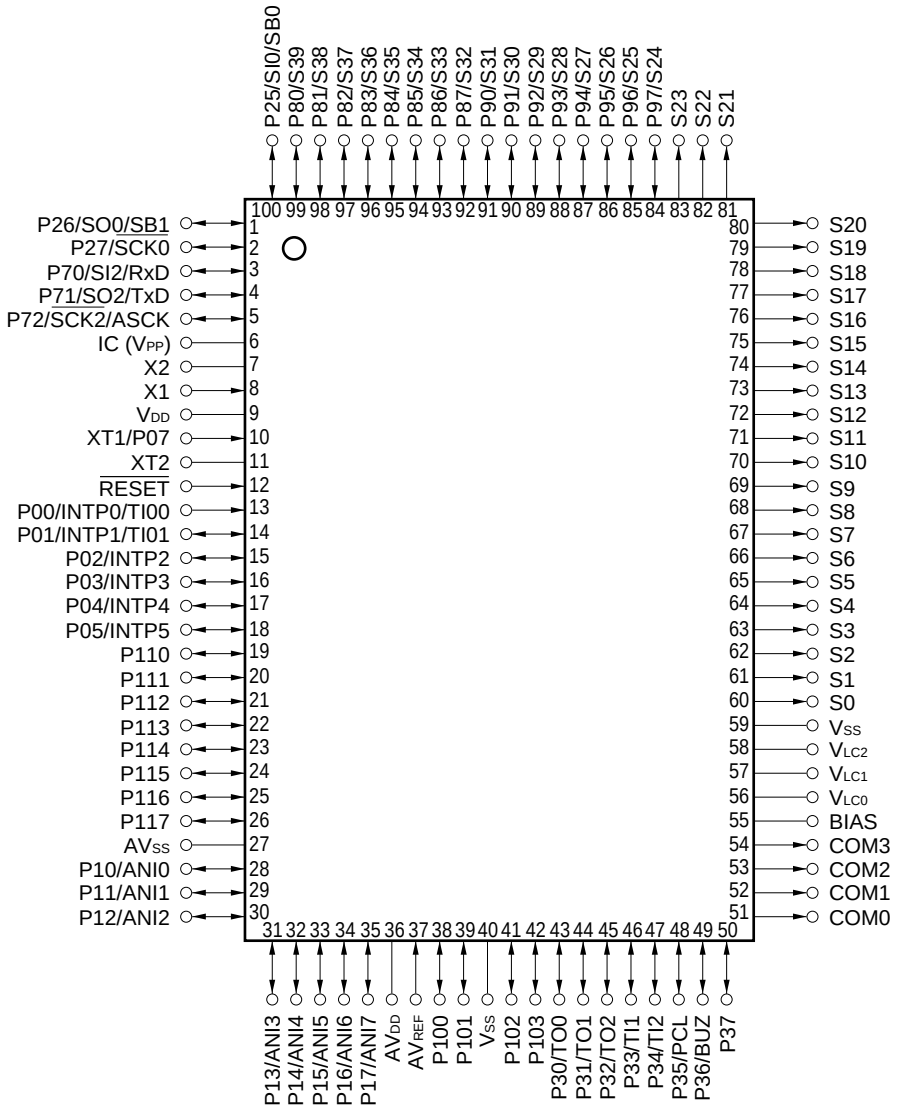
Cautions 1. Be sure to connect IC (Internally Connected) pin to V_{SS} directly.

★ **2. AV_{DD} pin shares the port power supply with that of the A/D converter. When using in applications where noise from inside the microcomputer has to be reduced, connect the AV_{DD} pin to a separate power supply, whose electrical potential is the same as that of V_{DD}.**

★ **3. AV_{SS} pin shares the port GND with that of the A/D converter. When using in applications where noise from inside the microcomputer has to be reduced, connect the AV_{SS} pin to a separate ground line.**

Remark Pin connection in parentheses is intended for the μPD78P064B.

★ • **100-pin plastic QFP (14 × 20 mm)**
μPD78064BGF-×××-3BA, 78P064BGF-3BA, 78064BGF(A)-×××-3BA



Cautions 1. Be sure to connect IC (Internally Connected) pin to V_{SS} directly.

★ **2. AV_{DD} pin shares the port power supply with that of the A/D converter. When using in applications where noise from inside the microcomputer has to be reduced, connect the AV_{DD} pin to a separate power supply, whose electrical potential is the same as that of V_{DD}.**

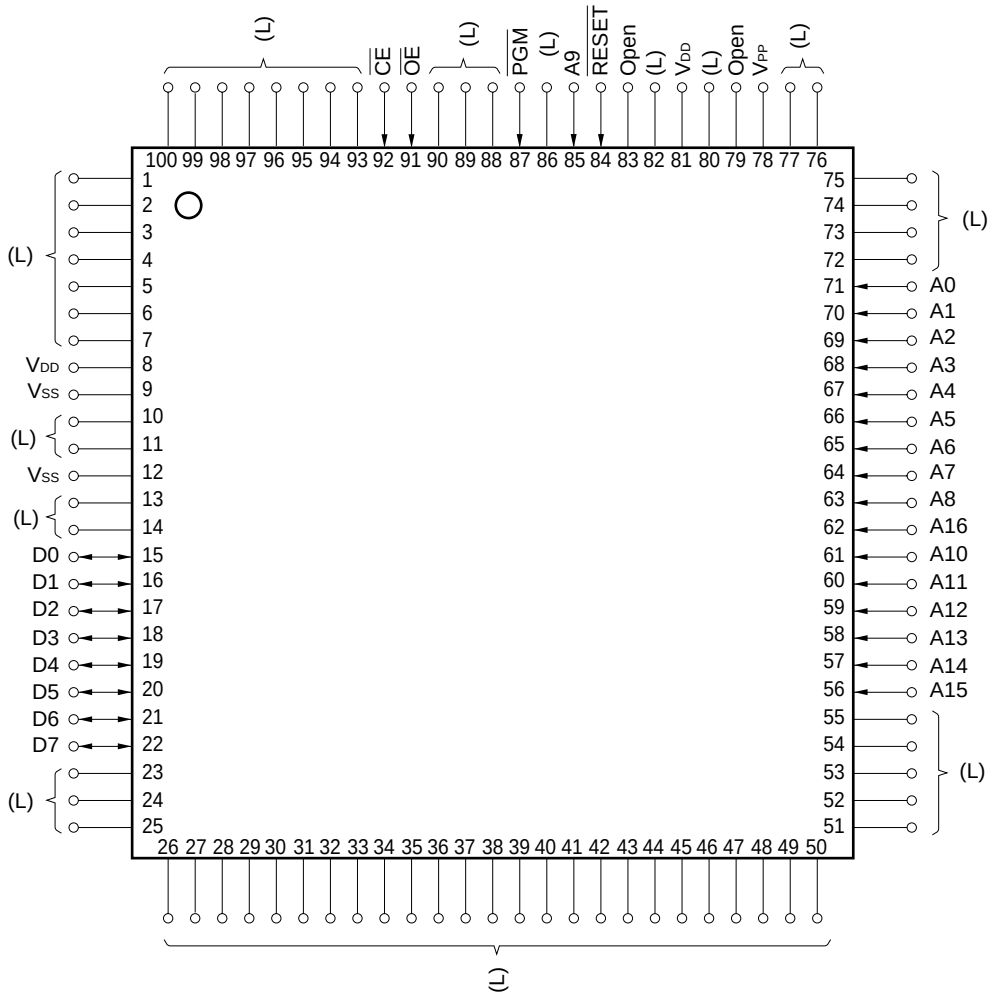
★ **3. AV_{SS} pin shares the port GND with that of the A/D converter. When using in applications where noise from inside the microcomputer has to be reduced, connect the AV_{SS} pin to a separate ground line.**

Remark Pin connection in parentheses is intended for the μPD78P064B.

ANI0-ANI7 : Analog Input
ASCK : Asynchronous Serial Clock
AV_{DD} : Analog Power Supply
AV_{REF} : Analog Reference Voltage
AV_{SS} : Analog Ground
BIAS : LCD Power Supply Bias Control
BUZ : Buzzer Clock
COM0-COM3 : Common Output
IC : Internally Connected
INTP0-INTP5 : Interrupt from Peripherals
P00-P05, P07 : Port 0
P10-P17 : Port 1
P25-P27 : Port 2
P30-P37 : Port 3
P70-P72 : Port 7
P80-P87 : Port 8
P90-P97 : Port 9
P100-P103 : Port 10
P110-P117 : Port 11
PCL : Programmable Clock
$\overline{RESET}$: Reset
RxD : Receive Data
S0-S39 : Segment Output
SB0, SB1 : Serial Bus
$\overline{SCK0}$, $\overline{SCK2}$: Serial Clock
SI0, SI2 : Serial Input
SO0, SO2 : Serial Output
TI00, TI01 : Timer Input
TI1, TI2 : Timer Input
TO0-TO2 : Timer Output
TxD : Transmit Data
V_{DD} : Power Supply
V_{LC0}-V_{LC2} : LCD Power Supply
V_{PP} : Programming Power Supply
V_{SS} : Ground
X1, X2 : Crystal (Main System Clock)
XT1, XT2 : Crystal (Subsystem Clock)

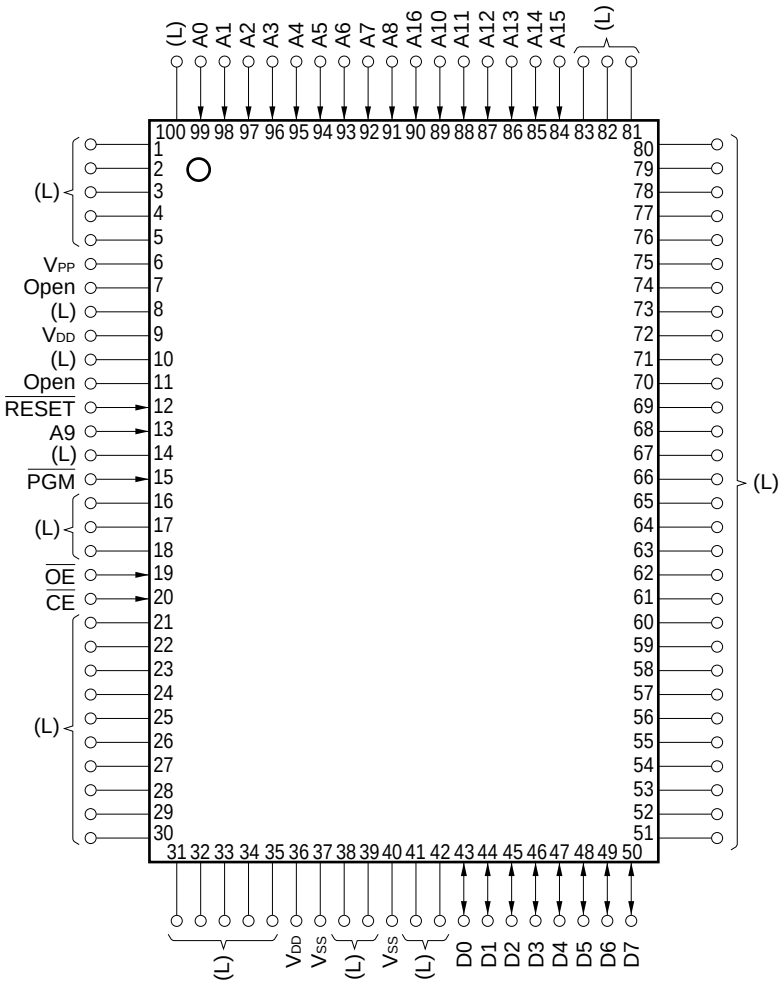
(2) PROM programming mode

- **100-pin plastic QFP (Fine pitch) (14 × 14 mm)**
  μPD78P064BGC-7EA
- ★ **100-pin plastic LQFP (Fine pitch) (14 × 14 mm)**
  μPD78P064BGC-8EU



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

- **100-pin plastic QFP (14 × 20 mm)**
  μPD78P064BGF-3BA

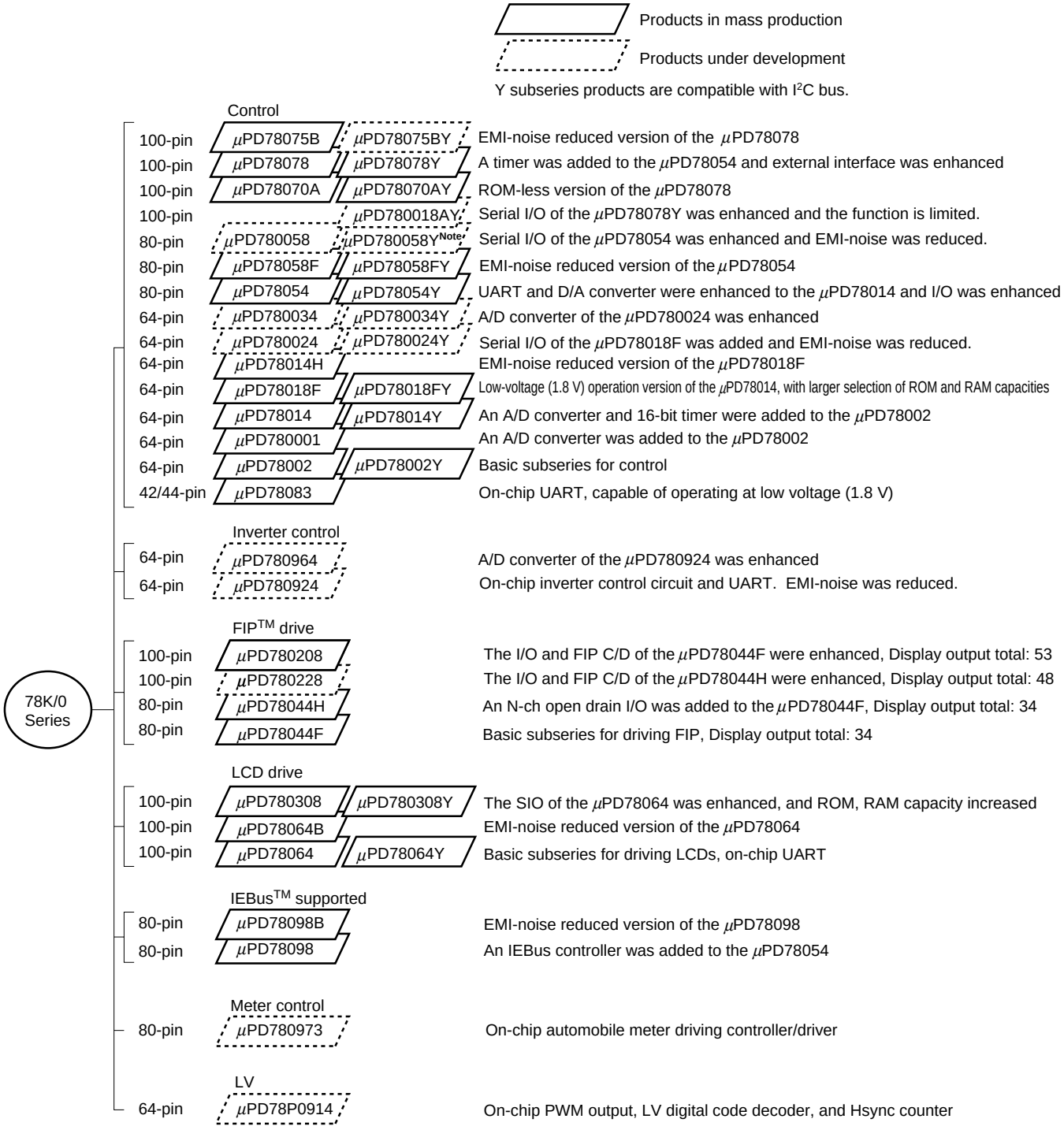


Cautions 1. (L) : Connect individually to V_{SS} via a pull-down resistor.
2. V_{SS} : Connect to the ground.
3. $\overline{RESET}$: Set to the low level.
4. Open : Do not connect anything.

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

★ 1.6 Development of 78K/0 Series

78K/0 series products evolution is illustrated below. Part numbers in the boxes indicate subseries names.



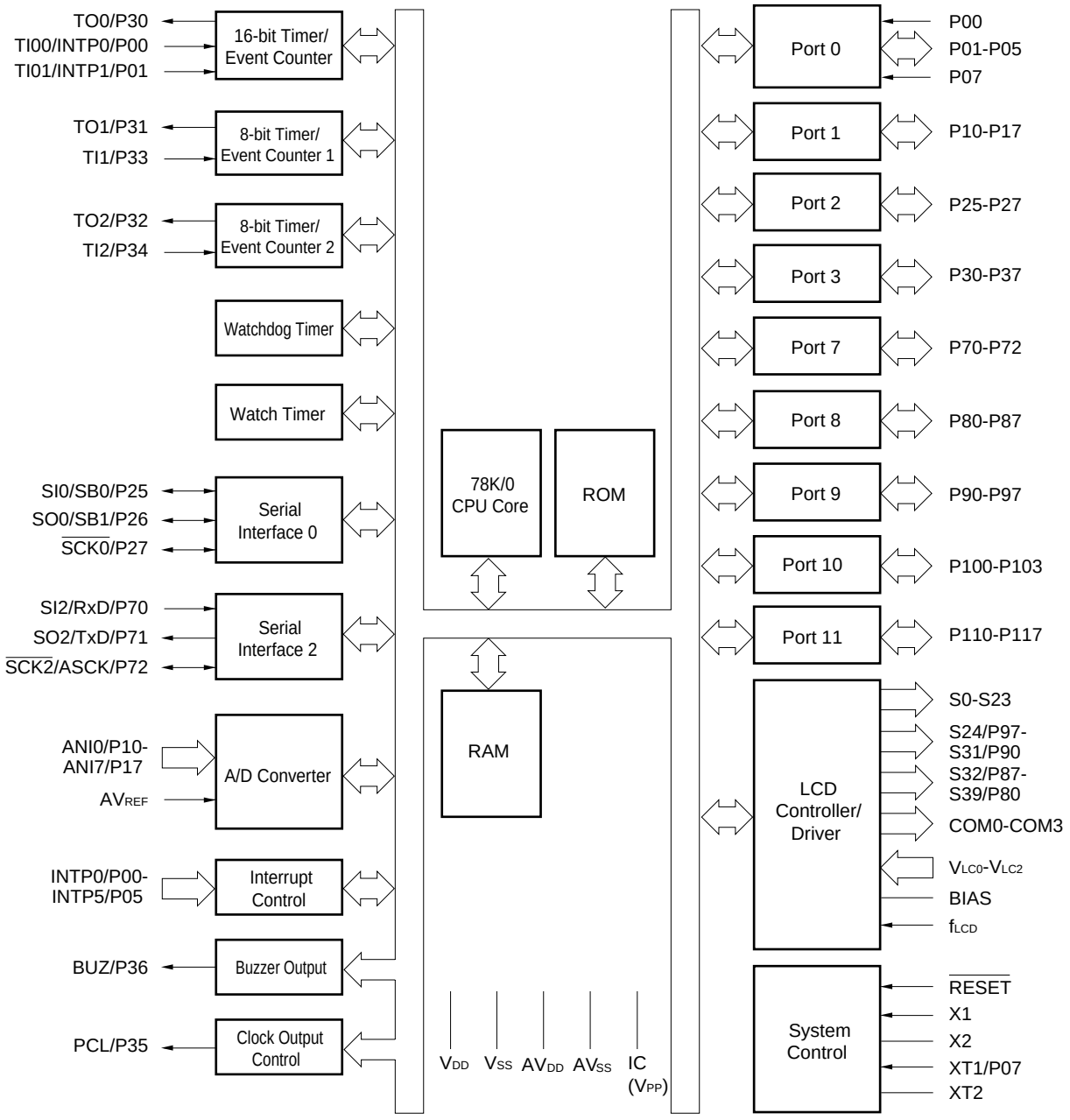
Note Under planning

The following lists the main functional differences between subseries products.

Subseries Name \ Function		ROM Capacity	Timer				8-bit A/D	10-bit A/D	8-bit D/A	Serial Interface	I/O	V_{DD} MIN. Value	External Expansion
			8-bit	16-bit	Watch	WDT							
Control	μPD78075B	32K-40K	4ch	1ch	1ch	1ch	8ch	–	2ch	3ch (UART: 1ch)	88	1.8 V	○
	μPD78078	48K-60K											
	μPD78070A	–									61	2.7 V	
	μPD780058	24K-60K	2ch						2ch	3ch (time division UART: 1ch)	68	1.8 V	
	μPD78058F	48K-60K								3ch (UART: 1ch)	69	2.7 V	
	μPD78054	16K-60K										2.0 V	
	μPD780034	8K-32K					–	8ch	–	3ch (UART: 1ch, time division 3-wire: 1ch)	51	1.8 V	
	μPD780024						8ch	–					
	μPD78014H									2ch	53	1.8 V	
	μPD78018F	8K-60K											
	μPD78014	8K-32K										2.7 V	
	μPD780001	8K		–	–					1ch	39		–
	μPD78002	8K-16K			1ch		–				53		○
	μPD78083				–		8ch			1ch (UART: 1ch)	33	1.8 V	–
Inverter control	μPD780964	8K-32K	3ch	**Note**	–	1ch	–	8ch	–	2ch (UART: 2ch)	47	2.7 V	○
	μPD780924						8ch	–					
FIP drive	μPD780208	32K-60K	2ch	1ch	1ch	1ch	8ch	–	–	2ch	74	2.7 V	–
	μPD780228	48K-60K	3ch	–	–					1ch	72	4.5 V	
	μPD78044H	32K-48K	2ch	1ch	1ch						68	2.7 V	
	μPD78044F	16K-40K								2ch			
LCD drive	μPD780308	48K-60K	2ch	1ch	1ch	1ch	8ch	–	–	3ch (time division UART: 1ch)	57	2.0 V	–
	μPD78064B	32K								2ch (UART: 1ch)			
	μPD78064	16K-32K											
IEBus supported	μPD78098B	40K-60K	2ch	1ch	1ch	1ch	8ch	–	2ch	3ch (UART: 1ch)	69	2.7 V	○
	μPD78098	32K-60K											
Meter control	μPD780973	24K-32K	3ch	1ch	1ch	1ch	5ch	–	–	2ch (UART: 1ch)	56	4.5 V	–
LV	μPD78P0914	32K	6ch	–	–	1ch	8ch	–	–	2ch	54	4.5 V	○

Note 10-bit timer: 1 channel

1.7 Block Diagram



Remark Pin connection in parentheses is intended for the μPD78P064B.

1.8 Outline of Function

Item \ Part Number		μPD78064B	μPD78P064B
Internal memory	ROM	Mask ROM	PROM
		32 Kbytes	
	High-speed RAM	1024 bytes	
	LCD display RAM	40 × 4 bits	
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)	
	With subsystem clock selected	122 μs (@ 32.768 kHz)	
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 (including alternative function pins for segment signal output)		• Total : 57 • CMOS input : 2 • CMOS I/O : 55	
A/D converter		8-bit resolution × 8 channels	
LCD controller/driver		• Segment signal output: Max. 40 • Common signal output: Max. 4 • Bias: 1/2, 1/3 bias selectable	
Serial interface		• 3-wire serial I/O/SBI/2-wire serial I/O mode selectable : 1 channel • 3-wire serial I/O mode / UART mode 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		Three outputs: (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 (@ 5.0 MHz with main system clock) 32.768 kHz (@ 32.768 kHz with subsystem clock)	
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (@ 5.0 MHz with main system clock)	
Vectored interrupt sources	Maskable	Internal: 12 External: 6	
	Non-maskable	Internal: 1	
	Software	1	
Test input		Internal: 1 External: 1	
Power supply voltage		V_{DD} = 2.0 to 6.0 V	
Operating ambient temperature		T_A = –40 to +85 °C	
Package		• 100-pin plastic QFP (Fine pitch) (14 × 14 mm) • 100-pin plastic LQFP (Fine pitch) (14 × 14 mm) • 100-pin plastic QFP (14 × 20 mm)	

★

1.9 Mask Options

The mask ROM versions (μPD78064B) provide dividing resistor mask options. By specifying this mask options at the time of ordering, dividing resistors which enable to generate LCD drive voltage suited to each bias method type can be incorporated. Using this mask option reduces the number of components to add to the device, resulting in board space saving.

The mask options provided in the μPD78064B subseries are shown in Table 1-1.

Table 1-1. Mask Options of Mask ROM Versions

Pin Names	Mask options
V_{LC0}-V_{LC2}	Dividing resistor can be incorporated.

★ 1.10 Differences between μPD78064B and μPD78064B(A)

Table 1-2. Differences between μPD78064B and μPD78064B(A)

Part Number / Item	μPD78064B	μPD78064B(A)
Quality grade	Standard	Special
Package	• 100-pin plastic QFP (fine-pitch) (14 × 14 mm) • 100-pin plastic LQFP (fine-pitch) (14 × 14 mm) • 100-pin plastic QFP (fine-pitch) (14 × 20 mm)	• 100-pin plastic QFP (fine-pitch) (14 × 14 mm) • 100-pin plastic QFP (fine-pitch) (14 × 20 mm)

1.11 Differences between μPD78064B Subseries and μPD78064 Subseries

1.11.1 Differences between μPD78064B subseries and μPD78064 subseries

The μPD78064B subseries features reduced noise (EMI noise) generated from the microcomputers compared to the μPD78064 subseries. Table 1-3 shows the differences between the μPD78064B subseries and μPD78064 subseries. Except for these differences, the μPD78064B subseries has the same functions as the μPD78064 subseries.

For the details of the μPD78064 subseries, refer to the **μPD78064, 78064Y Subseries User's Manual (U10105E)**.

For the electrical specifications, refer to the Data Sheet of the respective products.

Caution Do not perform the following operation on the pins shared with port pins (refer to (1) Port pins in 2.1.1 Normal operating mode pins) during A/D conversion operation; otherwise, the specifications of the absolute accuracy during A/D conversion cannot be satisfied (except the pins shared with LCD segment output pins).

(1) Rewriting the output latch of an output pin used as a port pin

(2) Changing the output level of an output pin even when it is not used as a port pin

Table 1-3. Differences between μPD78064B Subseries and μPD78064 Subseries

Item \ Part Number	μPD78064B Subseries	μPD78064 Subseries
Measures to reduce EMI noise	Provided	None
Internal ROM	32K bytes	16K to 32K bytes
Internal high-speed RAM	1024 bytes	512 to 1024 bytes
V_{DD} pin	Positive power (except ports)	Positive power (includes ports)
V_{SS} pin	Ground potential (except ports)	Ground potential (includes ports)
AV_{DD} pin	Analog power supply to A/D converter and power to ports	Analog power supply to A/D converter
AV_{SS} pin	Ground of A/D converter and ports	Ground of A/D converter
PROM	μPD78P064BGC μPD78P064BGF	μPD78P064GC μPD78P064GF μPD78P064KL-T

1.11.2 Reducing EMI noise

The noise generated from a microcomputer may affect the other electronic products and sets. A large part of the noise is generated by the current that flows when a circuit is turned ON or OFF. The generated noise affects the external circuits via the output ports of the microcomputer. To reduce the noise, the μPD78064B subseries adopts the following measures.

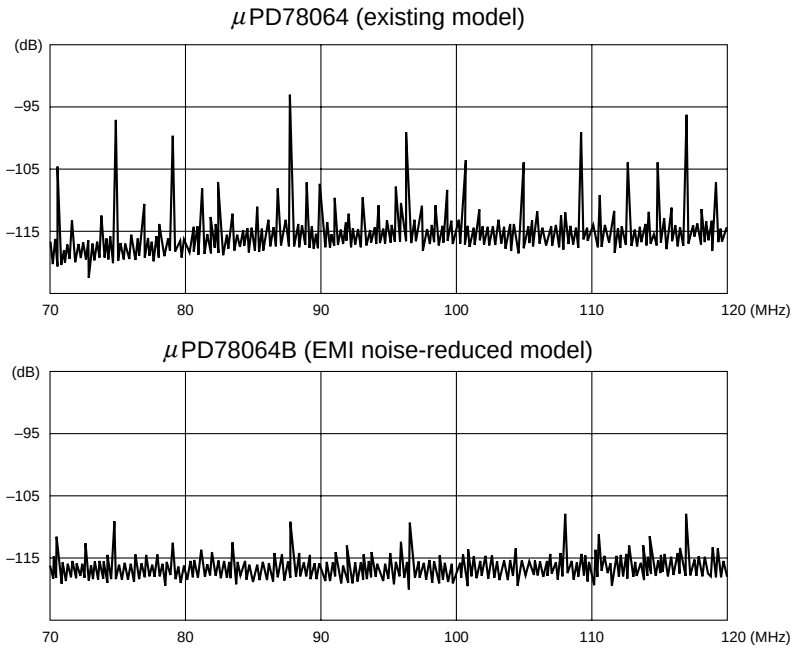
The internal circuits of the microcomputer through which an unnecessarily high current flows are checked and optimized. Moreover, the output port power supply and power supply for microcomputer operation are separated. The effects of these measures and how to measure the effects are explained below.

Supply different power to V_{DD} and AV_{DD} of the μPD78064B subseries, and separately ground V_{SS} and AV_{SS}.

- **Effect of noise reduction**

  The μPD78064B subseries reduces the total peak level of the noise by 5 to 10 dB as compared with the existing model (μPD78064).

Figure 1-1. Comparison of Noise Level between μPD78064B Subseries and Existing Model (μPD78064)



Caution The data shown above were measured with specific samples and for your reference only. These data do not guarantee the operations of the μPD78064B subseries.

- **Space noise test condition**
  An application evaluation program (supply current test mode) was executed on a 78K/0 evaluation board, and the noise component was measured by directly connecting a near-magnetic field probe to the IC on the evaluation board.

Table 1-4. Test Conditions

Test location	NEC laboratory (not shielded room)
Testers	Spectrum analyzer
	Plotter
	Near-magnetic field probe
	Amplifier for probe
	Oscilloscope
	Power supply
Evaluation software	78K/0 evaluation program
Operating frequency	4.19 MHz (crystal resonator)
Operating temperature	25 °C
Supply voltage	5 V
Processor clock control register	00H
Oscillation mode select register	01H

[MEMO]

CHAPTER 2 PIN FUNCTION

2.1 Pin Function List

2.1.1 Normal operating mode pins

(1) Port pins (1/2)

Pin Name	I/O	Function		After Reset	Dual-Function Pin
P00	Input	Port 0.	Input only	Input	INTP0/TI00
P01	Input/ output	7-bit input/output port.	Input/output mode can be specified bit-wise. If used as an input port, an internal pull-up resistor can be used by software.	Input	INTP1/TI01
P02					INTP2
P03					INTP3
P04					INTP4
P05					INTP5
P07Note 1	Input		Input only	Input	XT1
P10-P17	Input/ output	Port 1. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as input port, an internal pull-up resistor can be used by softwareNote 2.		Input	ANI0-ANI7
P25	Input/ output	Port 2. 3-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an internal pull-up resistor can be used by software.		Input	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 internal 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 internal to the subsystem clock oscillator).

2. When pins P10/ANI0 to P17/ANI7 are used as an analog input of the A/D converter, the internal pull-up resistor is automatically disabled.

(1) Port pins (2/2)

Pin Name	I/O	Function	After Reset	Dual-Function Pin
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 internal pull-up resistor can be used by software.	Input	SI2/RxD
P71				SO2/TxD
P72				$\overline{\text{SCK2}}$/ASCK
P80-P87	Input/ output	Port 8. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an internal pull-up resistor can be used by software. I/O port/segment signal output can be specified in 2-bit units by LCD display control register (LCDC).	Input	S39-S32
P90-P97	Input/ output	Port 9. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an internal pull-up resistor can be used by software. I/O port/segment signal output can be specified in 2-bit units by LCD display control register (LCDC).	Input	S31-S24
P100-P103	Input/ output	Port 10. 4-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an internal pull-up resistor can be used by software. A resistor can be connected. LED can be driven directly.	Input	—
P110-117	Input/ output	Port 11. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an internal pull-up resistor can be used by software. Falling edge can be detected.	Input	—

Caution Do not perform the following operation on the pins shared with port pins during A/D conversion operation; otherwise, the specifications of the absolute accuracy during A/D conversion cannot be satisfied (except the pins shared with LCD segment output pins).

(1) Rewriting the output latch of an output pin used as a port pin

(2) Changing the output level of an output pin even when it is not used as a port pin

(2) Pins other than port pins (1/2)

Pin Name	I/O	Function	After Reset	Dual-Function Pin
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
SI2				P70/RxD
SO0	Output	Serial interface serial data output	Input	P26/SB1
SO2				P71/TxD
SB0	Input/ output	Serial interface serial data input/output	Input	P25/SI0
SB1				P26/SO0
$\overline{SCK0}$	Input/ output	Serial interface serial clock input/output	Input	P27
$\overline{SCK2}$				P72/ASCK
RxD	Input	Asynchronous serial interface serial data input	Input	P70/SI2
TxD	Output	Asynchronous serial interface serial data output	Input	P71/SO2
ASCK	Input	Asynchronous serial interface serial clock input	Input	P72/$\overline{SCK2}$
TI00	Input	External count clock input to 16-bit timer (TM0)	Input	P00/INTP0
TI01		Capture trigger signal input to capture register (CR00)		P01/INTP1
TI1		External count clock input to 8-bit timer (TM1)		P33
TI2		External count clock input to 8-bit timer (TM2)		P34
TO0	Output	16-bit timer output (also used for 14-bit PWM output)	Input	P30
TO1		8-bit timer output		P31
TO2				P32
PCL	Output	Clock output (for main system clock and subsystem clock trimming)	Input	P35
BUZ	Output	Buzzer output	Input	P36
S0-S23	Output	Segment signal output of LCD controller/driver	Output	—
S24-S31			Input	P97-P90
S32-S39				P87-P80
COM0-COM3	Output	Common signal output of LCD controller/driver	Output	—
V_{LC0}-V_{LC2}	—	LCD drive voltage (mask ROM versions can incorporate dividing resistor (mask option.))	—	—
BIAS	—	Power supply for LCD drive	—	—
ANI0-ANI7	Input	A/D converter analog input	Input	P10-P17
AV_{REF}	Input	D/A converter analog output	—	—
AV_{DD}	—	A/D converter analog power supply (shared with power supply for port).	—	—

(2) Pins other than port pins (2/2)

Pin Name	I/O	Function	After Reset	Dual-Function Pin
AV_{SS}	—	A/D converter ground potential (Shared with ground potential of port).	—	—
$\overline{RESET}$	Input	System reset input	—	—
X1	Input	Crystal connection for main system clock oscillation	—	—
X2	—		—	—
XT1	Input	Crystal connection for subsystem clock oscillation	Input	P07
XT2	—		—	—
V_{DD}	—	Positive power supply (except port)	—	—
V_{PP}	—	High-voltage application for program write/verify. Connect directly to V_{SS} in normal operating mode.	—	—
V_{SS}	—	Ground potential (except port)	—	—
IC	—	Internal connection. Connect directly to V_{SS}.	—	—

★ **Cautions 1. AV_{DD} pin shares the port power supply with that of the A/D converter. When using in applications where noise from inside the microcomputer has to be reduced, connect the AV_{DD} pin to a separate power supply, whose electrical potential is the same as that of V_{DD}.**

★ **2. AV_{SS} pin shares the port GND with that of the A/D converter. When using in applications where noise from inside the microcomputer has to be reduced, connect the AV_{SS} pin to a separate ground line.**

2.1.2 PROM programming mode pins (μPD78P064B only)

Pin Name	I/O	Function
$\overline{RESET}$	Input	PROM programming mode setting. When +5 V or +12.5 V is applied to the V_{PP} pin or a low level voltage is applied to the $\overline{RESET}$ pin, the PROM programming mode is set.
V_{PP}	Input	High-voltage application for PROM programming mode setting and program write/verify.
A0-A16	Input	Address bus
D0-D7	Input/output	Data bus
$\overline{CE}$	Input	PROM enable input/program pulse input
$\overline{OE}$	Input	Read strobe input to PROM
$\overline{PGM}$	Input	Program/program inhibit input in PROM programming mode
V_{DD}	—	Positive power supply
V_{SS}	—	Ground potential

2.2 Description of Pin Functions

2.2.1 P00 to P05, P07 (Port 0)

These are 7-bit input/output ports. Besides serving as input/output ports, they function as an external interrupt request input, an external count clock input to the timer, a capture trigger signal input, and crystal connection for subsystem clock oscillation.

The following operating modes can be specified bit-wise.

(1) Port mode

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

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

(2) Control mode

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

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

(b) TI00

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

(c) TI01

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

(d) XT1

Crystal connect pin for subsystem clock oscillation

2.2.2 P10 to P17 (Port 1)

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

The following operating modes can be specified bit-wise.

(1) Port mode

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

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

(2) Control mode

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

2.2.3 P25 to P27 (Port 2)

These are 3-bit input/output ports. Besides serving as input/output ports, they function as data input/output to/from the serial interface and clock input/output.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 3-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 2 (PM2). When they are used as input ports, internal pull-up resistors can be used to them by defining the pull-up resistor option register L (PUOL).

(2) Control mode

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

(a) SI0, SO0

Serial interface serial data input/output pins

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

Serial interface serial clock input/output pins

(c) SB0 and SB1

NEC standard serial bus interface input/output pins

Caution When this port is used as a serial interface, the I/O and output latches must be set according to the function the user requires. For the setting, refer to Figure 13-4 Serial Operating Mode Register 0 Format.

2.2.4 P30 to P37 (Port 3)

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

The following operating modes can be specified bit-wise.

(1) Port mode

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

(2) Control mode

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

(a) TI1 and TI2

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

(b) TO0 to TO2

Timer output pins.

(c) PCL

Clock output pin.

(d) BUZ

Buzzer output pin.

2.2.5 P70 to P72 (Port 7)

These are 3-bit input/output ports. Beside serving as input/output ports, they function as serial interface data input/output, clock input/output.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 3-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 7 (PM7). When they are used as input ports, internal pull-up resistors can be used by defining the pull-up resistor option register L (PUOL).

(2) Control mode

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

(a) SI2, SO2

Serial interface serial data input/output pins

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

Serial interface serial clock input/output pin.

(c) RxD, TxD

Asynchronous serial interface serial data input/output pins.

(d) ASCK

Asynchronous serial interface serial clock input pin.

Caution When this port is used as a serial interface, the I/O and output latches must be set according to the function the user requires.
For the setting, refer to Table 14-2 Serial Interface Channel 2 Operating Mode Settings.

2.2.6 P80 to P87 (Port 8)

These are 8-bit input/output ports. Beside serving as input/output ports, they function as segment signal output of LCD controller/driver.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input/output ports with port mode register 8 (PM8). When they are used as input ports, internal pull-up resistors can be used by defining the pull-up resistor option register H (PUOH).

(2) Control mode

These ports function as segment signal output pins (S32 to S39) of LCD controller/driver.

2.2.7 P90 to P97 (Port 9)

These are 8-bit input/output ports. Beside serving as input/output ports, they function as segment signal output of LCD controller/driver.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input/output ports with port mode register 9 (PM9). When they are used as input ports, internal pull-up resistors can be used by defining the pull-up resistor option register H (PUOH).

(2) Control mode

These ports function as segment signal/output pins (S24 to S31) of LCD controller/driver.

2.2.8 P100 to P103 (Port 10)

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

LED can be driven directly.

2.2.9 P110 to P117 (Port 11)

These ports function as 8-bit input/output ports. They can be specified bit-wise as input/output ports with port mode register 11 (PM11). When they are used as input ports, internal pull-up resistors can be used by defining the pull-up resistor option register H (PUOH).

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

2.2.10 COM0 to COM3

These are LCD controller/driver common signal output pins. They output common signals under either of the following conditions:

– when the static mode is selected (COM0 to COM3 outputs)
– when 2-time-division (COM0, COM1 outputs) or 3-time-division (COM0 to COM2 outputs) operation is performed in 1/2 bias mode
– when 3-time-division (COM0 to COM2 outputs) or 4-time-division (COM0 to COM3 outputs) operation is performed in 1/3 bias mode

2.2.11 V_{LC0}-V_{LC2}

These are LCD-driving voltage pins. The mask ROM versions can have split resistors by mask option so that LCD driving voltage can be supplied inside the V_{LC0}-V_{LC2} pins according to the required bias without connecting external dividing resistors.

2.2.12 BIAS

These are LCD driving power supply pins. They should be connected to the V_{LC0} pin to realize user-desired LCD drive voltages to change resistance division ratios, or should be connected to external resistors together with the V_{LC0}-V_{LC2} pins and V_{SS} pin to fine-adjust the LCD-driving power voltage.

2.2.13 AV_{REF}

This pin inputs the reference voltage for the on-chip A/D converter.

When not using the A/D converter,connect this pin to the V_{SS} line.

2.2.14 AV_{DD}

This pin supplies analog voltage for the on-chip A/D converter and voltage for port. Even when not using the A/D converter, always use this pin at the same potential as V_{DD}.

2.2.15 AV_{SS}

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

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

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

2.2.17 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.

2.2.18 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.

2.2.19 V_{DD}

Positive power supply pin (except port)

2.2.20 V_{SS}

Ground potential pin (except port)

2.2.21 V_{PP} (μPD78P064B only)

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

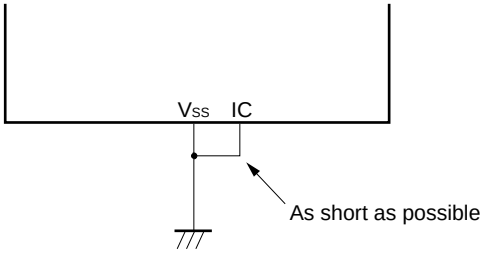
Connect directly to V_{SS} in normal operating mode.

2.2.22 IC (Mask ROM version only)

The IC (Internally Connected) pin is provided to set the test mode to check the μPD78064B subseries at delivery. Connect it directly to the V_{SS} with the shortest possible wire in the normal operating mode.

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

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



2.3 Input/output Circuits and Recommended Connection of Unused Pins

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

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

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
P00/INTP0/TI00	2	Input	Connect to V_{SS}.
P01/INTP1/TI01	8-D	Input/output	Connect independently via a resistor to V_{SS}.
P02/INTP2			
P03/INTP3			
P04/INTP4			
P05/INTP5			
P07/XT1	16	Input	Connect to V_{DD}.
P10/ANI0-P17/ANI7	11-C	Input/output	Connect independently via a resistor to V_{DD} or V_{SS}.
P25/SI0/SB0	10-C		
P26/SO0/SB1			
P27/$\overline{\text{SCK0}}$			
P30/TO0	5-J		
P31/TO1			
P32/TO2			
P33/TI1	8-D		
P34/TI2			
P35/PCL	5-J		
P36/BUZ			
P37			
P70/SI2/RxD	8-D		
P71/$\overline{\text{SO2}}$/TxD	5-J		
P72/SCK2/ASCK	8-D		
P80/S39-P87/S32	17-E		
P90/S31-P97/S24			
P100-P103	5-J		
P110-P117	8-D		Connect to V_{SS}.
S0-S23	17-D	Output	Open
COM0-COM3	18-B		
V_{LC0}-V_{LC2}	—	—	
BIAS			
$\overline{\text{RESET}}$	2	Input	—
XT2	16	—	Open

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

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
AV_{REF}	—	—	Connect to V_{SS}.
AV_{DD}			Connect to V_{DD}.
AV_{SS}			Connect to V_{SS}.
IC (Mask ROM version)			Connect directly to V_{SS}.
V_{PP} (μPD78P064B version)			

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

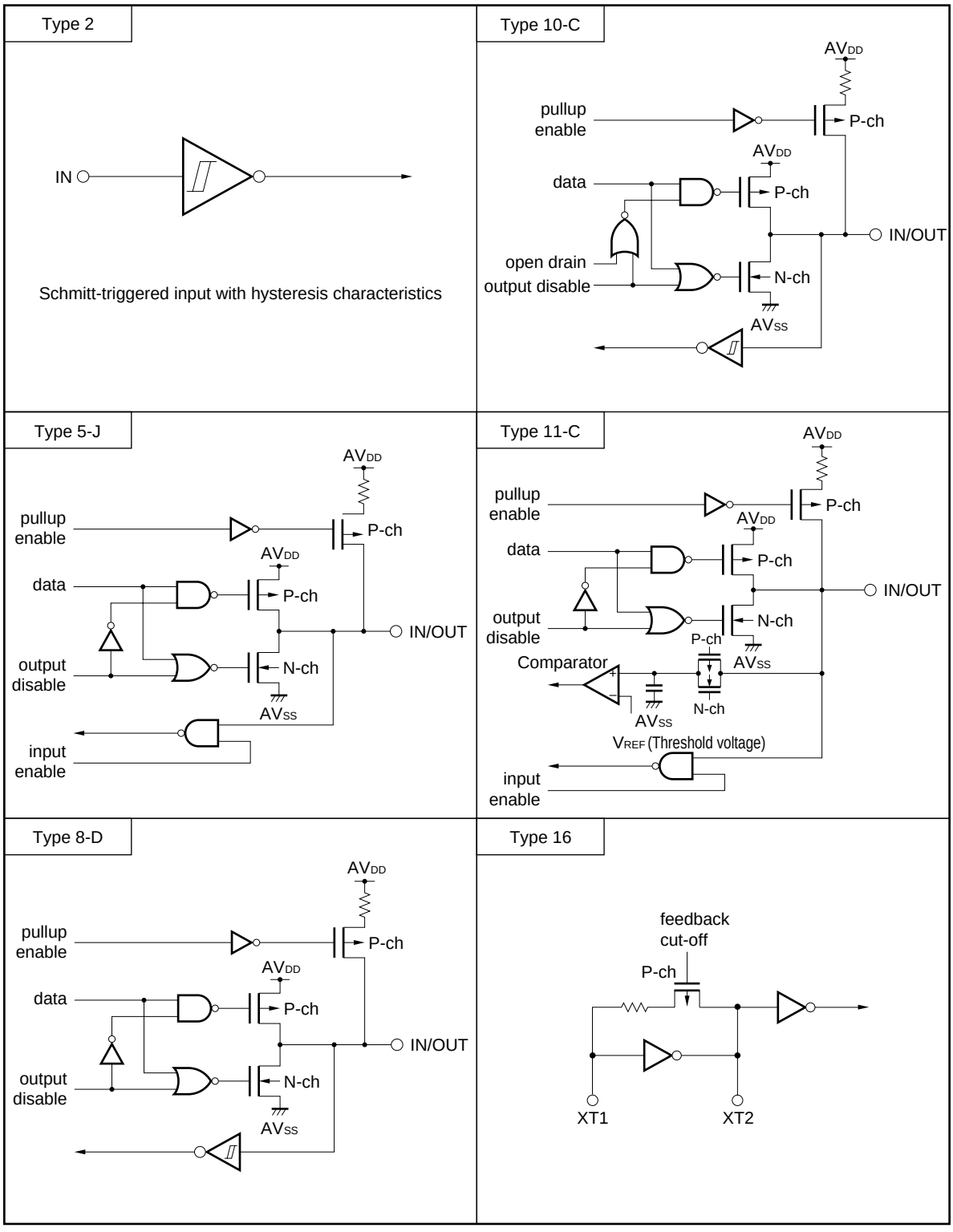
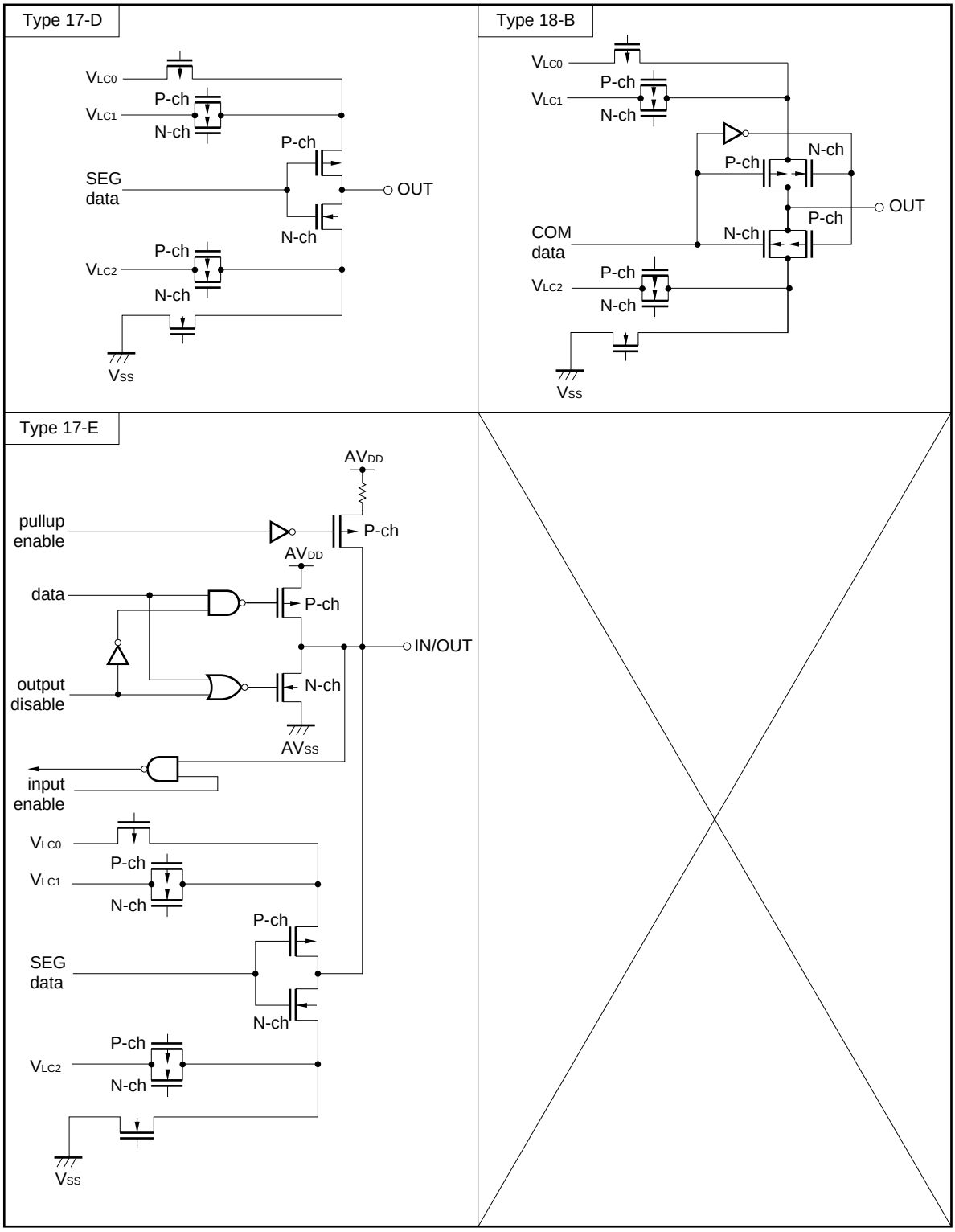


Figure 2-1. Pin Input/Output Circuit List (2/2)



[MEMO]

3.1 Memory Spaces

The μPD78064B Subseries models can access 64 K bytes memory space. Figures 3-1 to 3-2 shows memory maps.

Figure 3-1. Memory Map (μPD78064B)

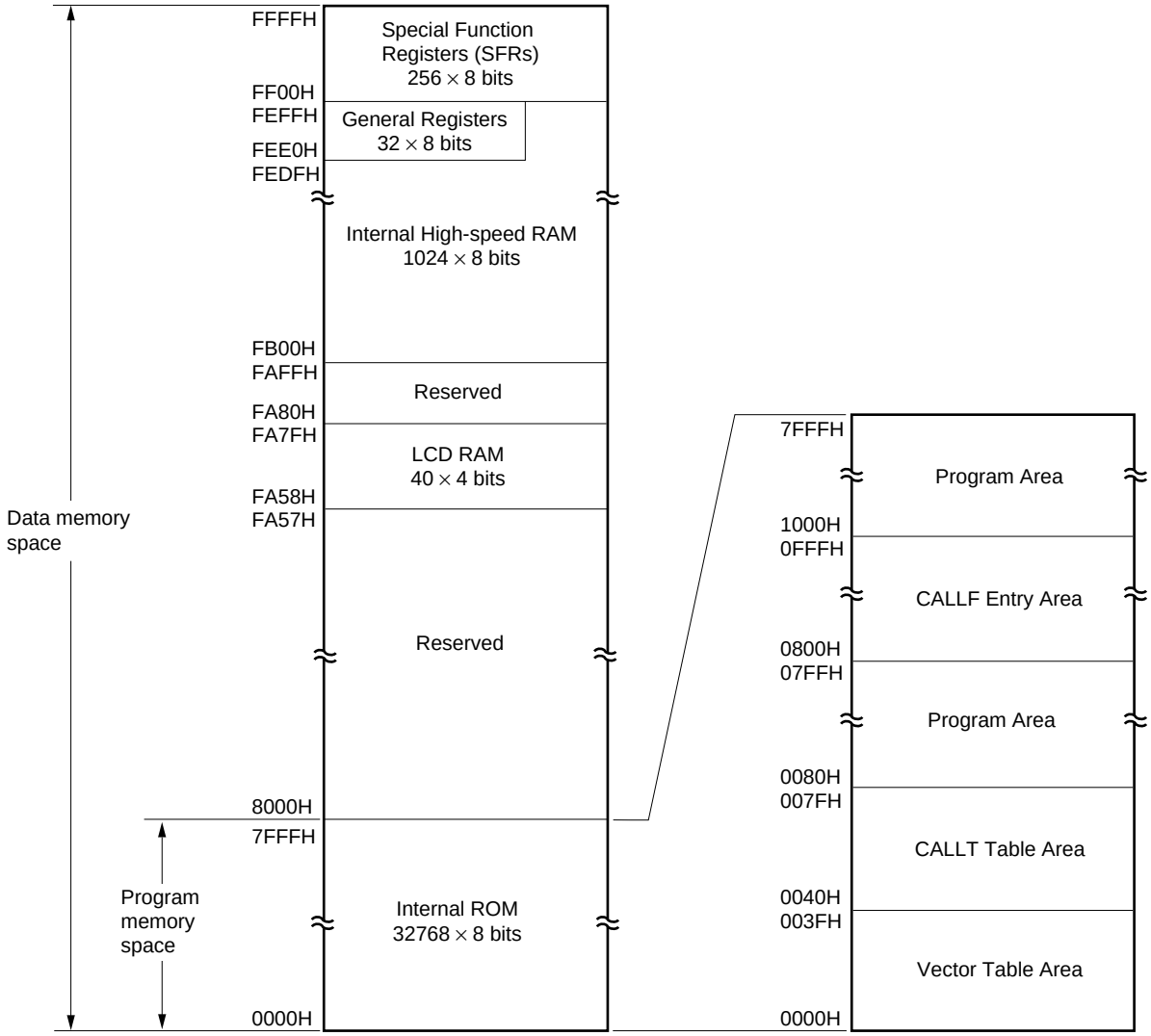
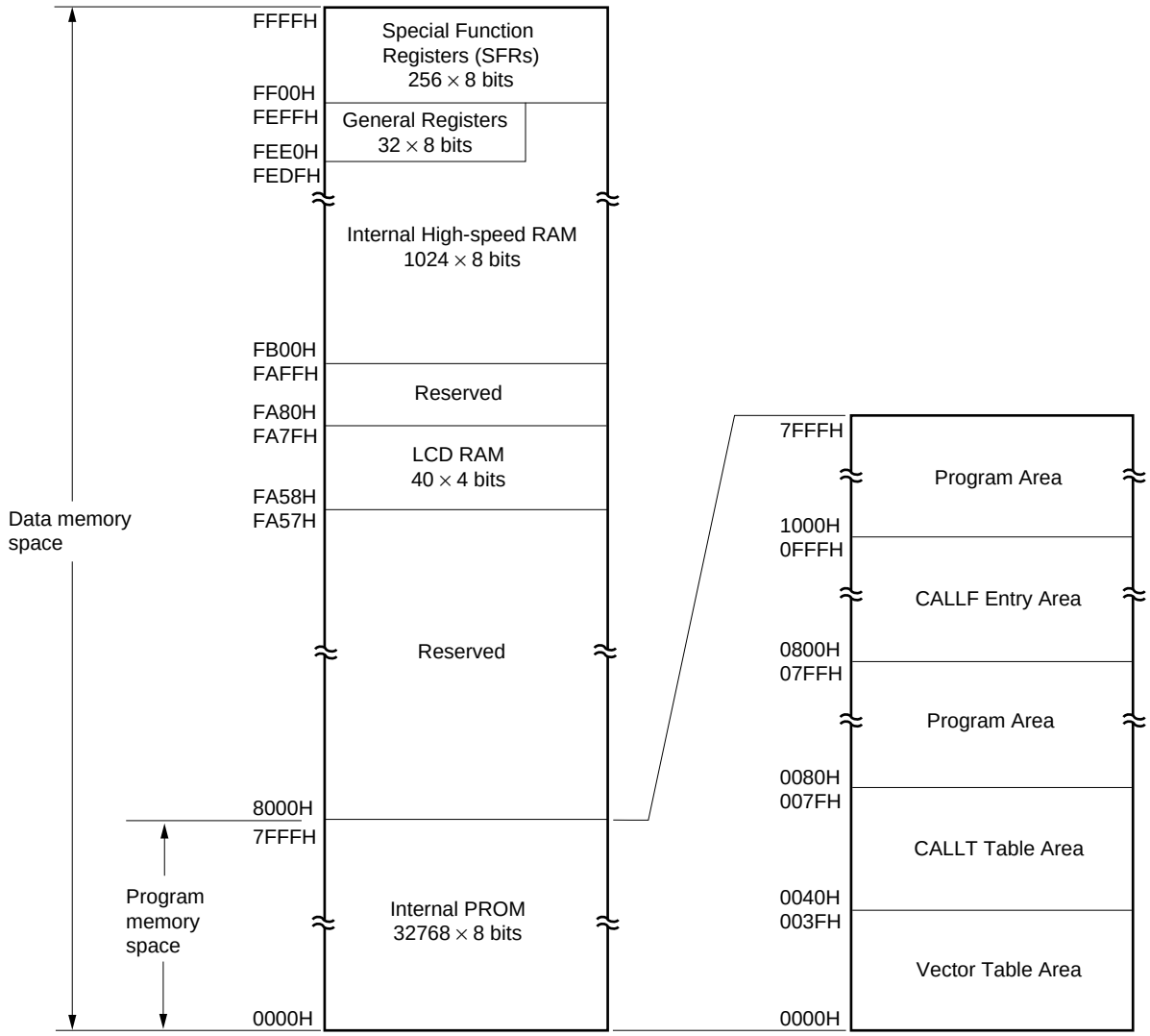


Figure 3-2. Memory Map (μPD78P064B)



3.1.1 Internal program memory space

The internal program memory space stores program data and table data. This space is generally accessed with program counter (PC).

The μPD78064B subseries has on chip ROM (or PROM) and the capacity of the memory varies depending on the part number.

Table 3-1. Internal ROM Capacity

Part Number	Internal ROM	
	Type	Capacity
μPD78064B	Mask ROM	32768 × 8 bits
μPD78P064B	PROM	

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 3-2. Vector Table

Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input
0004H	INTWDT
0006H	INTP0
0008H	INTP1
000AH	INTP2
000CH	INTP3
000EH	INTP4
0010H	INTP5
0014H	INTCSI0
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).

3.1.2 Internal data memory space

The μPD78064B subseries units incorporate the following RAMs.

(1) Internal high-speed RAM

The μPD78064B subseries has on-chip high-speed RAM, and the memory capacity varies depending on the part number as shown below.

Table 3-3. Internal High-Speed RAM Capacity

Part Number	Internal High-Speed RAM Capacity
μPD78064B	1024 × 8 bits
μPD78P064B	

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) LCD display RAM

Addresses FA58H to FA7FH of 40 × 4 bits are allocated for LCD display RAM, However, this area can also be used as general-purpose RAM.

3.1.3 Special Function Register (SFR) area

An on-chip peripheral hardware special-function register (SFR) is allocated in the area FF00H to FFFFH. (Refer to **3.2.3 Special function register (SFR: Special Function Register) Table 3-4 Special Function Register List**).

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

3.1.4 Data memory addressing

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

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

To address the memory to be manipulated when an instruction is executed, the μPD78064B subseries have many addressing modes to improve the 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. Figures 3-3 to 3-4 show the data memory addressing modes.

For details of each addressing, refer to **3.4 Operand Address Addressing**.

Figure 3-3. Data Memory Addressing (μPD78064B)

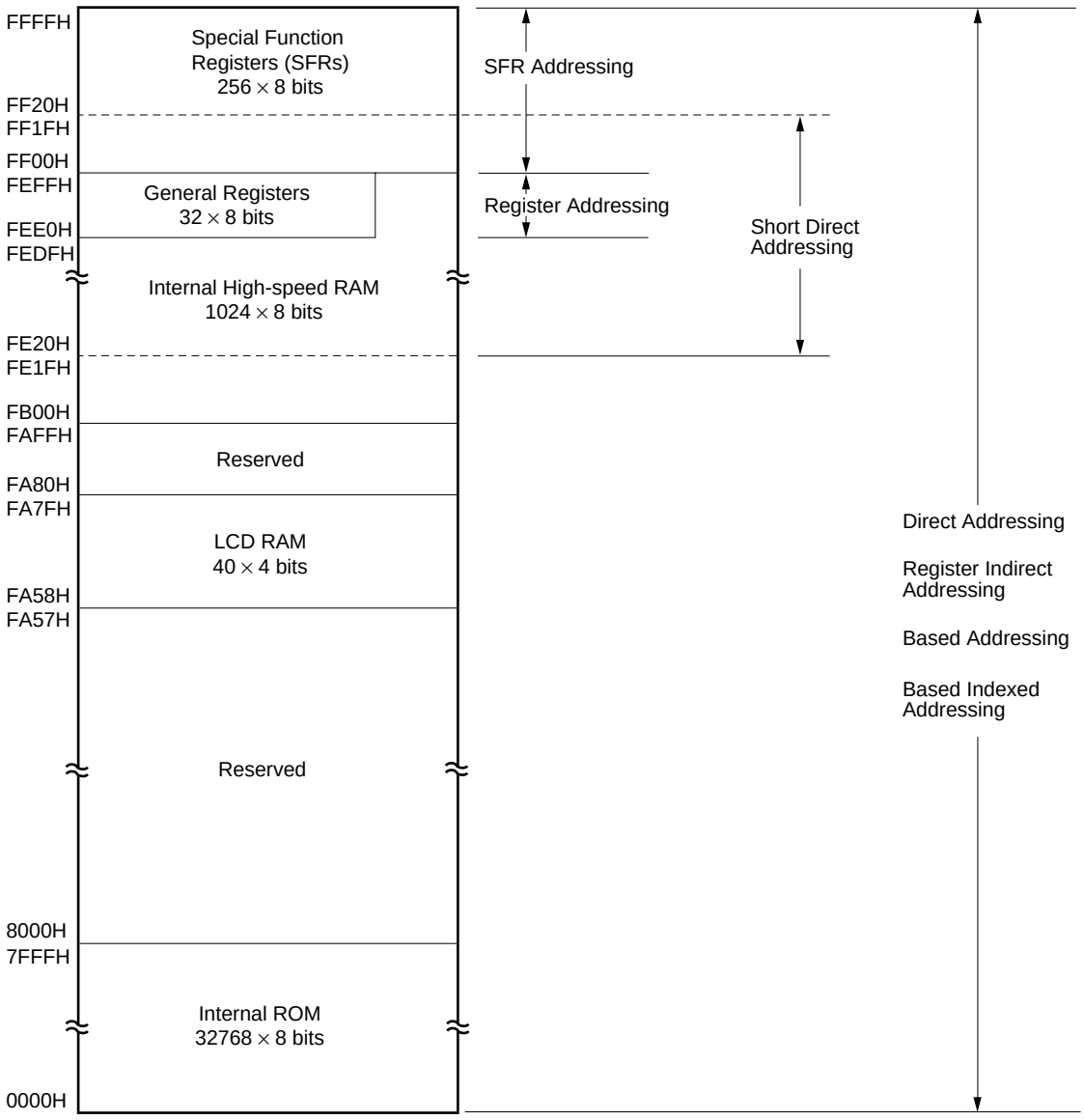
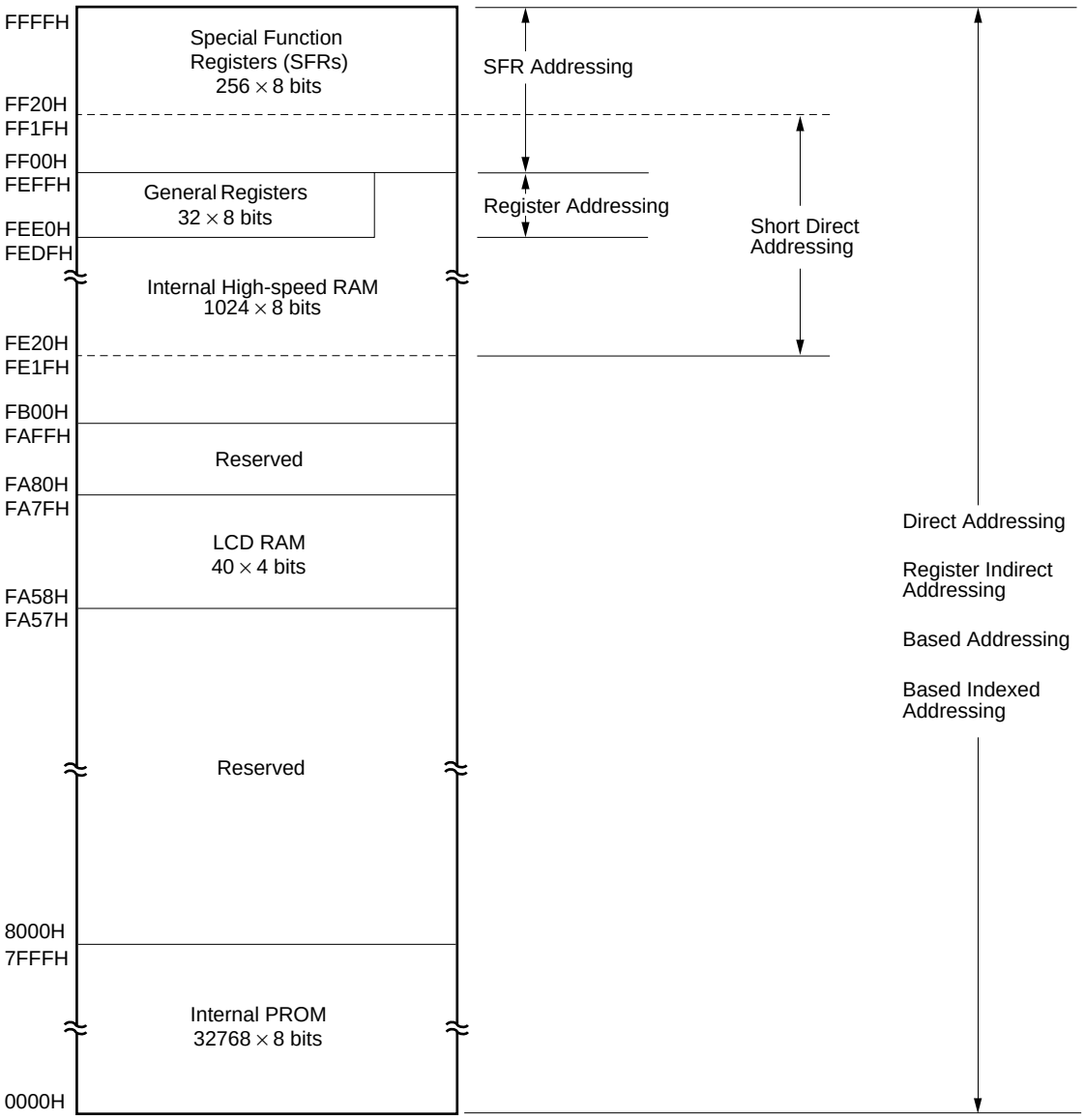


Figure 3-4. Data Memory Addressing (μPD78P064B)



3.2 Processor Registers

The μPD78064B subseries units incorporate the following processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

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

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

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

Figure 3-5. Program Counter Configuration

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

(2) Program status word (PSW)

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

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

Figure 3-6. Program Status Word Configuration

	7							0
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY

(a) Interrupt enable flag (IE)

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

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

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

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

(b) Zero flag (Z)

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

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

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

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

(d) Auxiliary carry flag (AC)

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

(e) In-service priority flag (ISP)

This flag manages the priority of acknowledgeable maskable vectored interrupts.

When ISP = 0, the vectored interrupt which is assigned a low priority by the priority specification flag registers (PR0L, PR0H, PR1L) (refer to **16.3 (3) Priority specification flag registers (PR0L, PR0H, PR1L)**) should not be accepted. Whether the interrupt is actually accepted 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-FEFFH) can be set as the stack area.

Figure 3-7. Stack Pointer Configuration

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

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

Each stack operation saves/restores data as shown in Figures 3-8 and 3-9.

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

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

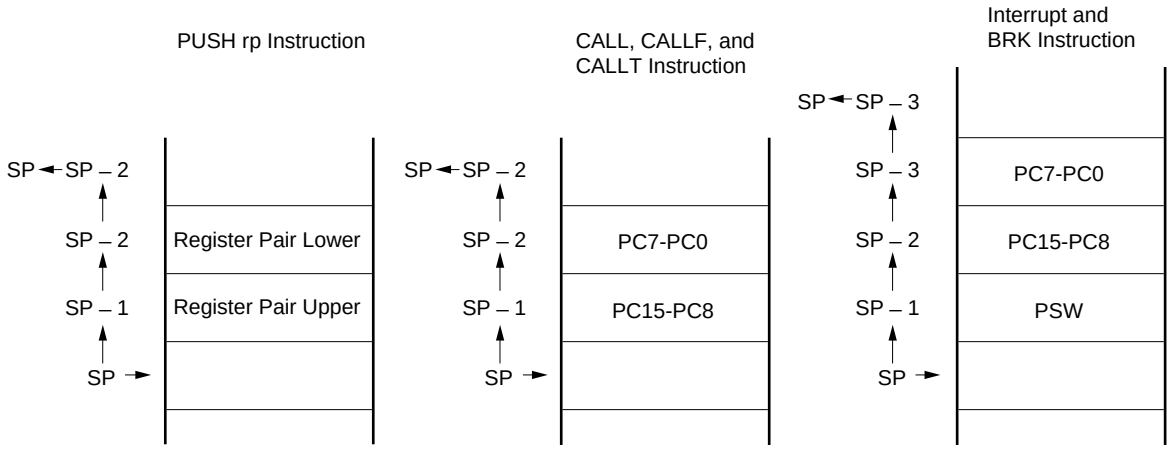
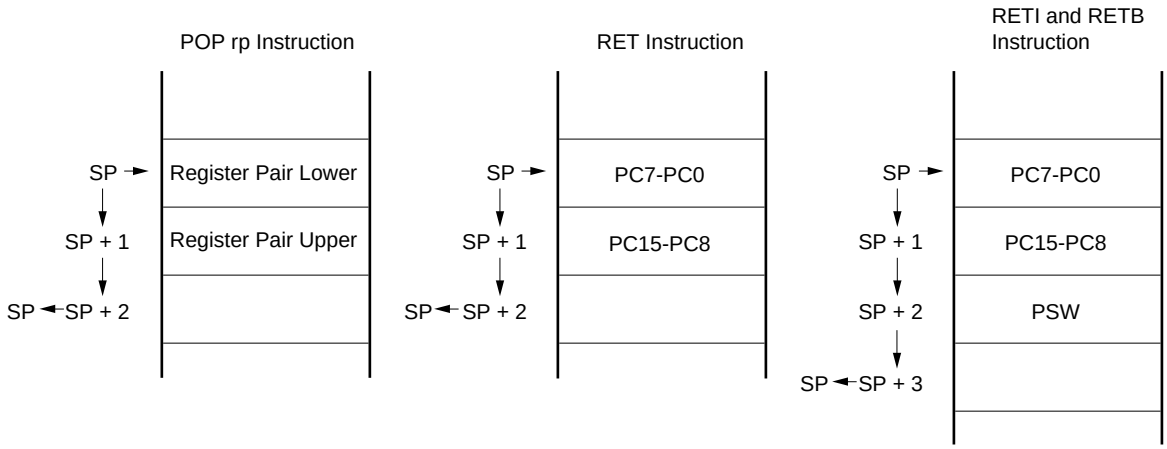


Figure 3-9. Data to be Restored from Stack Memory



3.2.2 General registers

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

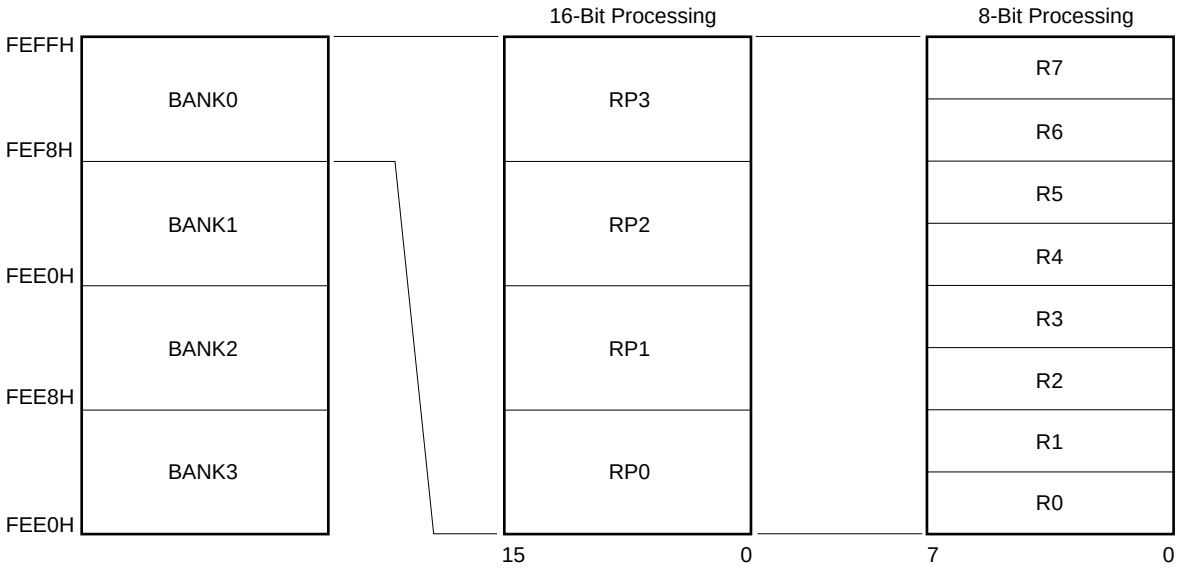
Each register can also be used as an 8-bit register. Two 8-bit registers can be used in pairs as a 16-bit register (AX, BC, DE and HL).

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

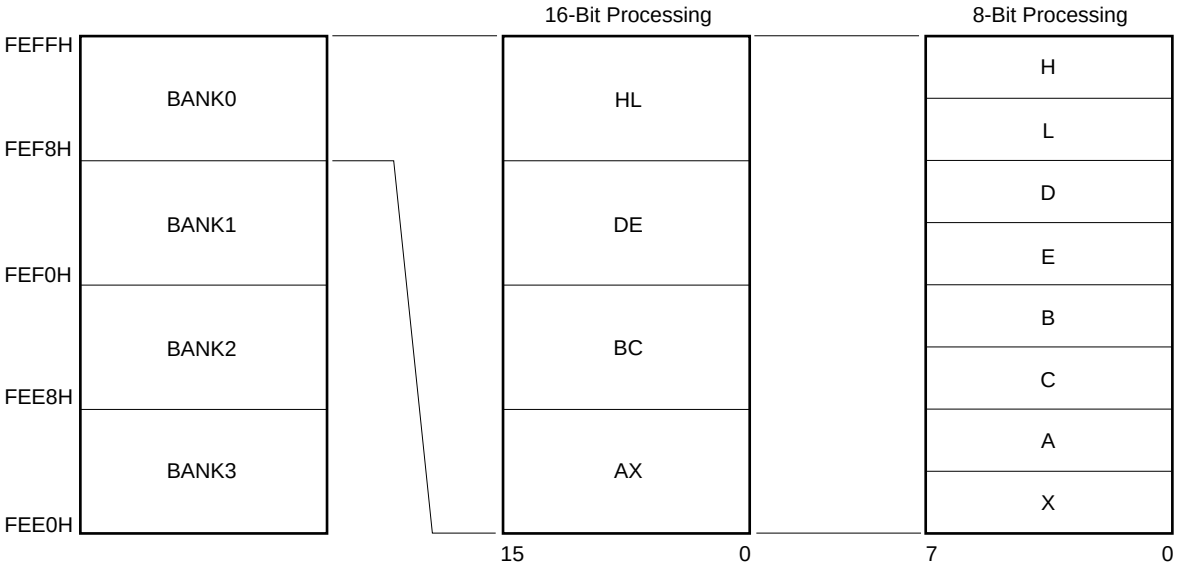
Register banks to be used for instruction execution are set with the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupt request for each bank.

Figure 3-10. General Register Configuration

(a) Absolute Name



(b) Function Name



3.2.3 Special Function Register (SFR)

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

It is allocated in the FF00H to FFFFH area.

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

Each manipulation bit unit can be specified as follows.

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

Table 3-4 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 registers. These symbols are the reserved words of the RA78K/0 and defined by header file sfrbit.h for CC78K/0.
  They can be described as the operands of instructions when the RA78K/0, ID78K0, and SD78K/0 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 unit**
  ○ indicates manipulatable bit units 1, 8, and 16. – indicates the bit unit that cannot be manipulated.
- **After reset**
  Indicates each register status upon $\overline{\text{RESET}}$ input.

Table 3-4. Special-Function Register List (1/2)

Address	Special-Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 bit	8 bits	16 bits	
FF00H	Port 0	P0	R/W	○	○	—	00H
FF01H	Port 1	P1		○	○	—	
FF02H	Port 2	P2		○	○	—	
FF03H	Port 3	P3		○	○	—	
FF07H	Port 7	P7		○	○	—	
FF08H	Port 8	P8		○	○	—	
FF09H	Port 9	P9		○	○	—	
FF0AH	Port 10	P10		○	○	—	
FF0BH	Port 11	P11		○	○	—	
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
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		○	○	—	
FF27H	Port mode register 7	PM7		○	○	—	
FF28H	Port mode register 8	PM8		○	○	—	
FF29H	Port mode register 9	PM9		○	○	—	
FF2AH	Port mode register 10	PM10		○	○	—	
FF2BH	Port mode register 11	PM11		○	○	—	
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		○	○	—	

Table 3-4. Special-Function Register List (2/2)

Address	Special-Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 bit	8 bits	16 bits	
FF49H	8-bit timer mode control register	TMC1		R/W	○	○	—	00H
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
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				
FF80H	A/D converter mode register	ADM		R/W	○	○	—	01H
FF84H	A/D converter input select register	ADIS			—	○	—	00H
FFB0H	LCD display mode register	LCDM			○	○	—	
FFB2H	LCD display control register	LCDC			○	○	—	
FFB8H	Key return mode register	KRM			○	○	—	02H
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			—	○	—	C8H
FFF2H	Oscillation mode selection register	OSMS		W	—	○	—	00H
FFF3H	Pull-up resistor option register H	PUOH		R/W	○	○	—	
FFF7H	Pull-up resistor option register L	PUOL			○	○	—	
FFF9H	Watchdog timer mode register	WDTM			○	○	—	
FFFAH	Oscillation stabilization time select register	OSTS			—	○	—	04H
FFFBH	Processor clock control register	PCC			○	○	—	

3.3 Instruction Address Addressing

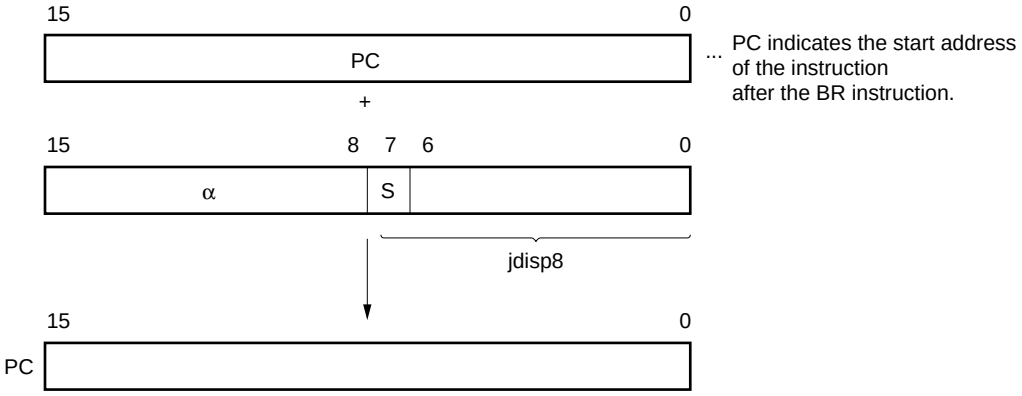
An instruction address is determined by program counter (PC) contents and the contents are normally incremented (+1 for each byte) automatically according to the number of bytes of an instruction to be fetched each time another instruction is executed. When a branch instruction is executed, the branch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

3.3.1 Relative Addressing

[Function]

The value obtained by adding 8-bit immediate data (displacement value: jdisp8) of an instruction code to the start address of the following instruction is transferred to the program counter (PC) and branched. The displacement value is treated as signed two's complement data (–128 to +127) and bit 7 becomes a sign bit. That is, using relative addressing, the program branches in the range –128 to +127 relative to the first address of the next instruction. This function is carried out when the BR $addr16 instruction or a conditional branch instruction is executed.

[Illustration]



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

3.3.2 Immediate addressing

[Function]

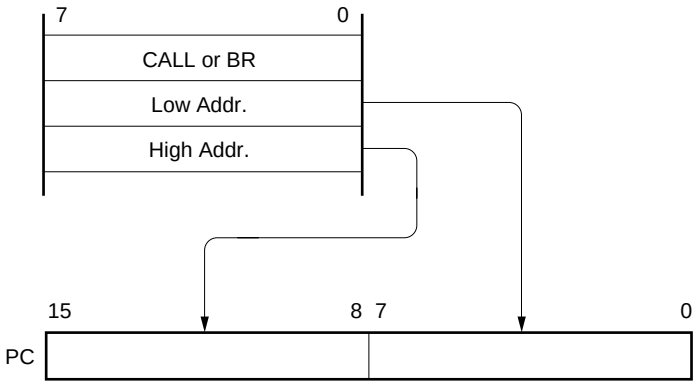
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

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

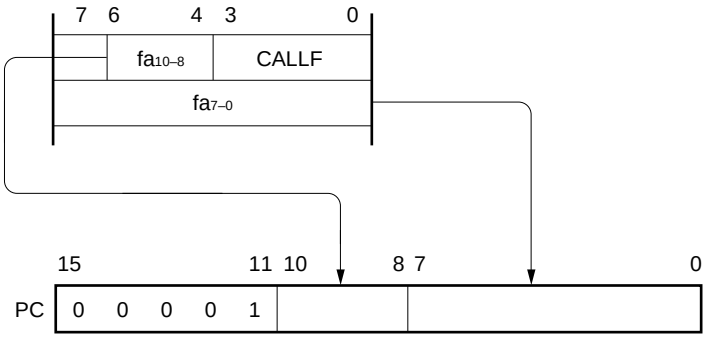
The CALL !addr16 and BR !addr16 instructions can be used to branch to any location in the memory. The CALLF !addr11 instruction is used to branch to the area between 0800H through 0FFFH.

[Illustration]

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



In the case of CALLF !addr11 instruction



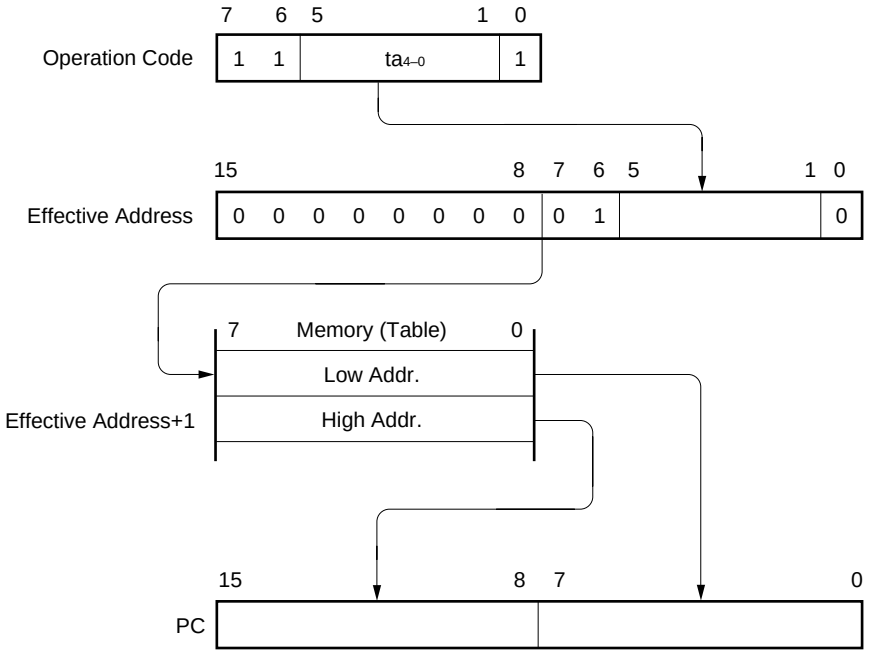
3.3.3 Table indirect addressing

[Function]

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

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

[Illustration]



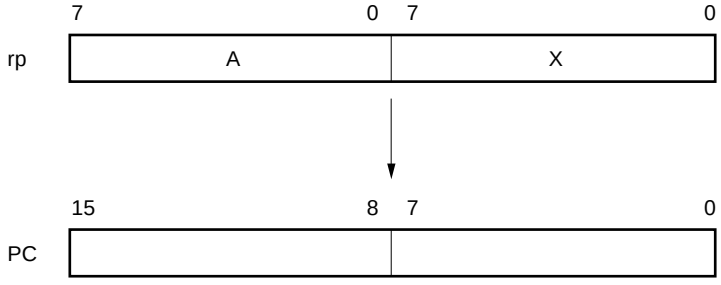
3.3.4 Register addressing

[Function]

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

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

[Illustration]



3.4 Operand Address Addressing

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

3.4.1 Implied addressing

[Function]

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

Of the μPD78064B subseries instruction words, the following instructions employ implied addressing.

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

[Operand format]

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

[Description example]

In the case of MULU X

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

3.4.2 Register addressing

[Function]

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

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

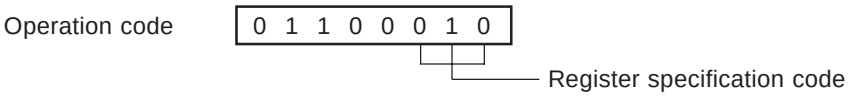
[Operand format]

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

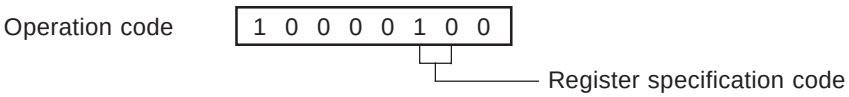
'r' and 'rp' can be described with function names (X, A, C, B, E, D, L, H, AX, BC, DE and HL) as well as absolute names (R0 to R7 and RP0 to RP3).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



3.4.3 Direct addressing

[Function]

This addressing is directly to address a memory indicating the immediate data in an instruction word as an operand address.

[Operand format]

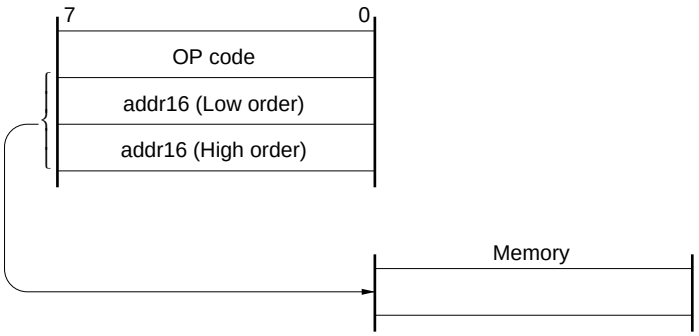
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

Operation code	1 0 0 0 1 1 1 0	Op code
	0 0 0 0 0 0 0 0	00H
	1 1 1 1 1 1 1 0	FEH

★ **[Illustration]**



3.4.4 Short direct addressing

[Function]

The memory to be manipulated in the fixed space is directly addressed with 8-bit data in an instruction word. The fixed area to which this addressing is applied is the 256-byte space FE20H to FF1FH. An internal high-speed RAM and a special-function register (SFR) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively. The SFR area (FF00H to FF1FH) where short direct addressing is applied is one part of all the SFR areas. In this area, ports which are frequently accessed in a program and a compare register of the timer/event counter and a capture register of the timer/event counter are mapped and these SFRs can be manipulated with a small number of bytes and clocks.

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

[Operand format]

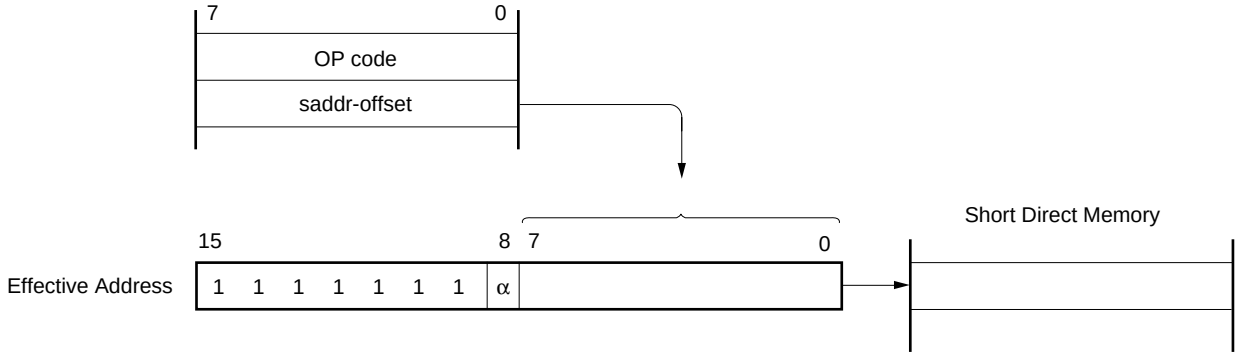
Identifier	Description
saddr	Label of FE20H to FF1FH immediate data
saddrp	Label of FE20H to FF1FH immediate data (even address only)

[Description example]

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

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

[Illustration]



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

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

3.4.5 Special-Function Register (SFR) addressing

[Function]

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

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

[Operand format]

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

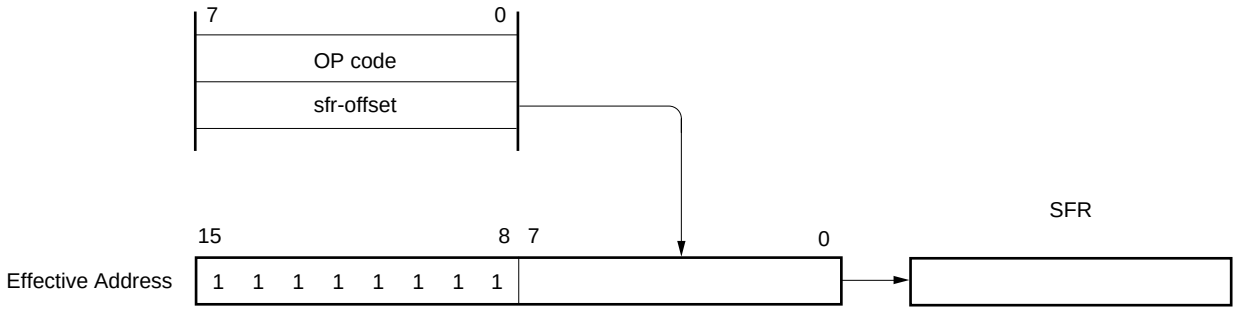
[Description example]

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

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

| 0 0 1 0 0 0 0 0 | 20H (sfr-offset)

[Illustration]



3.4.6 Register indirect addressing

[Function]

This addressing mode is used to address the memory by using 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 the register pair specification code in the operation code. This addressing can be carried out for all the memory spaces.

[Operand format]

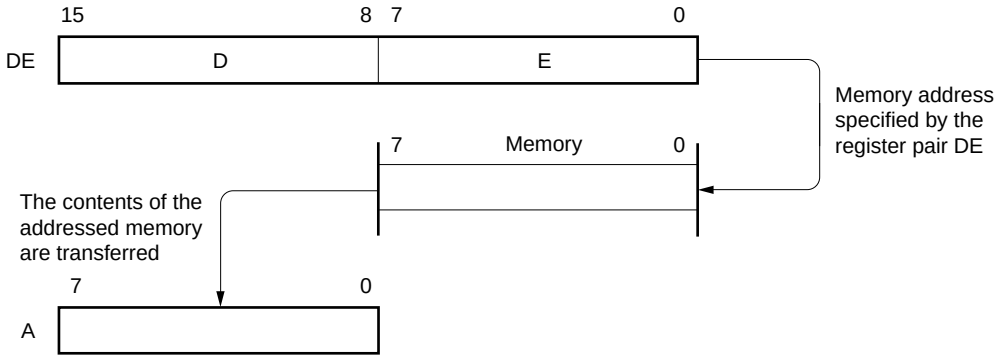
Identifier	Description
—	[DE], [HL]

[Description example]

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

Operation code	1	0	0	0	0	1	0	1

[Illustration]



3.4.7 Based addressing

[Function]

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

[Operand format]

Identifier	Description
—	[HL + byte]

[Description example]

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

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

3.4.8 Based indexed addressing

[Function]

This addressing mode is used to address the memory by using the result of adding the contents of the B or C register specified in the instruction word to the HL register pair as a base register. The HL, B, and C registers to be accessed are in the register bank specified by the register bank select flags (RBS0 and RBS1). The addition is executed by extending the contents of the B or C register to a 16-bit 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

3.4.9 Stack addressing

[Function]

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

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

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

[Description example]

In the case of PUSH DE

Operation code	1	0	1	1	0	1	0	1

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The μPD78064B subseries units incorporate two input ports and 55 input/output ports. Figure 4-1 shows the port configuration. Every port is capable of 1-bit and 8-bit manipulations and can carry out considerably varied control operations. Besides port functions, the ports can also serve as on-chip hardware input/output pins.

Figure 4-1. Port Types

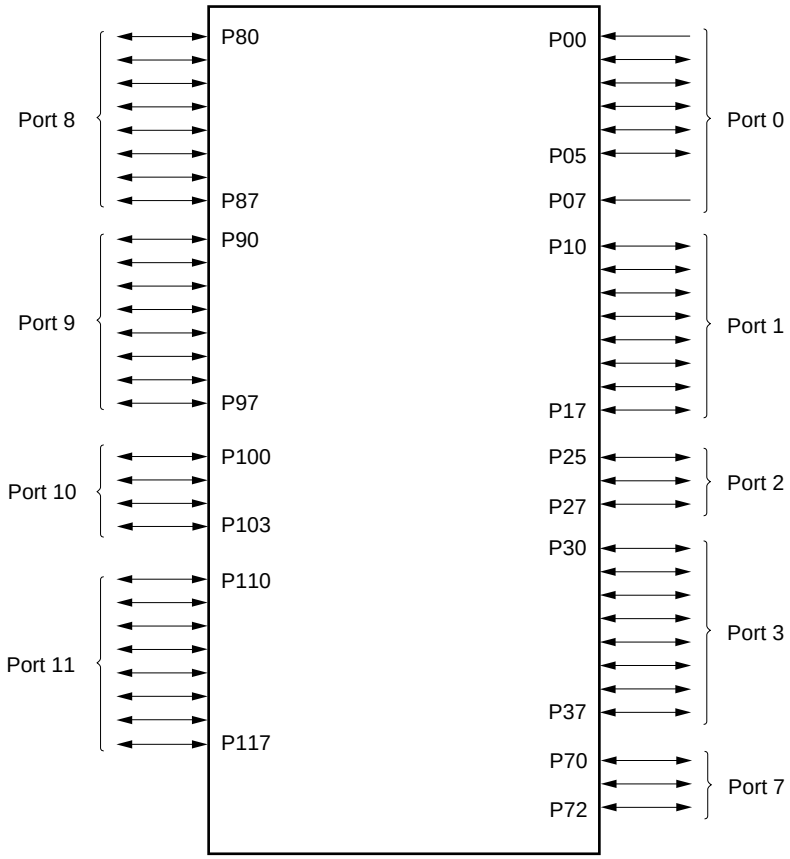


Table 4-1. Port Functions

Pin Name	Function		Dual-Function Pin
P00	Port 0.	Input only	INTP0/TI00
P01	7-bit input/output port.	Input/output mode can be specified bitwise.	INTP1/TI01
P02		If used as an input port, an internal pull-up	INTP2
P03		resistor can be used by software.	INTP3
P04			INTP4
P05			INTP5
P07		Input only	XT1
P10-P17	Port 1. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an internal pull-up resistor can be used by software.		ANI0-ANI7
P25	Port 2.		SI0/SB0
P26	3-bit input/output port. Input/output mode can be specified bit-wise.		SO0/SB1
P27	If used as an input port, an internal pull-up resistor can be used by software.		$\overline{\text{SCK0}}$
P30	Port 3.		TO0
P31	8-bit input/output port. Input/output mode can be specified bit-wise.		TO1
P32	If used as an input port, an internal pull-up resistor can be used by software.		TO2
P33			TI1
P34			TI2
P35			PCL
P36			BUZ
P37			—
P70	Port 7.		SI2/RxD
P71	3-bit input/output port. Input/output mode can be specified bit-wise.		SO2/TxD
P72	If used as an input port, an internal pull-up resistor can be used by software.		$\overline{\text{SCK2}}$/ASCK
P80-P87	Port 8. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an internal pull-up resistor can be used by software. This port can be used as a segment signal output port or an I/O port in 2-bit units by setting LCD display control register (LCDC).		S39-S32
P90-P97	Port 9. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an internal pull-up resistor can be used by software. This port can be used as a segment signal output port or an I/O port in 2-bit units by setting LCD display control register (LCDC).		S31-S24
P100-P103	Port 10. 4-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an internal pull-up resistor can be used by software. This port can directly drive LEDs.		—
P110-P117	Port 11. 8-bit input/output port. Input/output mode can be specified bit-wise. If used as an input port, an internal pull-up resistor can be used by software. Falling edge detection is possible.		—

Caution Do not perform the following operation on the pins shared with port pins during A/D conversion operation; otherwise, the specifications of the absolute accuracy during A/D conversion cannot be satisfied (except the pins shared with LCD segment output pins).

(1) Rewriting the output latch of an output pin used as a port pin

(2) Changing the output level of an output pin even when it is not used as a port pin

4.2 Port Configuration

A port consists of the following hardware:

Table 4-2. Port Configuration

Item	Configuration
Control register	Port mode register (PMm: m = 0 to 3, 7 to 11) Pull-up resistor option register (PUOH, PUOL) Key return mode register (KRM)
Port	Total: 57 ports (2 inputs, 55 inputs/outputs)
Pull-up resistor	Total: 55 (software specifiable: 55)

4.2.1 Port 0

Port 0 is a 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 internal pull-up resistor can be used to them in 5-bit units with a pull-up resistor option register L (PUOL).

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

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

Figures 4-2 and 4-3 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 4-2. Block Diagram of P00 and P07

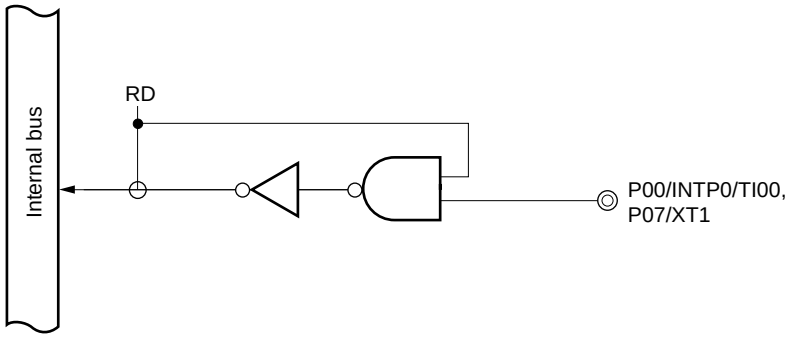
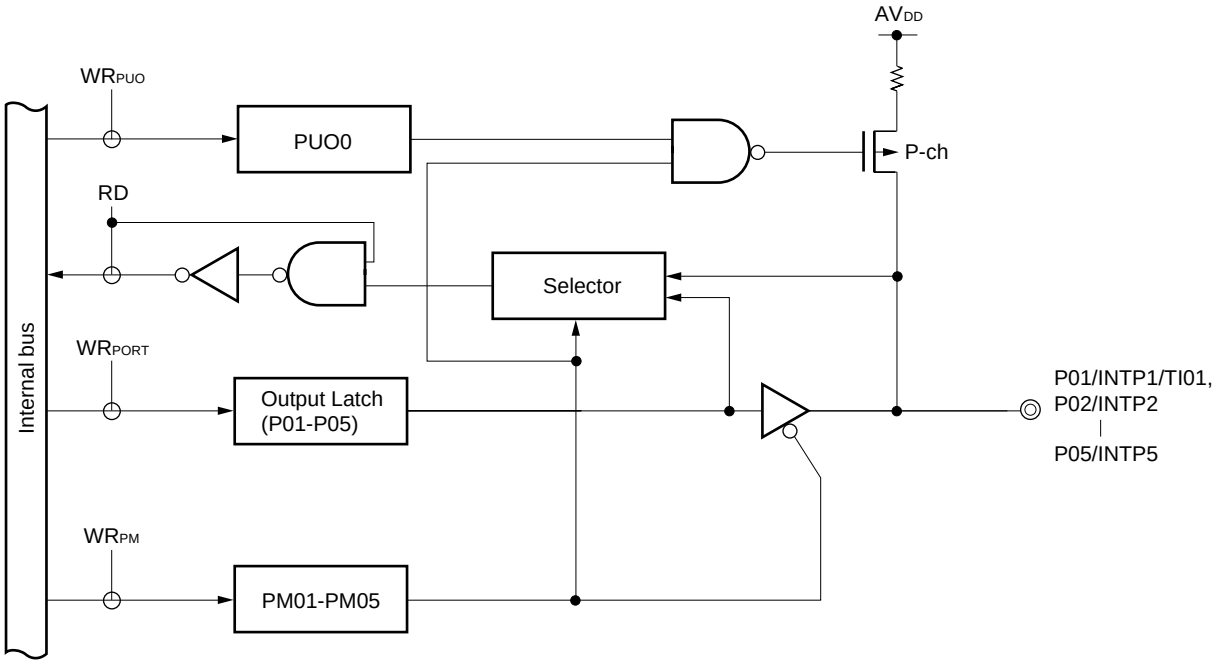


Figure 4-3. Block Diagram of P01 to P05



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

4.2.2 Port 1

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

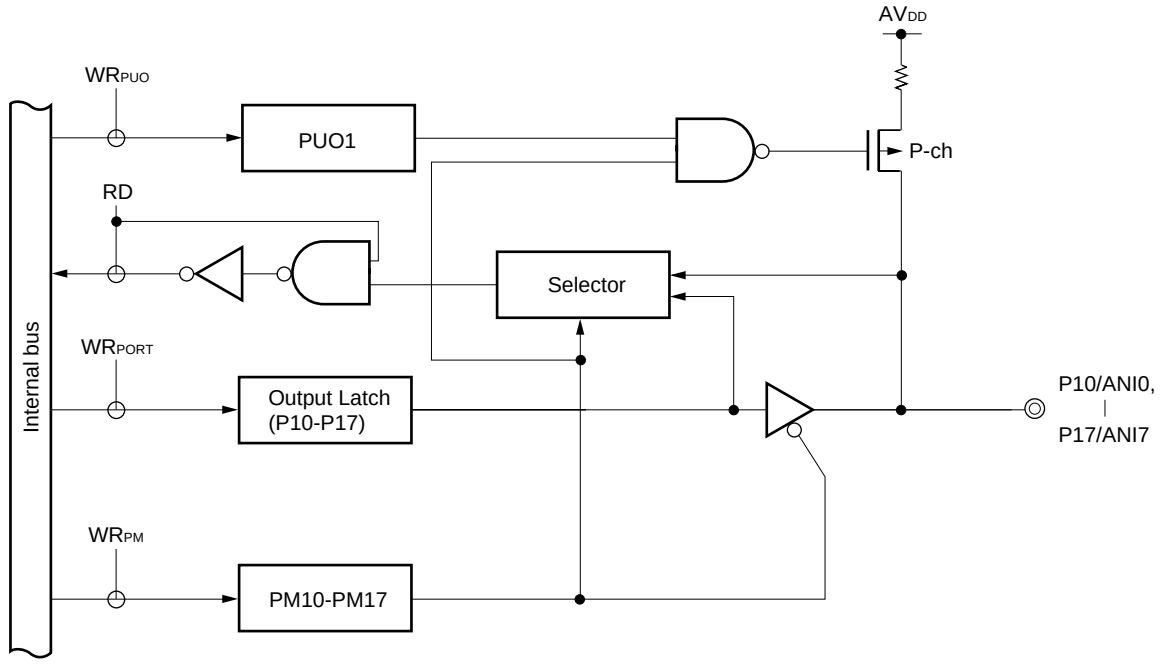
Dual-functions include an A/D converter analog input.

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

Figure 4-4 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 4-4. Block Diagram of P10 to P17



PUO : Pull-up resistor option register
PM : Port mode register
RD : Port 1 read signal
WR : Port 1 write signal

4.2.3 Port 2

Port 2 is a 3-bit input/output port with output latch. P25 to P27 pins can specify the input mode/output mode in 1-bit units with the port mode register 2 (PM2). When P25 to P27 pins are used as input ports, an internal pull-up resistor can be used to them in 3-bit units with a pull-up resistor option register L (PUOL).

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

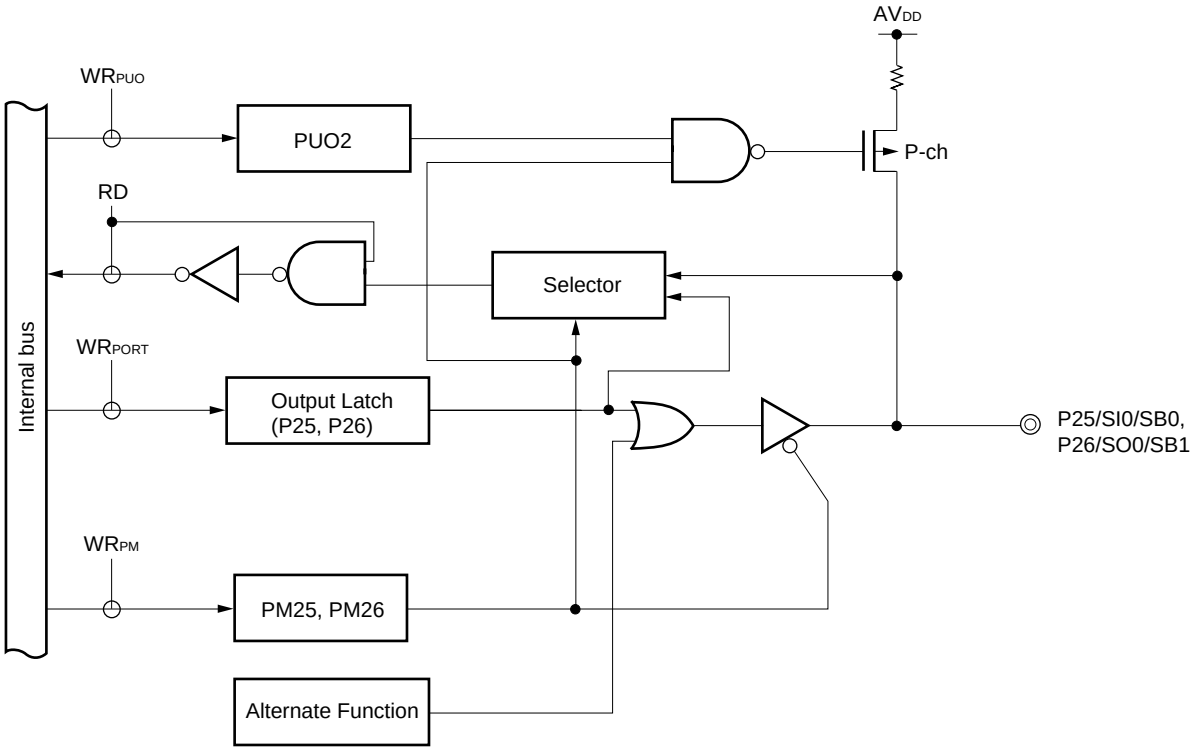
$\overline{\text{RESET}}$ input sets port 2 to input mode.

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

Cautions 1. When used as a serial interface, set the input/output and output latch according to its functions. For the setting method, refer to Figure 13-4 Serial Operating Mode Register 0 Format.

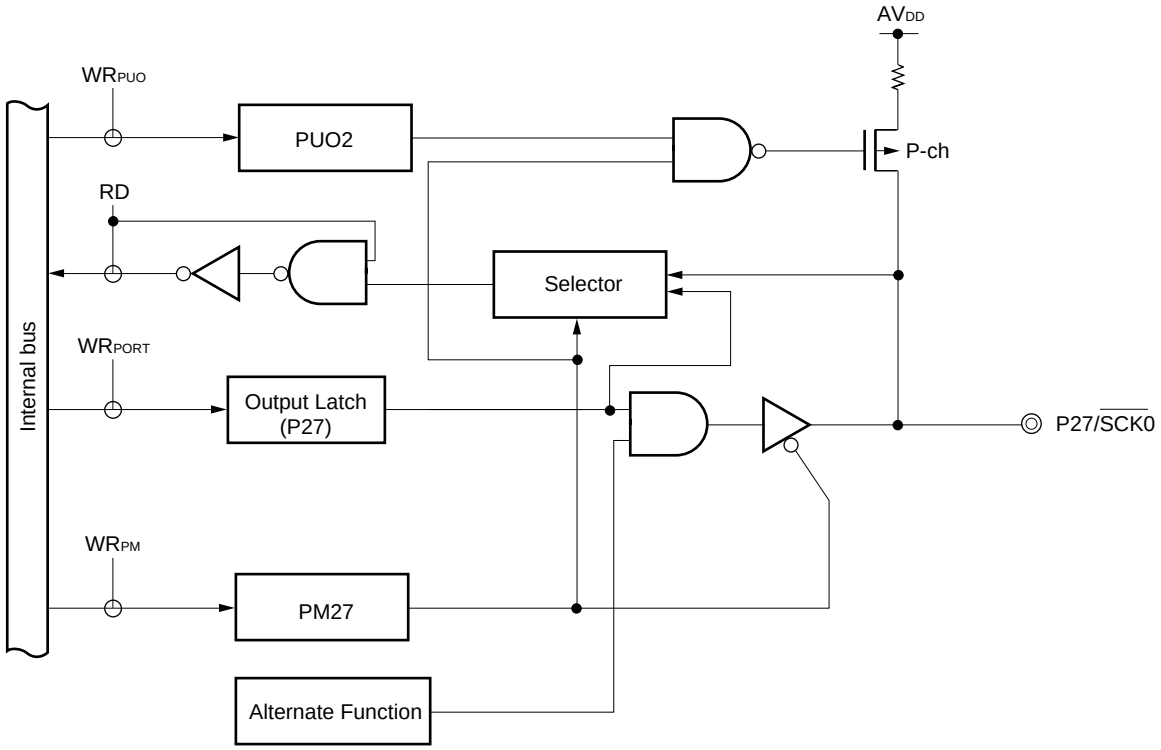
2. When reading the pin state in SBI mode, set PM2n bit of PM2 to 1 (n = 5, 6) (Refer to the description of (10) Method to judge busy state of a slave in 13.4.3 SBI mode operation).

Figure 4-5. Block Diagram of P25 and P26



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

Figure 4-6. Block Diagram of P27



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

4.2.4 Port 3

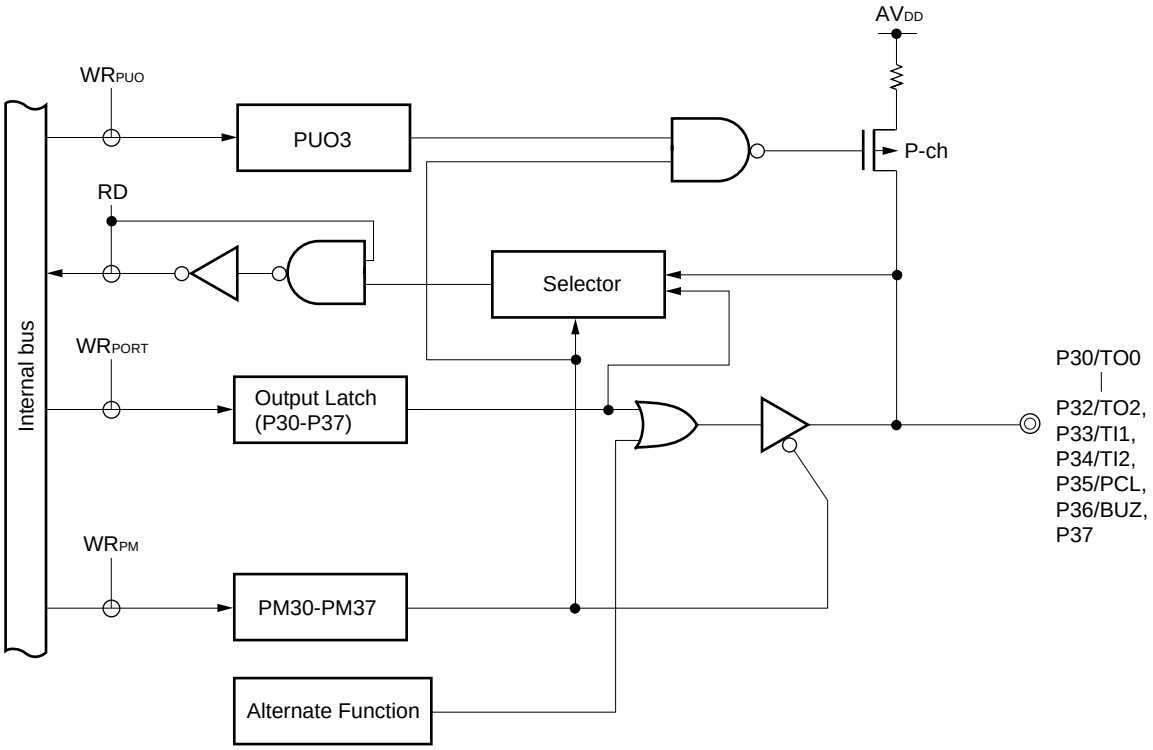
Port 3 is an 8-bit input/output port with output latch. P30 to P37 pins can specify the input mode/output mode in 1-bit units with the port mode register 3 (PM3). When P30 to P37 pins are used as input ports, an internal pull-up resistor can be used to them in 8-bit units with a pull-up resistor option 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 4-7 shows a block diagram of port 3.

Figure 4-7. Block Diagram of P30 to P37



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

4.2.5 Port 7

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 internal pull-up resistor can be used as a 3-bit unit by means of pull-up resistor option register L (PUOL).

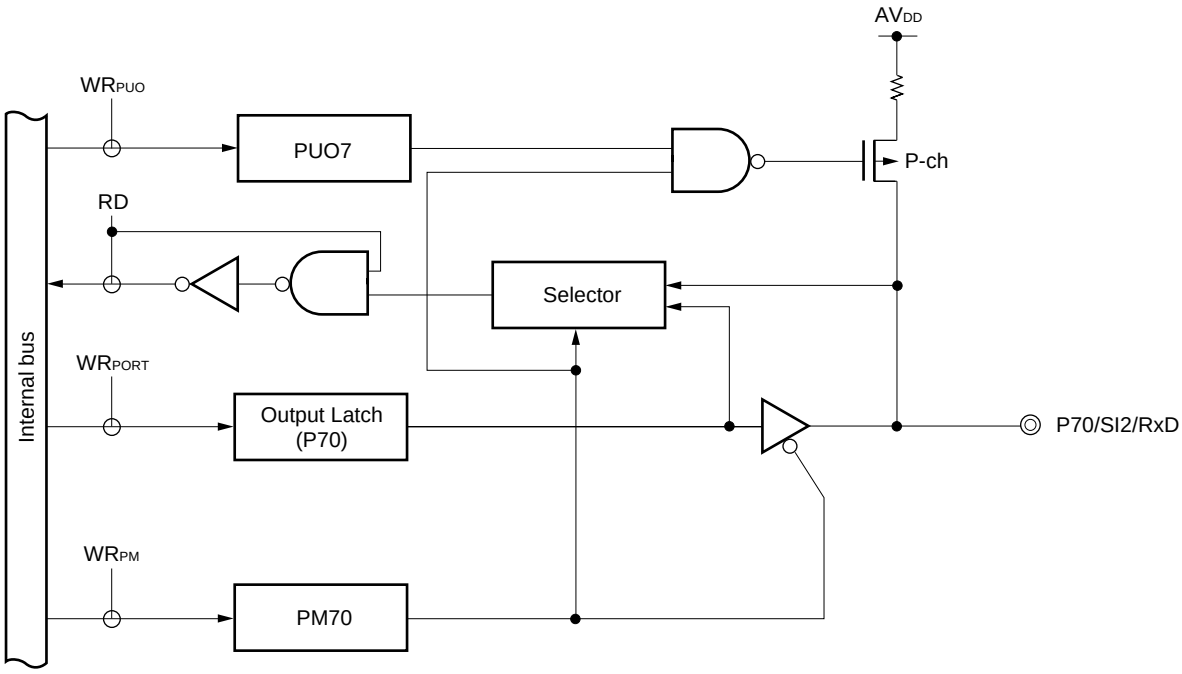
Dual-functions include serial interface channel 2 data input/output and clock input/output.

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

Port 7 block diagrams are shown in Figures 4-8 and 4-9.

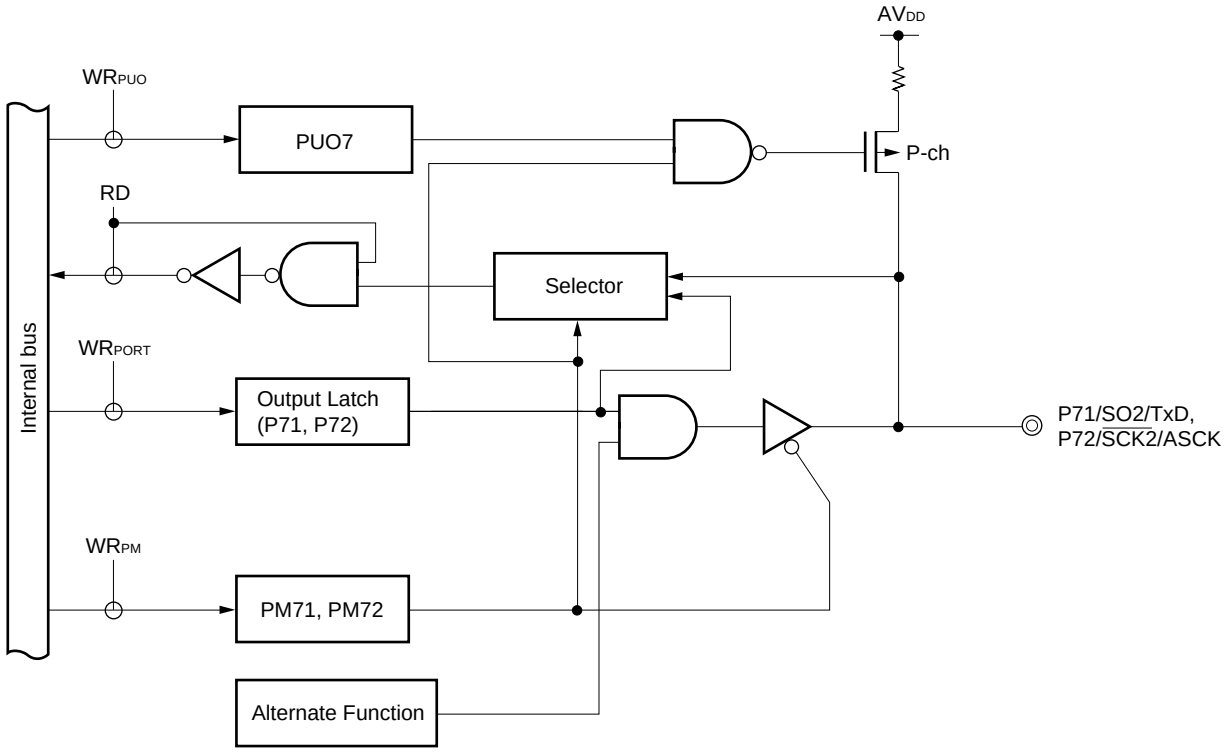
Caution When used as a serial interface, set the input/output and output latch according to its functions. For the setting method, refer to Table 14-2 Serial Interface Channel 2 Operating Mode Settings.

Figure 4-8. Block Diagram of P70



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

Figure 4-9. Block Diagram of P71 and P72



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

4.2.6 Port 8

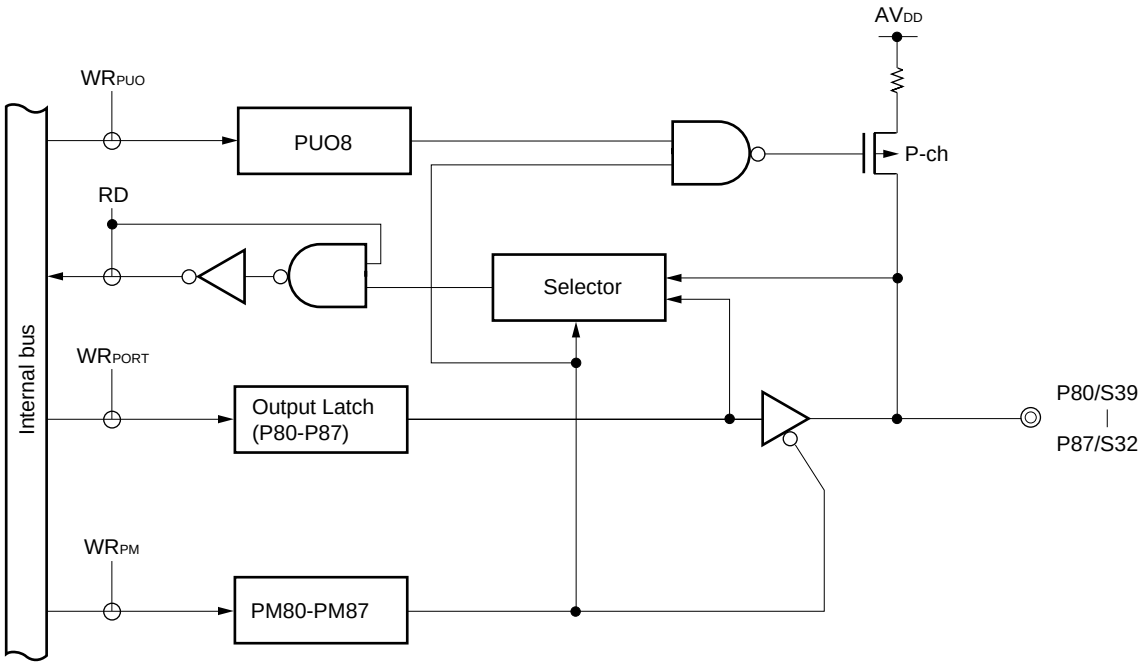
This is an 8-bit input/output port with output latches. Input mode/output mode can be specified bit-wise by means of port mode register 8 (PM8). When pins P80 to P87 are used as input port pins, an internal pull-up resistor can be used as an 8-bit unit by means of pull-up resistor option register H (PUOH).

These pins are dual-function pins and serve as LCD controller/driver segment signal outputs.

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

The port 8 block diagram is shown in Figure 4-10.

Figure 4-10. Block Diagram of P80 to P87



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

4.2.7 Port 9

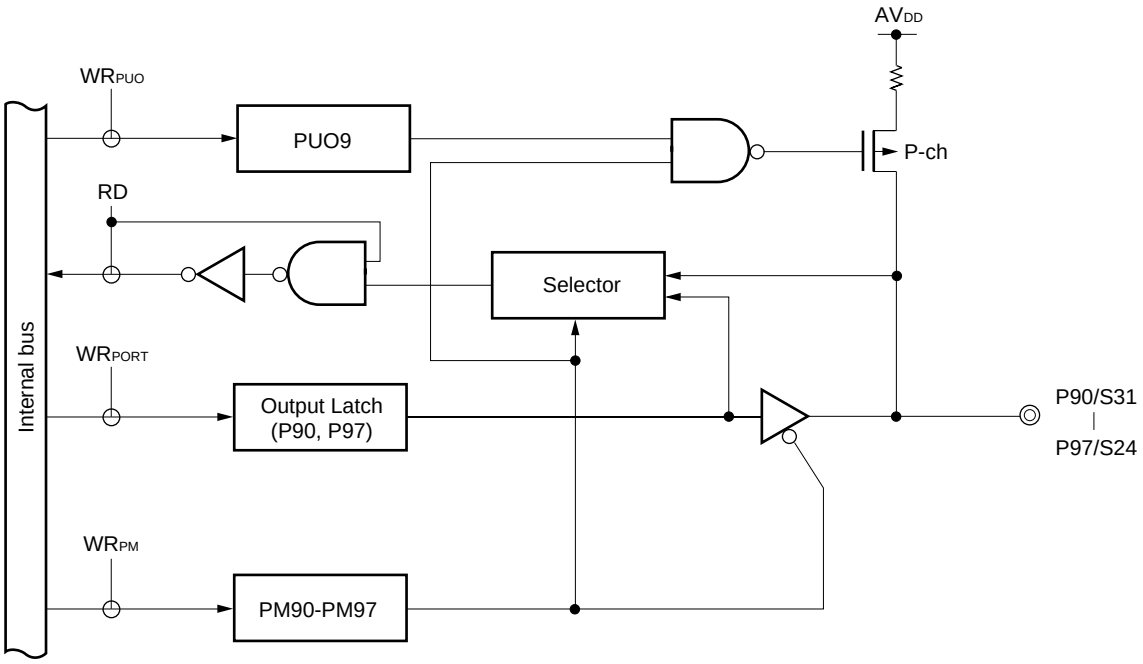
This is an 8-bit input/output port with output latches. Input mode/output mode can be specified bit-wise by means of port mode register 9 (PM9). When pins P90 to P97 are used as input port pins, an internal pull-up resistor can be used as a 8-bit unit by means of pull-up resistor option register H (PUOH).

These pins are dual-function pins and serve as LCD controller/driver segment signal outputs.

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

The port 9 block diagram is shown in Figure 4-11.

Figure 4-11. Block Diagram of P90 to P97



PUO : Pull-up resistor option register
PM : Port mode register
RD : Port 9 read signal
WR : Port 9 write signal

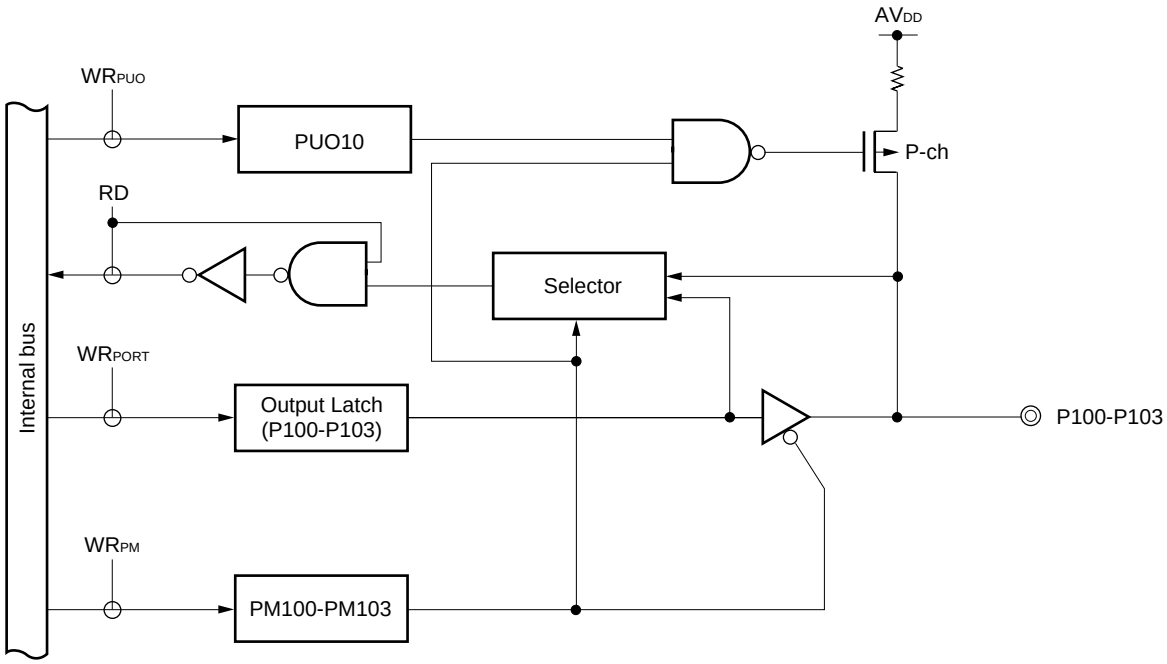
4.2.8 Port 10

This is a 4-bit input/output port with output latches. Input mode/output mode can be specified bit-wise by means of port mode register 10 (PM10). When pins P100 to P103 are used as input port pins, an internal pull-up resistor can be used as a 4-bit unit by means of pull-up resistor option register H (PUOH).

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

The port 10 block diagram is shown in Figure 4-12.

Figure 4-12. Block Diagram of P100 to P103



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

4.2.9 Port 11

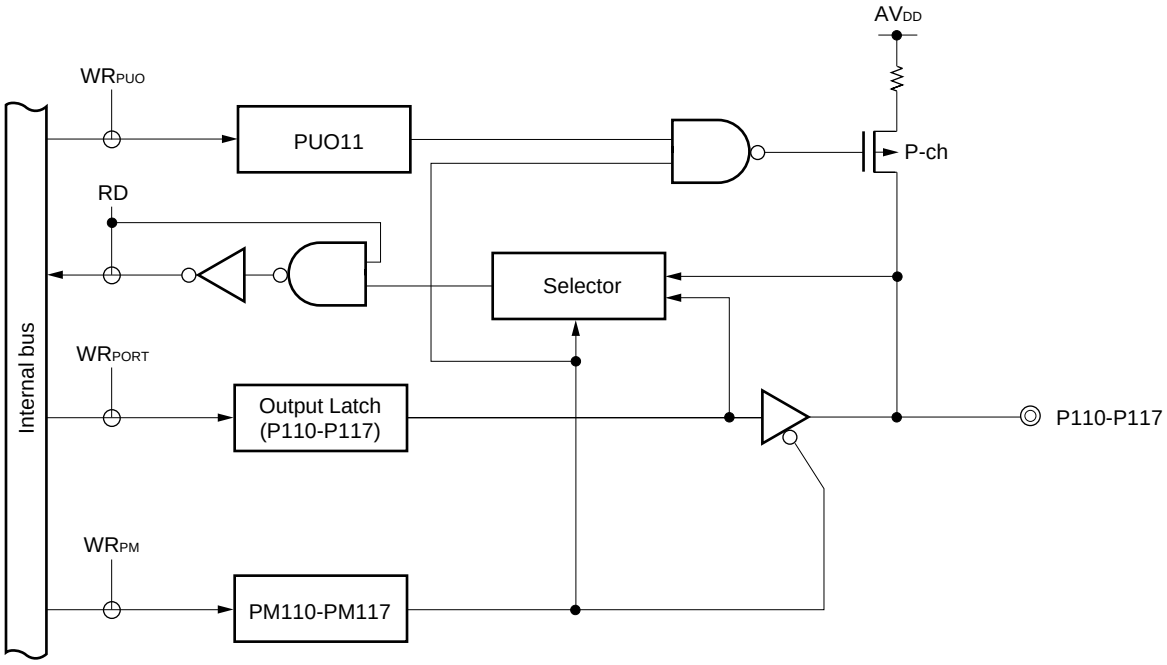
Port 11 is an 8-bit input/output port with output latches. P110 to P117 pins can specify the input mode/output mode in 8-bit units with the port mode register 11 (PM11). When they are used as input ports, an internal pull-up resistor can be used to them in 8-bit units with pull-up resistor option register H (PUOH).

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

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

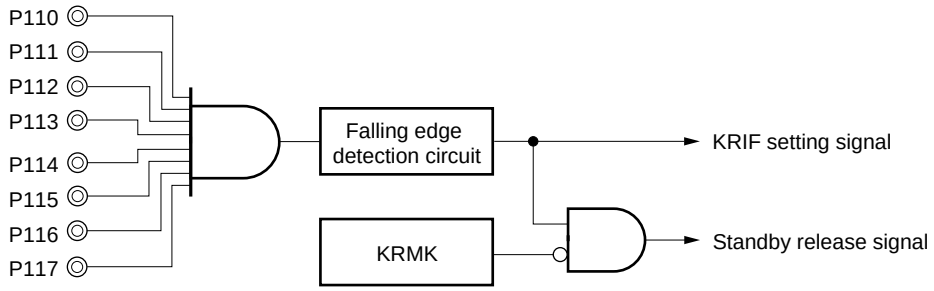
Figures 4-13 and 4-14 show the block diagrams of port 11 and falling edge detection circuit, respectively.

Figure 4-13. Block Diagram of P110 to P117



PUO : Pull-up resistor option register
MM : Memory expansion mode register
RD : Port 11 read signal
WR : Port 11 write signal

Figure 4-14. Block Diagram of Falling Edge Detection Circuit



KRIF : test input flag
KRMK : test mask flag

4.3 Port Function Control Registers

The following three types of registers control the ports.

- Port mode registers (PM0 to PM3, PM7 to PM11)
- Pull-up resistor option register (PUOH, PUOL)
- Key return mode register (KRM)

(1) Port mode registers (PM0 to PM3, PM7 to PM11)

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

PM0 to PM3 and PM7 to PM11 are independently set with a 1-bit or 8-bit memory manipulation instruction $\overline{\text{RESET}}$ input sets registers to FFH.

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

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

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

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

Pin Name	Dual-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-P05	INTP2-INTP5	Input	1	×
P07[Note1]	XT1	Input	1 (Fixed)	None
P10-P17[Note1]	ANI0-ANI7	Input	1	×
P30-P32	TO0-TO2	Output	0	0
P33, P34	TI1, TI2	Input	1	×
P35	PCL	Output	0	0
P36	BUZ	Output	0	0
P80-P87	S39-S32	Output	×[Note 2]	
P90-P97	S31-S24	Output	×[Note 2]	

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

2. When the P80 to P87 and P90 to P97 pins are used for dual functions, set the function by the LCD display control register (LCDC).

Caution 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, refer to Figure 13-4 Serial Operating Mode Register 0 Format and Table 14-2 Serial Interface Channel 2 Operating Mode Settings.

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

Figure 4-15. 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	1	1	1	1	1	FF22H	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM7	1	1	1	1	1	PM72	PM71	PM70	FF27H	FFH	R/W
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80	FF28H	FFH	R/W
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90	FF29H	FFH	R/W
PM10	1	1	1	1	PM103	PM102	PM101	PM100	FF2AH	FFH	R/W
PM11	PM117	PM116	PM115	PM114	PM113	PM112	PM111	PM110	FF2BH	FFH	R/W

PMmn	Pmn Pin Input/Output Mode Selection (m = 0-3, 7-11 : n = 0-7)
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 internal pull-up resistor use has been specified with PUOH, PUOL. No pull-up resistors can be used to the bits set to the output mode or to the bits used as an analog input pin, irrespective of PUOH or PUOL setting.

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

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

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

2. When ports 1, 8, and 9 are used as dual-function pins, an internal pull-up resistor cannot be used even if 1 is set in PUOm (m = 1, 8, 9).

Figure 4-16. Pull-Up Resistor Option Register Format

Symbol	7	6	5	4	③	②	①	⓪	Address	After Reset	R/W
PUOH	0	0	0	0	PUO11	PUO10	PUO9	PUO8	FFF3H	00H	R/W
	⑦	6	5	4	③	②	①	⓪			
PUOL	PUO7	0	0	0	PUO3	PUO2	PUO1	PUO0	FFF7H	00H	R/W

PUOm	Pm Internal Pull-up Resistor Selection (m = 0-3, 7-11)
0	Internal pull-up resistor not used
1	Internal pull-up resistor used

Caution Zeros must be set to bits 4 to 7 of PUOH and bit 4 to 6 of PUOL.

(3) Key return mode register (KRM)

This register sets enabling/disabling of standby function release by a key return signal (falling edge detection of port 11), and selects the port 11 falling edge input.

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

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

Figure 4-17. Key Return Mode Register Format

Symbol	7	6	5	4	3	2	①	⓪	Address	After Reset	R/W
KRM	0	0	0	0	KRM3	KRM2	KRMK	KRIF	FFB8H	02H	R/W

KRIF	Key Return Signal Detection Flag
0	Not Detected
1	Detected (Falling edge detection of port 11)

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

KRM3	KRM2	Selection of Port 11 Falling Edge Input
0	0	P117
0	1	P114-P117
1	0	P112-P117
1	1	P110-P117

Caution When falling edge detection of port 11 is used, KRIF should be cleared to 0 (not cleared to 0 automatically).

4.4 Port Function Operations

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

4.4.1 Writing to input/output port

(1) Output mode

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

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

(2) Input mode

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

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

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

4.4.2 Reading from input/output port

(1) Output mode

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

(2) Input mode

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

4.4.3 Operations on input/output port

(1) Output mode

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

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

(2) Input mode

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

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

[MEMO]

CHAPTER 5 CLOCK GENERATOR

5.1 Clock Generator Functions

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

(1) Main system clock oscillator

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

(2) Subsystem clock oscillator

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

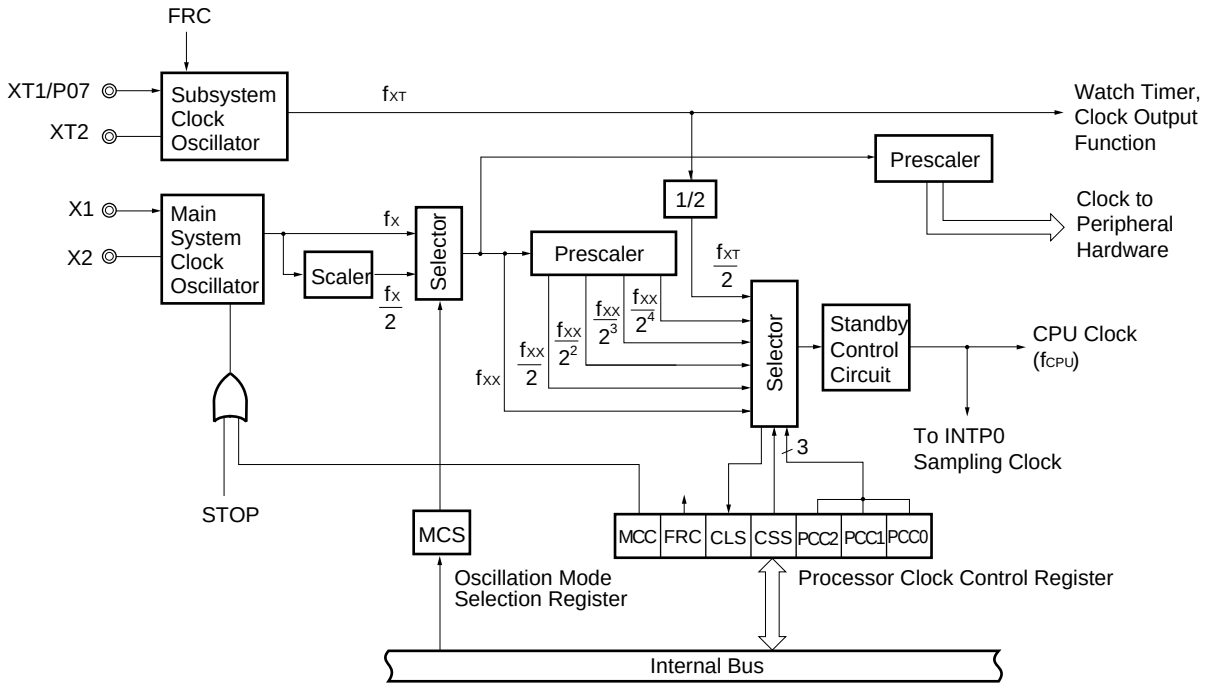
5.2 Clock Generator Configuration

The clock generator consists of the following hardware.

Table 5-1. Clock Generator Configuration

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

Figure 5-1. Block Diagram of Clock Generator



5.3 Clock Generator Control Register

The clock generator is controlled by the following two registers:

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

(1) Processor clock control register (PCC)

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

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

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

Figure 5-2. Subsystem Clock Feedback Resistor

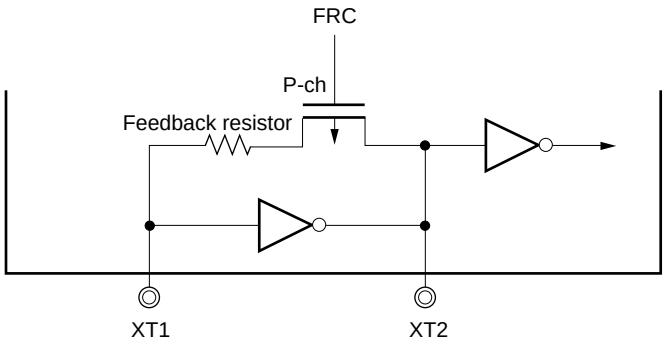


Figure 5-3. Processor Clock Control Register Format

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

R/W

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

R

CLS	CPU Clock Status
0	Main system clock
1	Subsystem clock

R/W

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

R/W

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

Notes 1. Bit 5 is Read Only.

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

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 μPD78064B 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 5-2.

Table 5-2. Relation between CPU Clock and Minimum Instruction Execution Time

CPU Clock (f_{CPU})	Minimum Instruction Execution Time: $2/f_{CPU}$
f_X	0.4 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}	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 8-bit memory manipulation instruction.

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

Figure 5-4. Oscillation Mode Selection Register Format

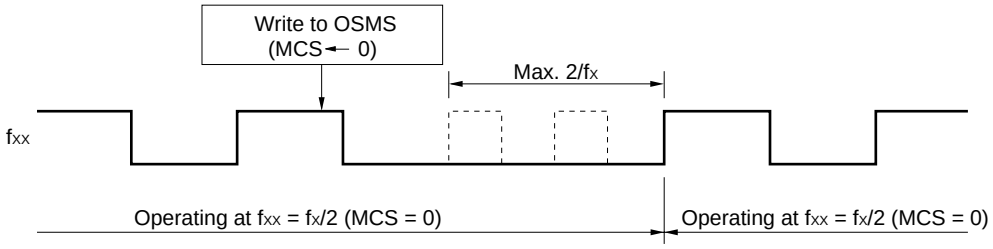
Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
OSMS	0	0	0	0	0	0	0	MCS	FFF2H	00H	W

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 5-5 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.

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

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



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

f_X : Main system clock oscillation frequency

5.4 System Clock Oscillator

5.4.1 Main system clock oscillator

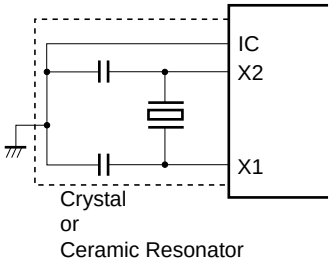
The main system clock oscillator oscillates with a crystal resonator or a ceramic resonator (5.0 MHz TYP.) connected to the X1 and X2 pins.

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.

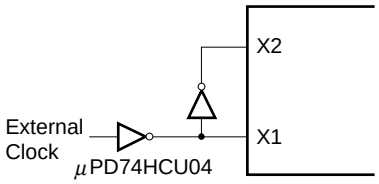
Figure 5-6 shows an external circuit of the main system clock oscillator.

Figure 5-6. 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 bit 7 (MCC) of the processor clock control register (PCC) to 1 if an external clock is used. This is because the X2 pin is connected to V_{DD} via a pull-up resistor.

5.4.2 Subsystem clock oscillator

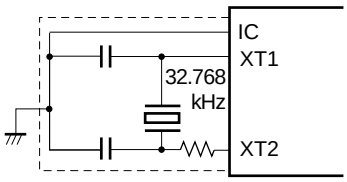
The subsystem clock oscillator oscillates with a crystal resonator (32.768 kHz TYP.) connected to the XT1 and XT2 pins.

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

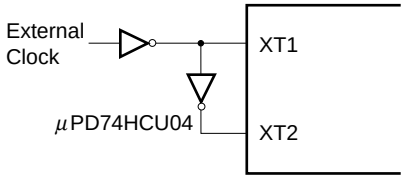
Figure 5-7 shows an external circuit of the subsystem clock oscillator.

Figure 5-7. External Circuit of Subsystem Clock Oscillator

(a) Crystal oscillation



(b) External clock



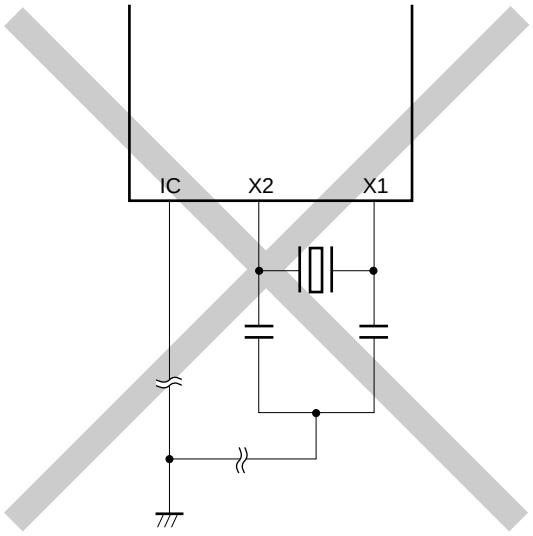
Cautions 1. When using a main system clock oscillator and a subsystem clock oscillator, carry out wiring in the broken line area in Figures 5-6 and 5-7 to prevent any effects from wiring capacities.

- **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_{SS}. Do not ground to any ground pattern where high current is present.**
- **Do not fetch signals from the oscillator.**

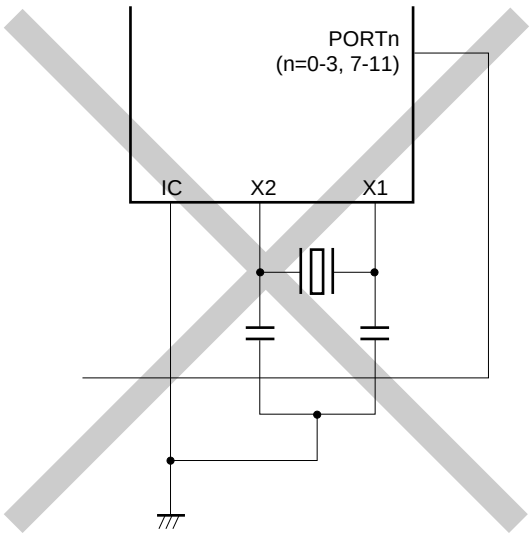
Take special note of the fact that the subsystem clock oscillator is a circuit with low-level amplification so that current consumption is maintained at low levels.
Figure 5-8 shows examples of resonator having bad connection.

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

(a) Wiring of connection circuits is too long



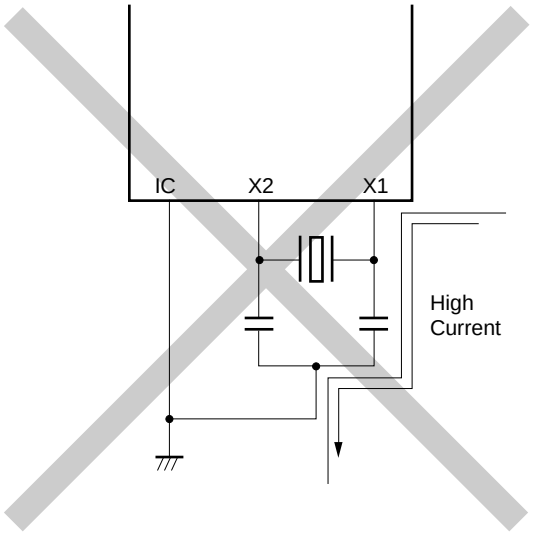
(b) Signal conductors intersect each other



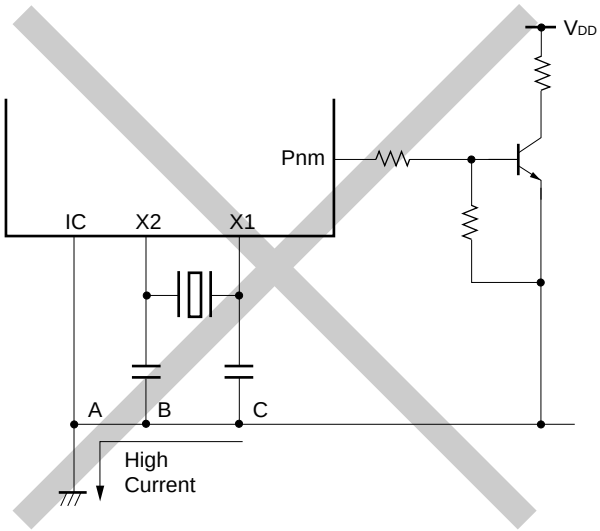
Remark When using a subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Further, insert resistors in series on the side of XT2.

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

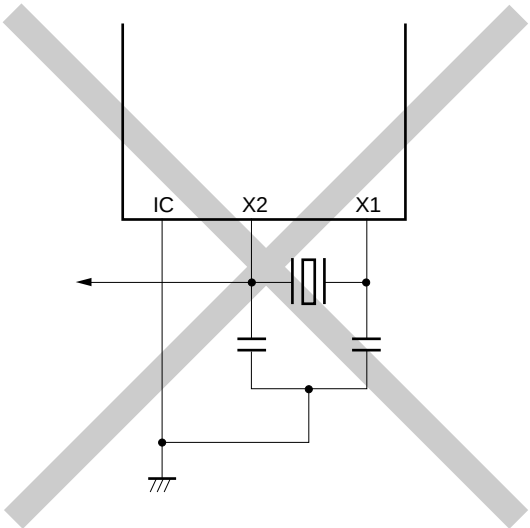
(c) Changing high current is too near a signal conductor



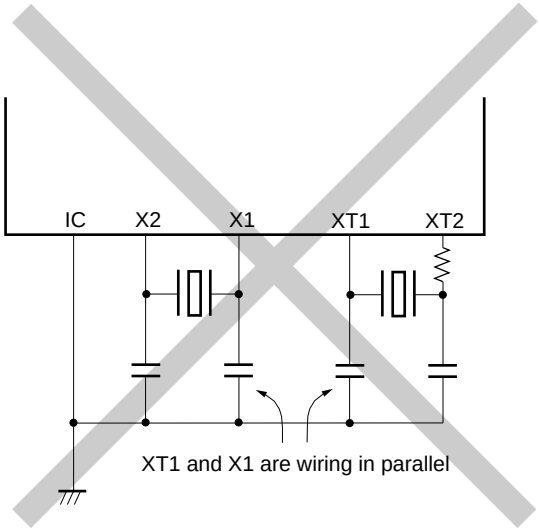
(d) Current flows through the grounding line of the oscillator (potential at points A, B, and C fluctuate)



(e) Signals are fetched



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



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

Cautions 2. In Figure 5-8 (f), XT1 and X1 are wired in parallel. Thus, the cross-talk noise of X1 may increase with XT1, resulting in malfunctioning. To prevent that from occurring, it is recommended to wire XT1 and X1 so that they are not in parallel.

5.4.3 Scaler

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

5.4.4 When no subsystem clocks are used

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

XT1: Connect to V_{DD}
XT2: Open

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

5.5 Clock Generator Operations

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

- Main system clock f_{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 at 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 reduced by specifying with the bit 6 of PCC (FRC) 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)

5.5.1 Main system clock operations

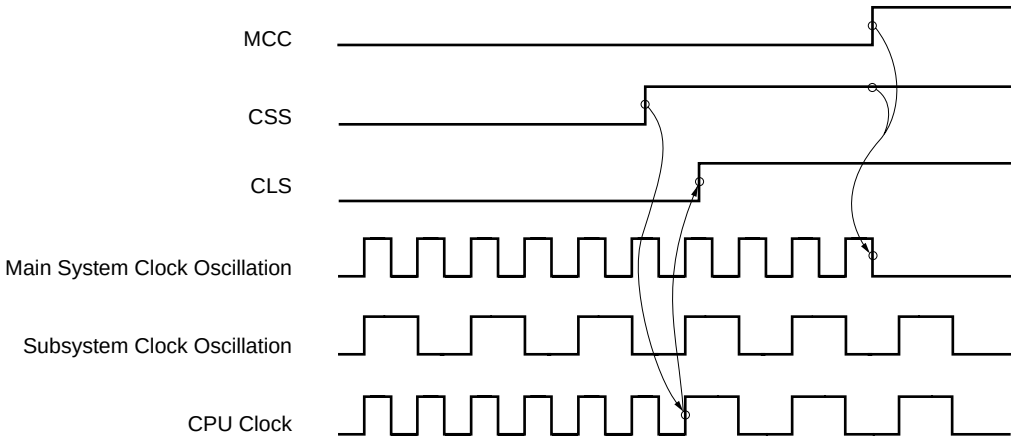
When operated with the main system clock (with bit 5 (CLS) of the processor clock control register (PCC) set to 0), the following operations are carried out by PCC setting.

(a) Because the operation guarantee instruction execution speed depends on the power supply voltage, the minimum instruction execution time can be changed by bit 0 to bit 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 (refer to **Figure 5-9**).

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

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



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

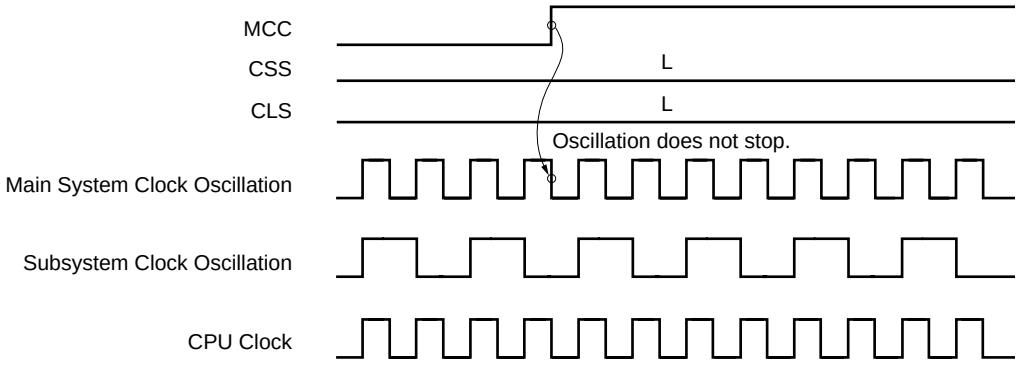
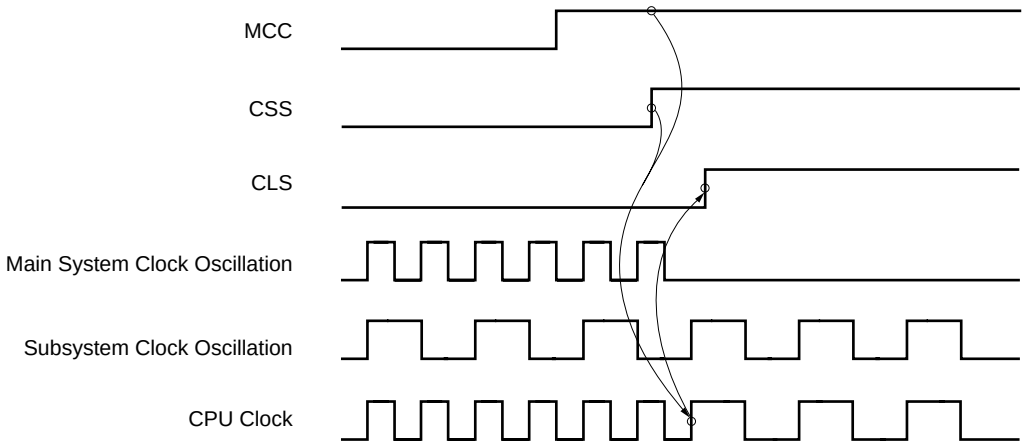


Figure 5-9. Main System Clock Stop Function (2/2)

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



5.5.2 Subsystem clock operations

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

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

(b) Watchdog timer counting stops.

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

5.6 Changing System Clock and CPU Clock Settings

5.6.1 Time required for switchover between system clock and CPU clock

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

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

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

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

Set Values after Switchover					Set Values before Switchover																							
MCS	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0
					0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	X	X	X
×	0	0	0	0					8 instructions				4 instructions				2 instructions				1 instruction				1 instruction			
		0	0	1	16 instructions								4 instructions				2 instructions				1 instruction				1 instruction			
		0	1	0	16 instructions				8 instructions								2 instructions				1 instruction				1 instruction			
		0	1	1	16 instructions				8 instructions				4 instructions								1 instruction				1 instruction			
		1	0	0	16 instructions				8 instructions				4 instructions				2 instructions								1 instruction			
1	1	×	×	×	$f_X/2f_{XT}$ instruction (77 instructions)				$f_X/4f_{XT}$ instruction (39 instructions)				$f_X/8f_{XT}$ instruction (20 instructions)				$f_X/16f_{XT}$ instruction (10 instructions)				$f_X/32f_{XT}$ instruction (5 instructions)							
0					$f_X/4f_{XT}$ instruction (39 instructions)				$f_X/8f_{XT}$ instruction (20 instructions)				$f_X/16f_{XT}$ instruction (10 instructions)				$f_X/32f_{XT}$ instruction (5 instructions)				$f_X/64f_{XT}$ instruction (3 instructions)							

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

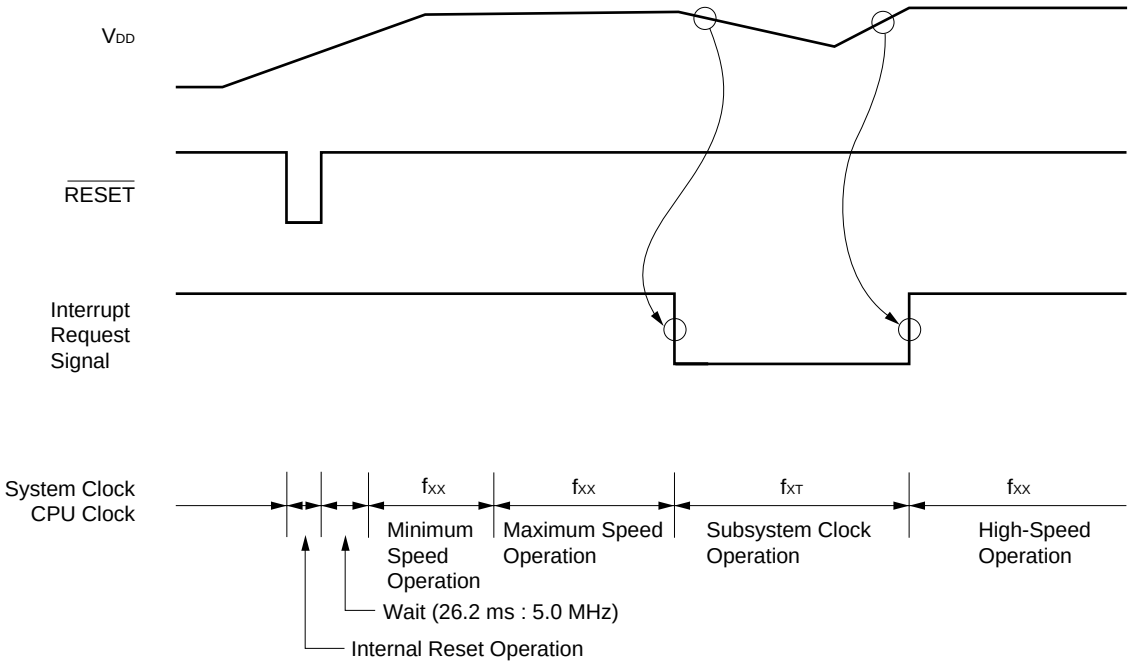
Remarks 1. One instruction is the minimum instruction execution time with the pre-switchover CPU clock.

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

5.6.2 System clock and CPU clock switching procedure

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

Figure 5-10. System Clock and CPU Clock Switching



(1) The CPU is reset by setting the $\overline{\text{RESET}}$ signal to low level after power-on. After that, when reset is released by setting the $\overline{\text{RESET}}$ signal to high level, 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, 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, 0 is set to the bit 7 of PCC (MCC) and oscillation of the main system clock is started. After the lapse of time required for stabilization of oscillation, the PCC and OSMS are rewritten and the maximum-speed operation is resumed.

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

[MEMO]

CHAPTER 6 16-BIT TIMER/EVENT COUNTER

6.1 Outline of μPD78064B Subseries Internal Timer

This chapter describe the 16-bit timer/event counter. First, the timers incorporated into the μPD78064B subseries 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 (Refer to **CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 1 AND 2**).

(3) Watch timer (TM3)

This timer can set a flag every 0.5 sec. and simultaneously generates interrupt requests at the preset time intervals (Refer to **CHAPTER 8 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 (Refer to **CHAPTER 9 WATCHDOG TIMER**).

(5) Clock output control circuit

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

(6) Buzzer output control circuit

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

Table 6-1. Timer/Event Counter Types and Functions

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

Notes 1. When capture/compare registers (CR00, CR01) are specified as compare registers.

2. Watch timer (TM3) can perform both watch timer and interval timer functions at the same time.

3. Watchdog timer (WDTM) can perform either the watchdog timer function or the interval timer function.

6.2 16-Bit Timer/Event Counter Functions

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

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

(1) Interval timer

TM0 generates interrupt requests at the preset time interval.

Table 6-2. 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	
—	$2 \times 1/f_X$ (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: Oscillation mode selection register (OSMS) bit 0

3. Values in parentheses when operated at 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 6-3. 16-Bit Timer/Event Counter Square-Wave Output Ranges

Minimum Pulse Width		Maximum Pulse Width		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
2 × 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: Oscillation mode selection register (OSMS) bit 0

3. Values in parentheses when operated at 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.

6.3 16-Bit Timer/Event Counter Configuration

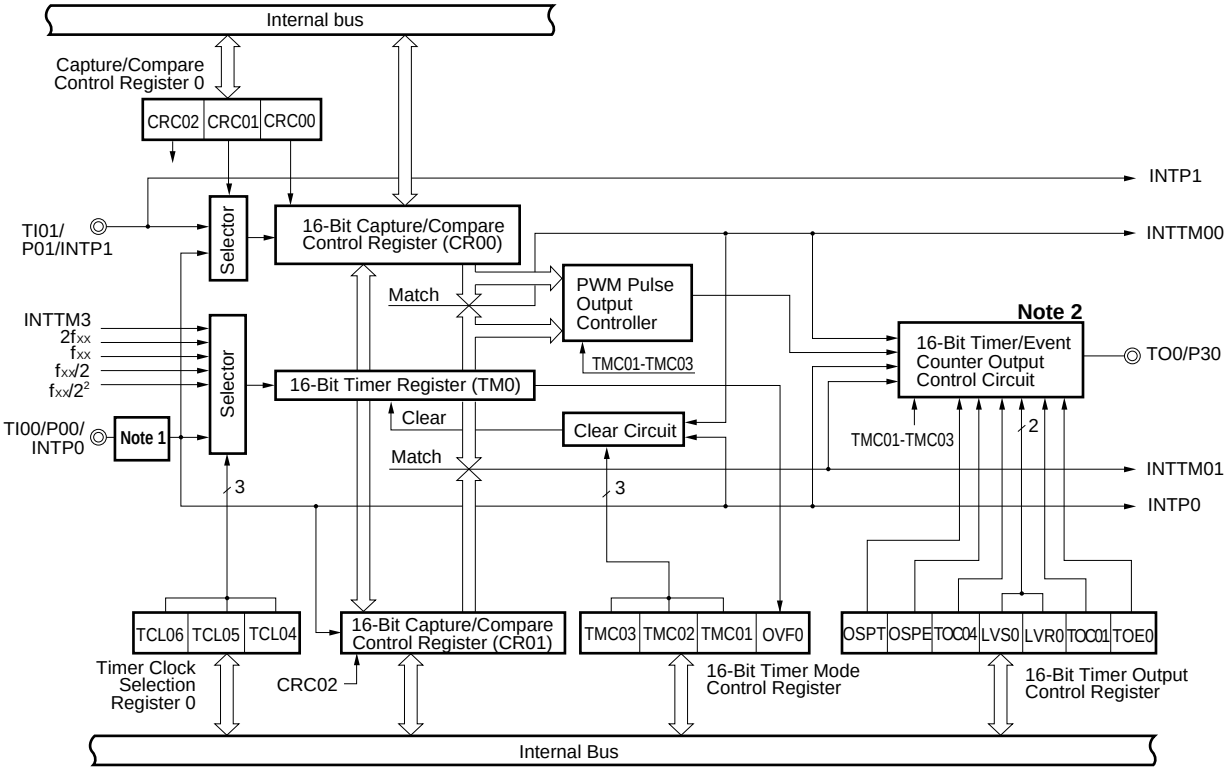
The 16-bit timer/event counter consists of the following hardware.

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

Item	Configuration
Timer register	16 bits × 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 6-1 16-Bit Timer/Event Counter Block Diagram**.

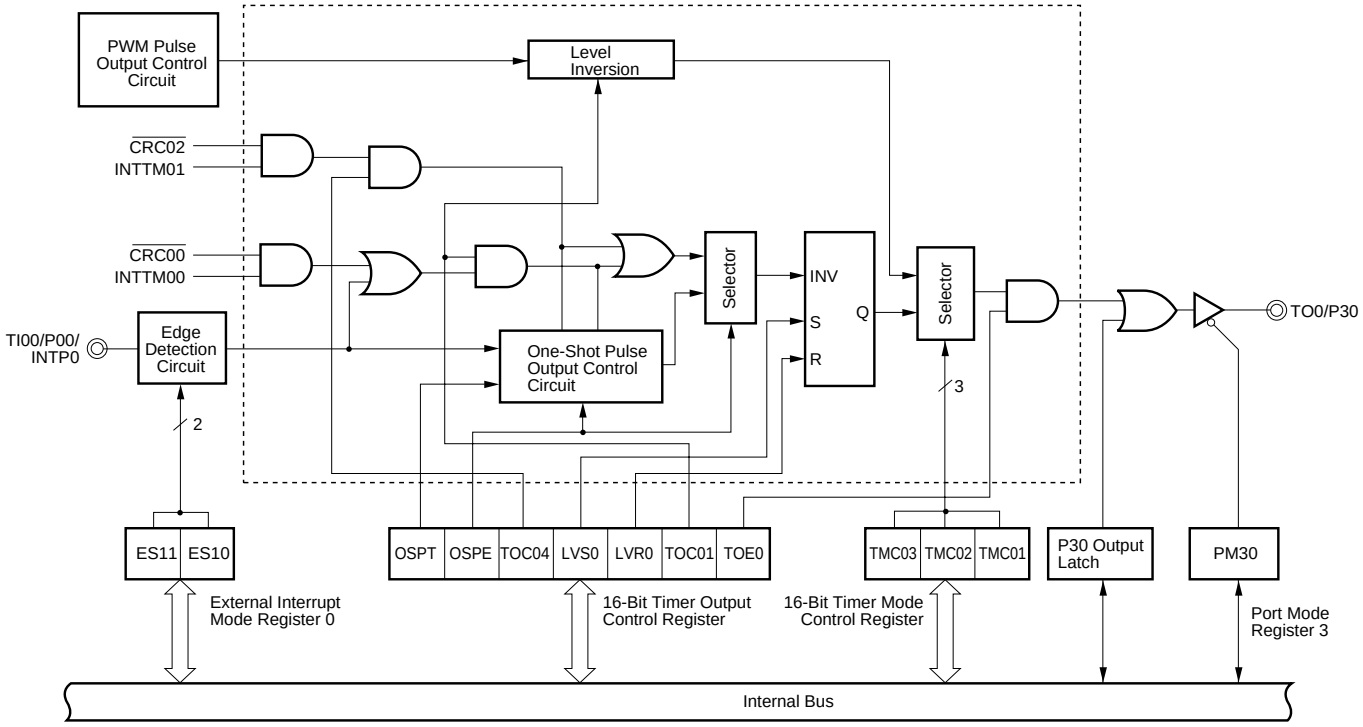
Figure 6-1. 16-Bit Timer/Event Counter Block Diagram



Notes 1. Edge detection circuit

2. The configuration of the 16-bit timer/event counter output control circuit is shown in Figure 6-2.

Figure 6-2. 16-Bit Timer/Event Counter Output Control Circuit Block Diagram



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

(1) Capture/compare register 00 (CR00)

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

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

When CR00 is used as a capture register, it is possible to select the valid edge of the INTP0/TI00 pin or the INTP1/TI01 pin as the capture trigger. Setting of the INTP0/TI00 or INTP1/TI01 valid edge is performed 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 6-5. INTP0/TI00 Pin Valid Edge and CR00 Capture Trigger Valid Edge

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

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

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

Cautions 1. Set a value other than 0000H to CR00. When the timer is used as an event counter, therefore, one pulse cannot be counted.

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

(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 (CRC0).

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

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

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

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

Caution If a valid edge of the TI0/P00 pin is input while CR01 is being read, CR01 does not perform the capture operation but holds the data. However, the interrupt request flag (PIF0), which is set on detection of the valid edge, 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 sets TM0 to 0000H.

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

6.4 16-Bit Timer/Event Counter Control Registers

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

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

(1) Timer clock select register 0 (TCL0)

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

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

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

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

Figure 6-3. Timer Clock Selection Register 0 Format

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

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. Setting of the TI00/INTP0 pin valid edge is performed by external interrupt mode register 0 (INTM0), and selection of the sampling clock frequency is performed by the sampling clock selection register (SCS).

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

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

4. When rewriting TCL0 to other data, stop the 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 of f_{XT} = 32.768 kHz.

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

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

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

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

Caution The 16-bit timer register starts operation 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 6-4. 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 Clear Mode Selection	TO0 Output Timing Selection	Interrupt Request Generation
0	0	0	Operation stop (TM0 cleared to 0)	No change	Not Generated
0	0	1	PWM mode (free running)	PWM pulse output	Generated on match between TM0 and CR00, 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 T00 output timing after stopping the timer operation (by setting TMC01 to TMC03 to 0, 0, 0).

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

3. When using the PWM mode, set the PWM 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.

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

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

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

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

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

Figure 6-5. 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, CR00 should not be specified as a capture register.

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

This register controls the operation of the 16-bit timer/event counter output control circuit. It sets R-S type flip-flop (LV0) setting/resetting, the active level in PWM mode, inversion enabling/disabling in modes other than PWM mode, 16-bit timer/event counter timer output enabling/disabling, one-shot pulse output operation enabling/disabling, and output trigger for a one-shop pulse by software. TOC0 is set with a 1-bit or 8-bit memory manipulation instruction. $\overline{\text{RESET}}$ input sets TOC0 value to 00H.

Figure 6-6. 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 (0)
1	0	Timer output F/F set (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 (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 value to FFH.

Figure 6-7. 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 sets INTM0 value to 00H.

Figure 6-8. 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 pin, clear the bits 1 through 3 (TMC01 through TMC03) of the 16-bit timer mode control register 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 sets SCS value to 00H.

Figure 6-9. 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. $f_{XX}/2^N$ is stopped in HALT mode.

Remarks 1. N : Value set in bit 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.

6.5 16-Bit Timer/Event Counter Operations

6.5.1 Interval timer operations

Setting the 16-bit timer mode control register (TMC0) and capture/compare control register 0 (CRC0) as shown in Figure 6-10 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.

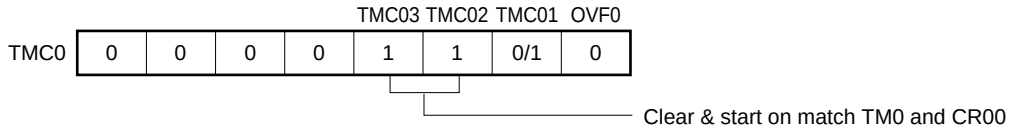
When the count value of the 16-bit timer register (TM0) matches the value set to CR00, counting continues with the TM0 value cleared to 0 and the interrupt request signal (INTTM00) is generated.

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

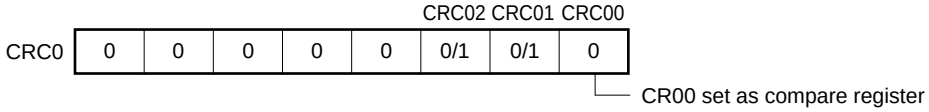
For the operation when the value of the compare register is changed during timer count operation, refer to **6.6 16-Bit Timer/Event Counter Operating Precautions**.

Figure 6-10. Control Register Settings for Interval Timer Operation

(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 the interval timer. Refer to the description of the respective control registers for details.

Figure 6-11. Interval Timer Configuration Diagram

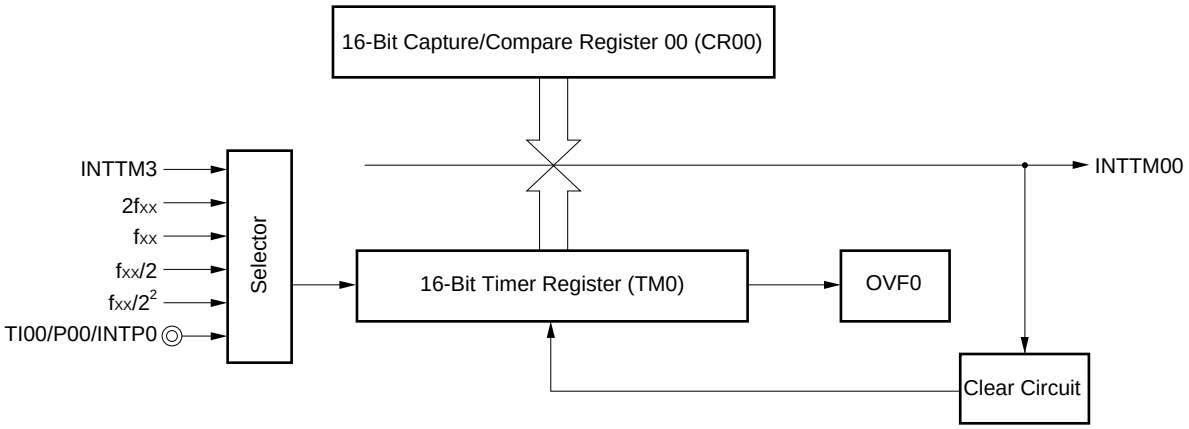
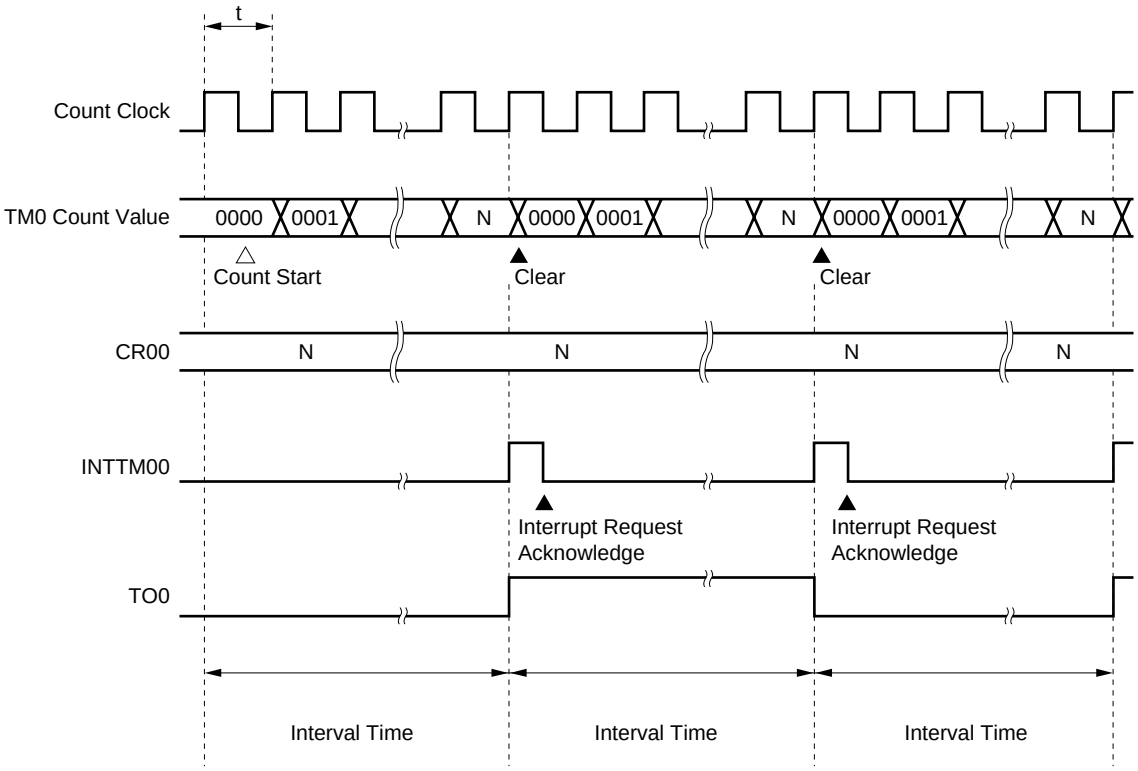


Figure 6-12. Interval Timer Operation Timings



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

Table 6-6. 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. Figures in parentheses apply to operation with fx = 5.0 MHz

6.5.2 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 6-13 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 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 bit 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.

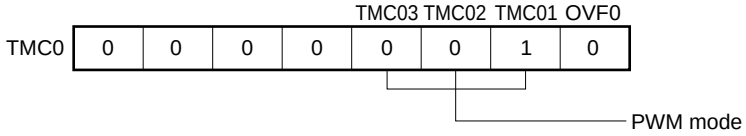
Cautions 1. PWM operation mode should be selected before setting CR00.

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

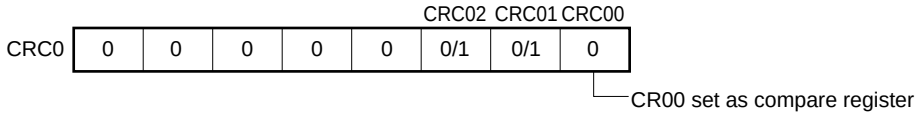
3. Do not select PWM operation mode for external clock input from the TI00/P00 pin.

Figure 6-13. Control Register Settings for PWM Output Operation

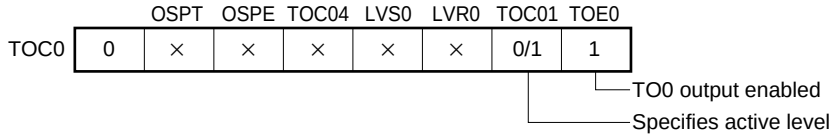
(a) 16-bit timer mode control register (TMC0)



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



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



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

2. × : 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 6-14 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 6-14. Example of D/A Converter Configuration with PWM Output

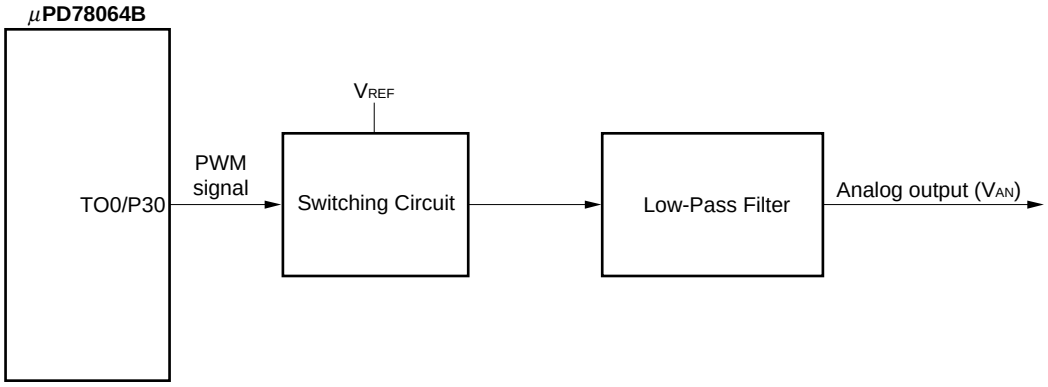
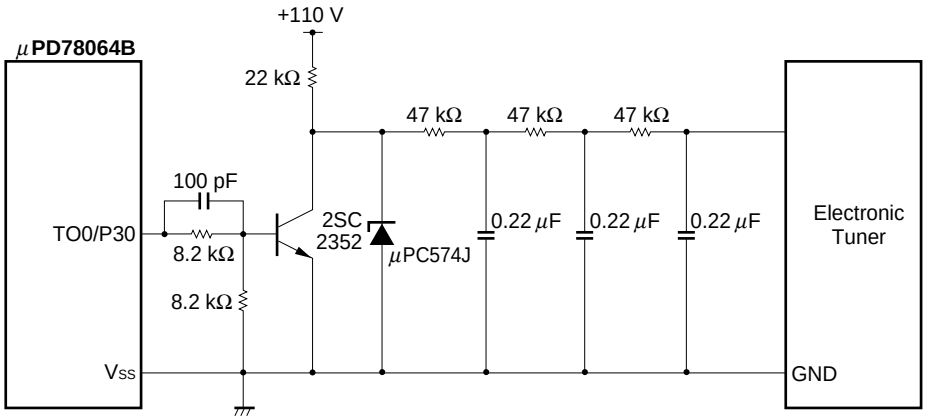


Figure 6-15 shows an example in which PWM output is converted to an analog voltage and used in a voltage synthesizer type TV tuner.

Figure 6-15. TV Tuner Application Circuit Example



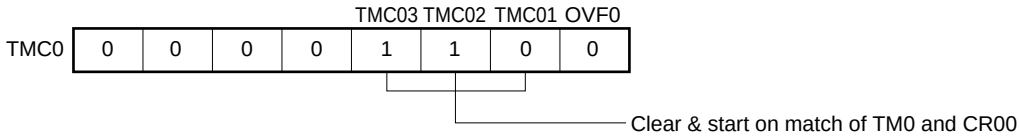
6.5.3 PPG output operations

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

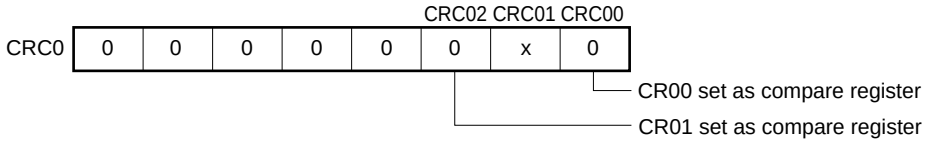
In the PPG output operation, square waves are output from the TO0/P30 pin with the pulse width and the cycle that correspond to the count values set beforehand in 16-bit capture/compare register 01 (CR01) and in 16-bit capture/compare register 00 (CR00), respectively.

Figure 6-16. Control Register Settings for PPG Output Operation

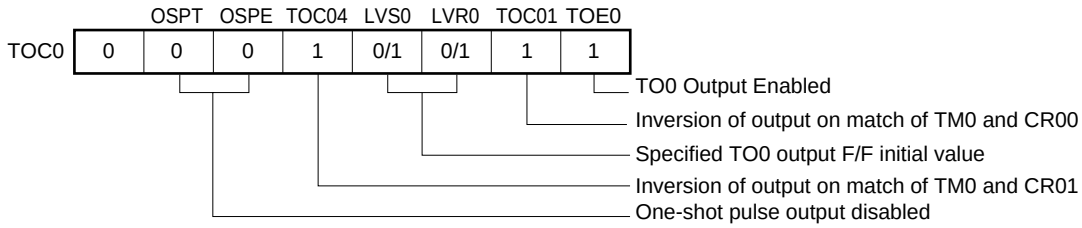
(a) 16-bit timer mode control register (TMC0)



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



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



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

Remark ×: Don't care

6.5.4 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

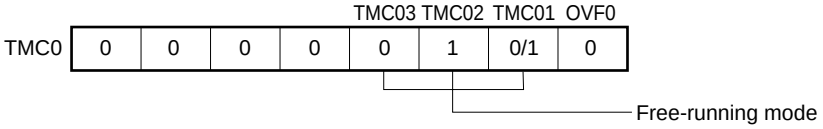
When the 16-bit timer register (TM0) is operated in free-running mode (refer to register settings in Figure 6-17), and the edge specified by external interrupt mode register 0 (INTM0) is input to the TI00/P00 pin, the value of TM0 is taken into 16-bit capture/compare register 01 (CR01) and an external 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.

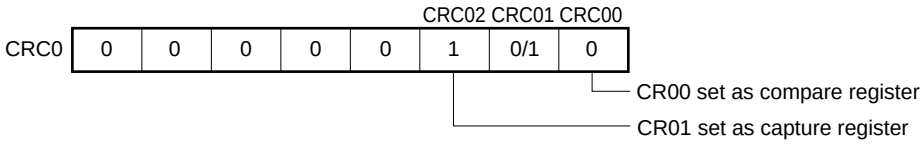
For valid edge detection, sampling is performed at the interval selected by means of the sampling clock selection register (SCS), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Figure 6-17. 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. Refer to the description of the respective control registers for details.

Figure 6-18. Configuration Diagram for Pulse Width Measurement by Free-Running Counter

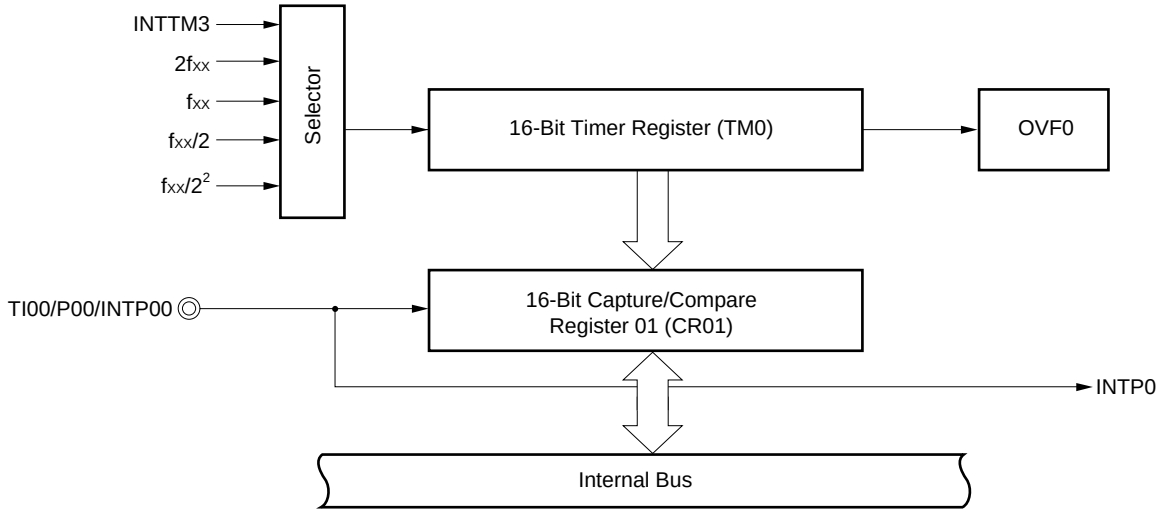
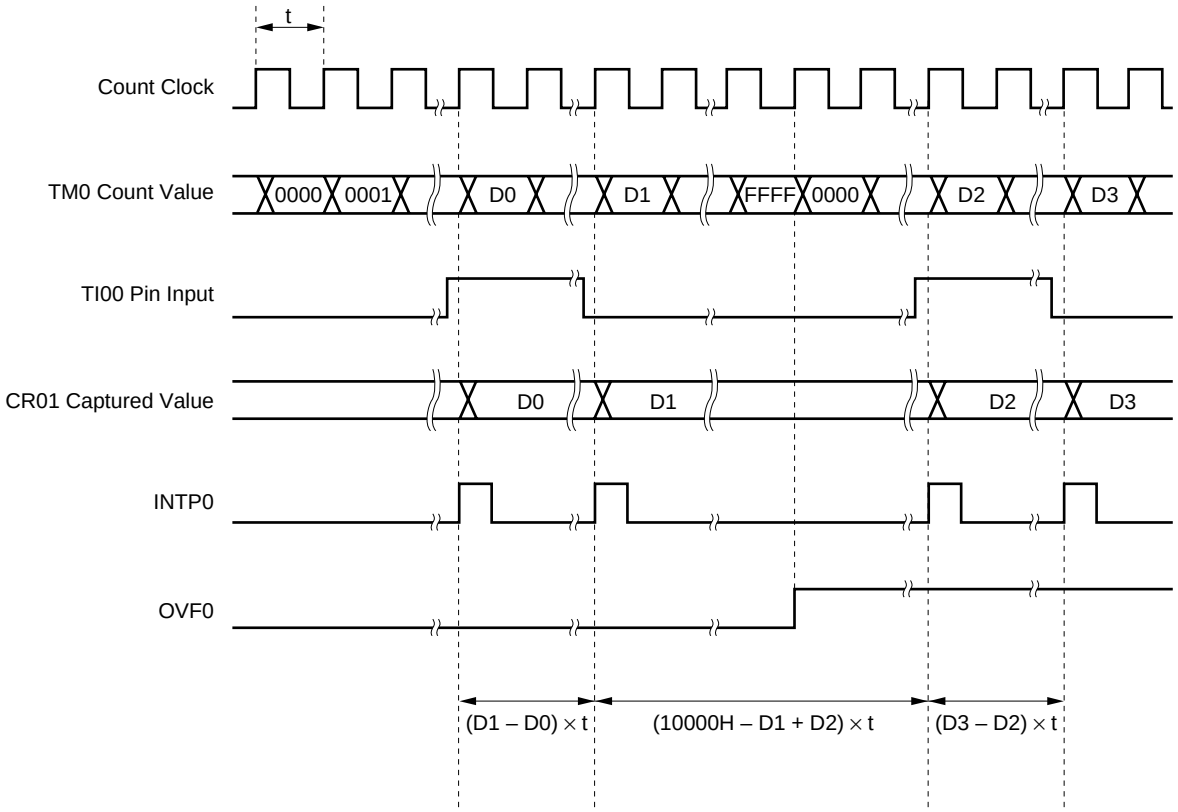


Figure 6-19. Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)



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

When the 16-bit timer register (TM0) is operated in free-running mode (refer to register settings in Figure 6-20), 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.

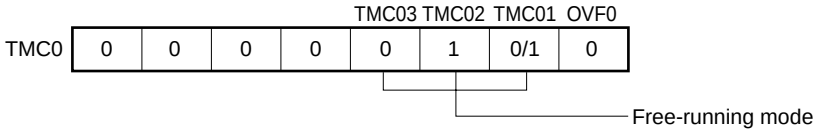
Also, when the edge specified by bits 4 and 5 (ES20 and ES21) of INTM0 is input to the TI01/P01 pin, the value of TM0 is taken into 16-bit capture/compare register 00 (CR00) and an external interrupt request signal (INTP1) is set.

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.

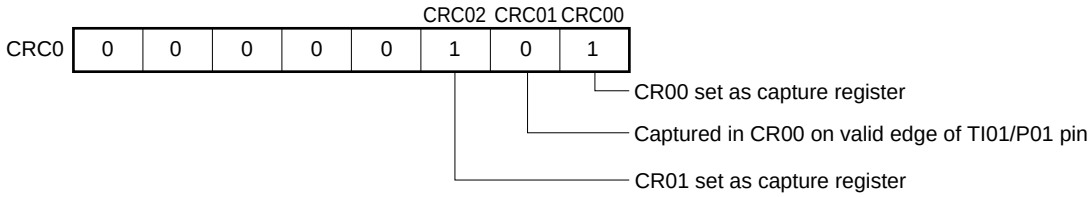
For TI00/P00 pin valid edge detection, sampling is performed at the interval selected by means of the sampling clock selection register (SCS), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Figure 6-20. Control Register Settings for Two Pulse Width Measurements with Free-Running Counter

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

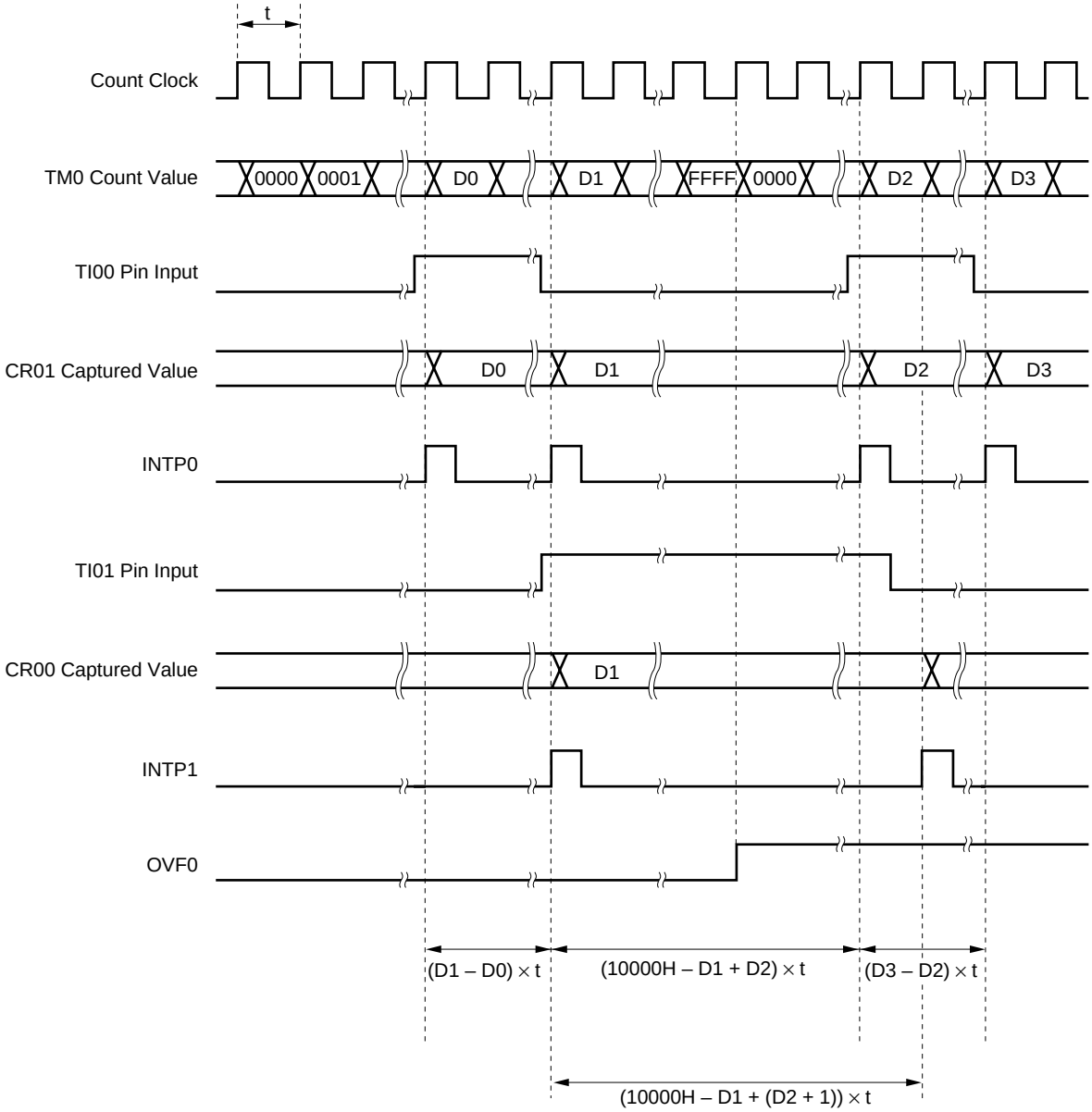


(b) Capture/compare control register 0 (CRC0)



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

Figure 6-21. 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 (refer to register settings in Figure 6-22), 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).

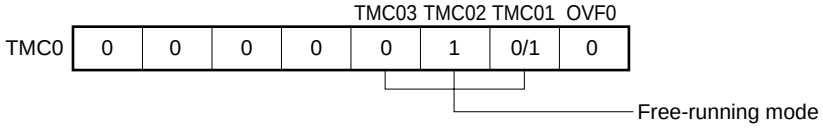
Either of two edge specifications can be selected—rising or falling—as the valid edges for the TI00/P00 pin by means of bits 2 and 3 (ES10 and ES11) of INTM0.

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.

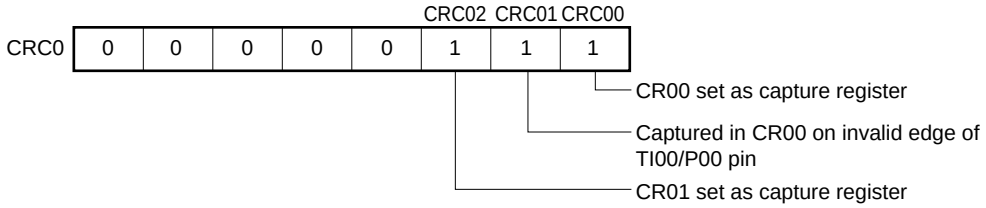
Caution If the valid edge of TI00/P00 is specified to be both rising and falling edge, capture/compare register 00 (CR00) cannot perform the capture operation.

Figure 6-22. Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers

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

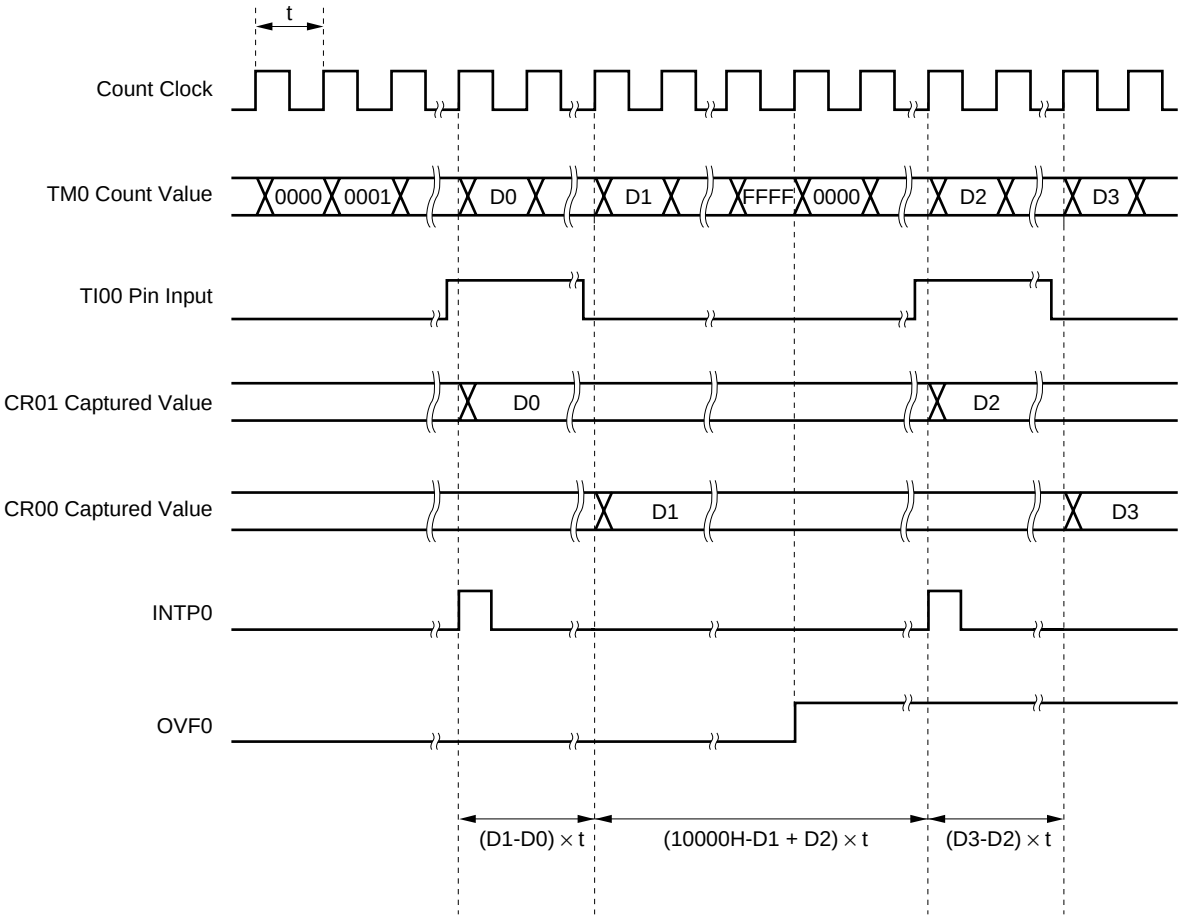


(b) Capture/compare control register 0 (CRC0)



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

Figure 6-23. 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

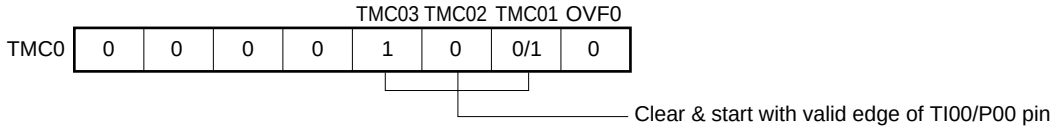
When input of a valid edge to the TI00/P00 pin is detected, the count value of the 16-bit timer register (TM0) is taken into 16-bit capture/compare register 01 (CR01), and then the pulse width of the signal input to the TI00/P00 pin is measured by clearing TM0 and restarting the count (refer to register settings in Figure 6-24). The edge specification can be selected from two types, rising and falling edges by the 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.

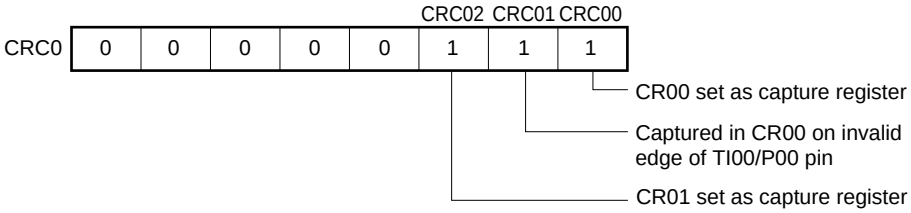
Caution If the valid edge of TI00/P00 is specified to be both rising and falling edge, the 16-bit capture/compare register 00 (CR00) cannot perform the capture operation.

Figure 6-24. Control Register Settings for Pulse Width Measurement by Means of Restart

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

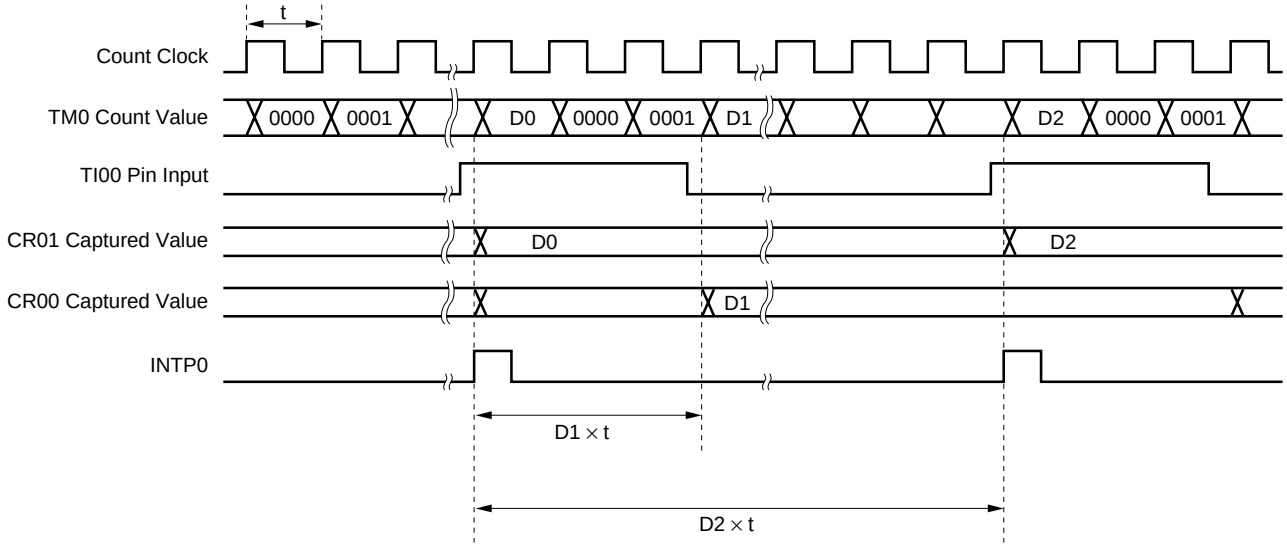


(b) Capture/compare control register 0 (CRC0)



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

Figure 6-25. Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)



6.5.5 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.

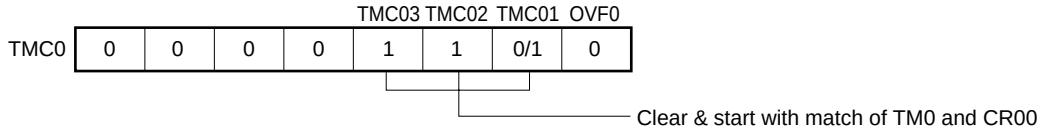
Set a value other than 0000H to the 16-bit capture/compare register 00 (CR00). (1-pulse counting cannot be operated.)

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

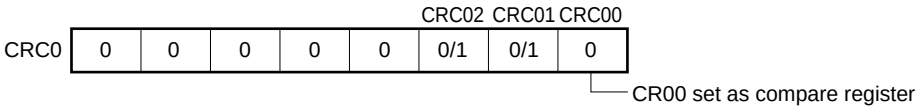
Because operation is carried out only after the valid edge is detected twice by sampling at the interval selected with the sampling clock select register (SCS), noise with short pulse widths can be removed.

Figure 6-26. Control Register Settings in External Event Counter Mode

(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 the external event counter. Refer to the description of the respective control registers for details.

Figure 6-27. External Event Counter Configuration Diagram

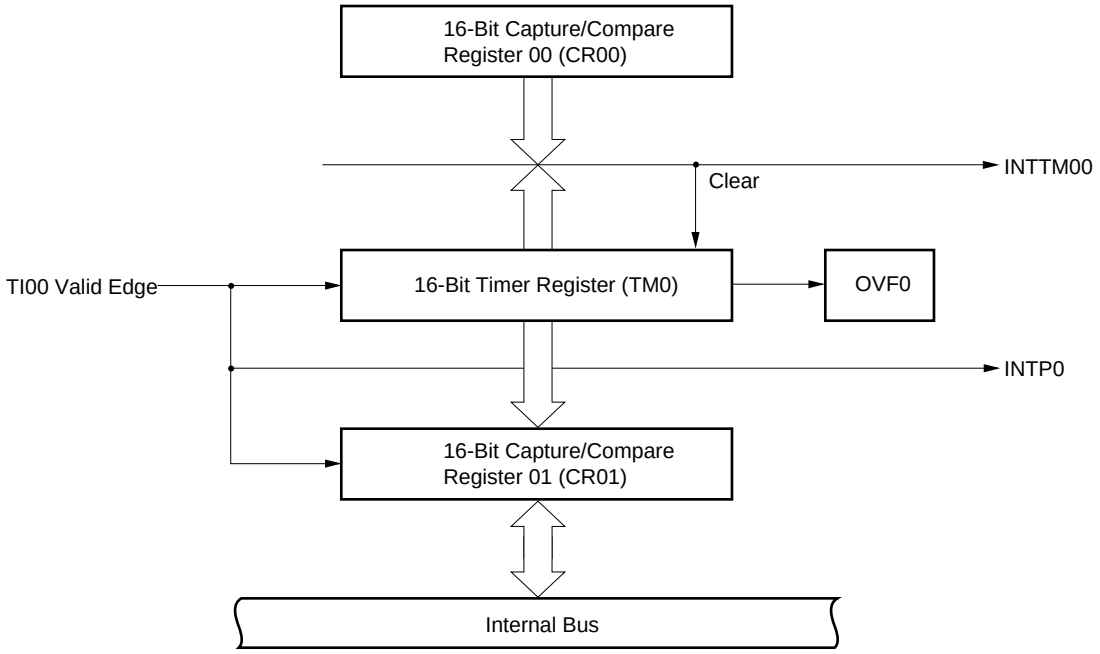
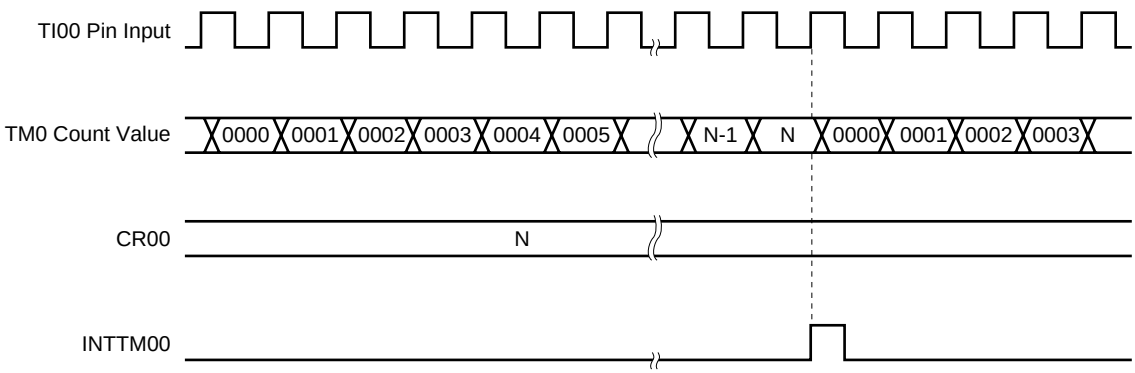


Figure 6-28. External Event Counter Operation Timings (with Rising Edge Specified)



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

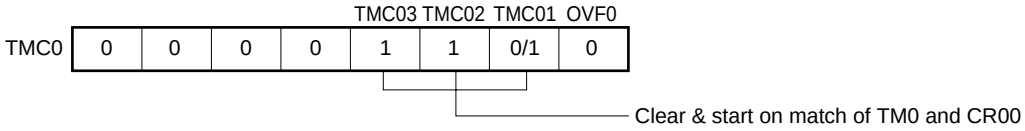
6.5.6 Square-wave output operation

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

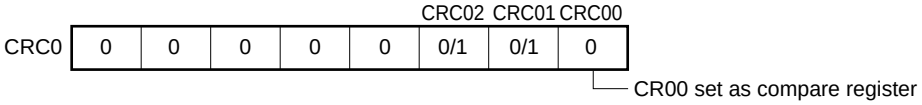
The TO0/P30 pin output status is reversed at intervals of the count value preset to CR00 by setting bit 0 (TOE0) and bit 1 (TOC01) of the 16-bit timer output control register to 1. This enables a square wave with any selected frequency to be output.

Figure 6-29. Control Register Settings in Square-Wave Output Mode

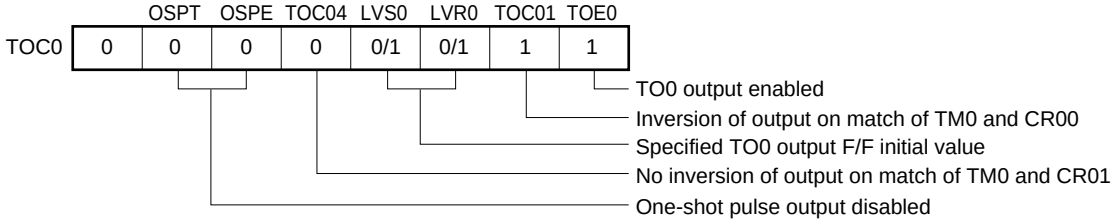
(a) 16-bit timer mode control register (TMC0)



(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 square-wave output. Refer to the description of the respective control registers for details.

Figure 6-30. Square-Wave Output Operation Timing

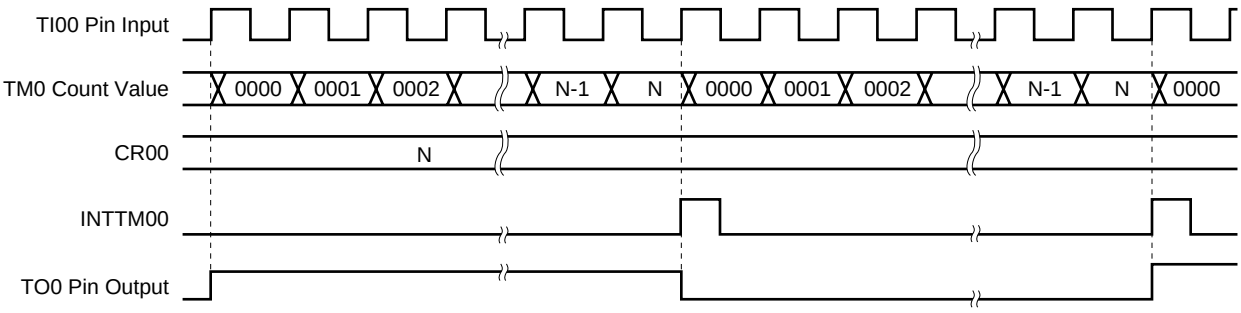


Table 6-7. 16-Bit Timer/Event Count Square-Wave Output Ranges

Minimum Pulse Width		Maximum Pulse Width		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
$2 \times$ TI00 input cycle		$2^{16} \times$ 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)
$2 \times$ watch timer output cycle		$2^{16} \times$ watch timer output cycle		Watch timer output edge cycle	

Remarks 1. f_X : Main system clock oscillation frequency

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

3. Values in parentheses when operated at f_X = 5.0 MHz

6.5.7 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 6-31, and 1 is set in bit 6 (OSPT) of TOC0 by software, a one-shot pulse is output from the TO0/P30 pin.

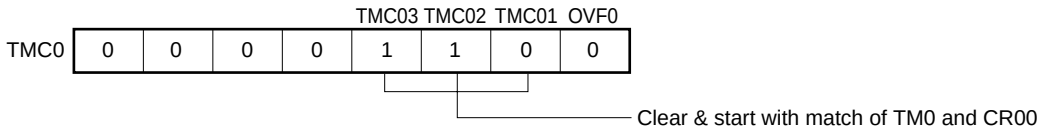
By setting 1 in OSPT, the 16-bit timer/event counter is cleared and started, and output is activated by the count value set beforehand in 16-bit capture/compare register 01 (CR01). Thereafter, output is inactivated by the count value set beforehand in 16-bit capture/compare register 00 (CR00).

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

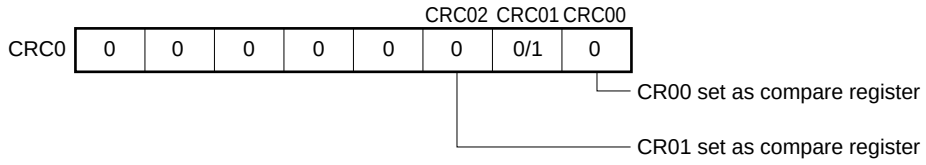
Caution When outputting one-shot pulse, do not set 1 in OSPT. When outputting one-shot pulse again, execute after the INTTM00, or interrupt match signal with CR00, is generated.

Figure 6-31. Control Register Settings for One-Shot Pulse Output Operation Using Software Trigger

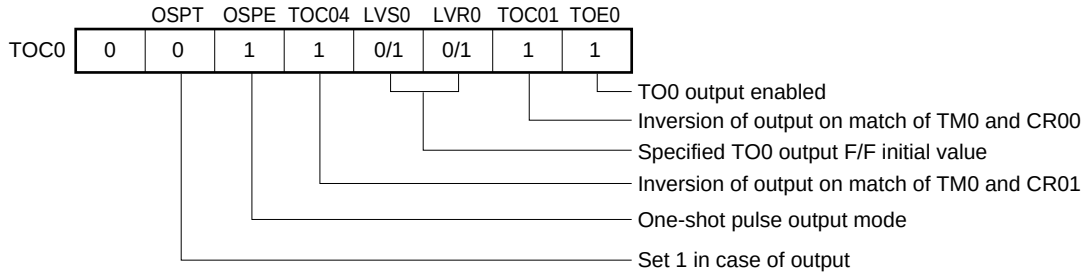
(a) 16-bit timer mode control register (TMC0)



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



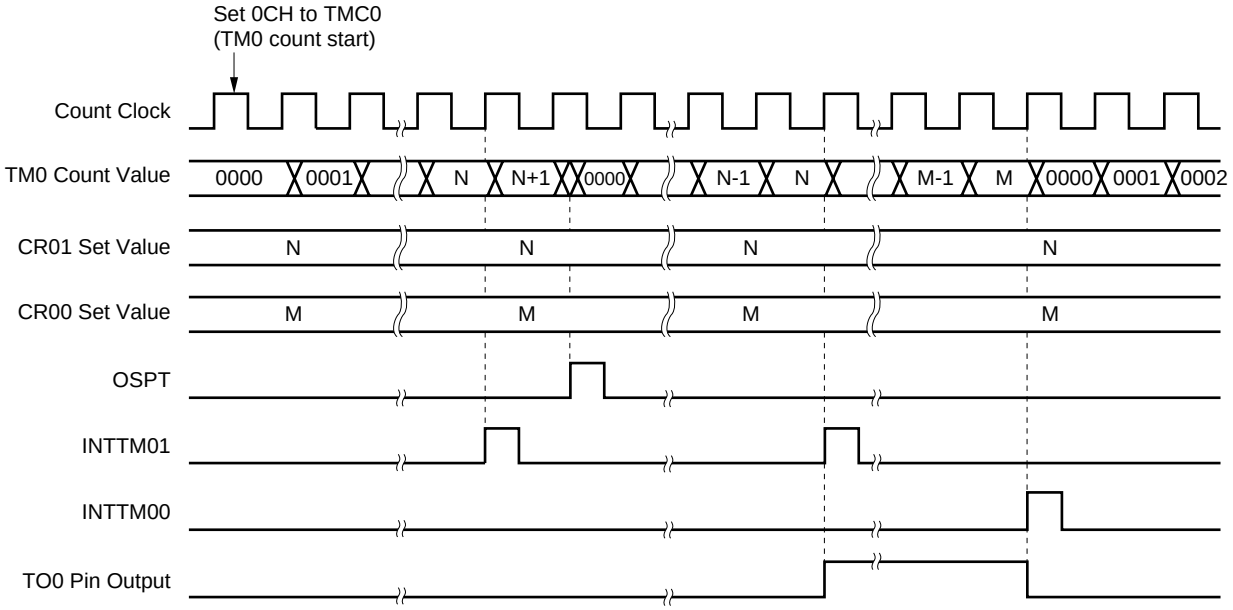
(c) 16-bit timer output control register (TOC0)



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

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

Figure 6-32. Timing of One-Shot Pulse Output Operation Using Software Trigger



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

(2) One-shot pulse output using external trigger

If the 16-bit timer mode control register (TMC0), capture/compare control register 0 (CRC0), and the 16-bit timer output control register (TOC0) are set as shown in Figure 6-33, a one-shot pulse is output from the TO0/P30 pin with a TI00/P00 valid edge as an external trigger.

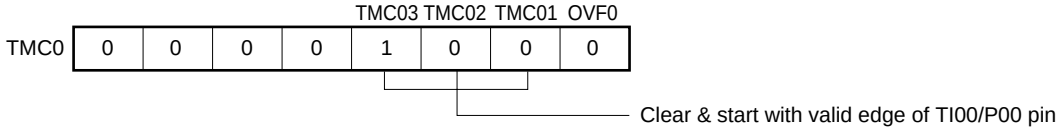
Any of three edge specifications can be selected—rising, falling, or both edges — as the valid edges for the TI00/P00 pin by means of bits 2 and 3 (ES10 and ES11) of external interrupt mode register 0 (INTM0).

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).

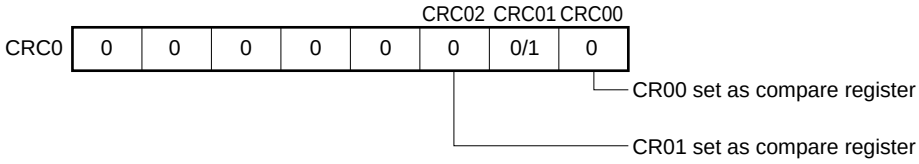
Caution When outputting one-shot pulses, external trigger is ignored if generated again.

Figure 6-33. Control Register Settings for One-Shot Pulse Output Operation Using External Trigger

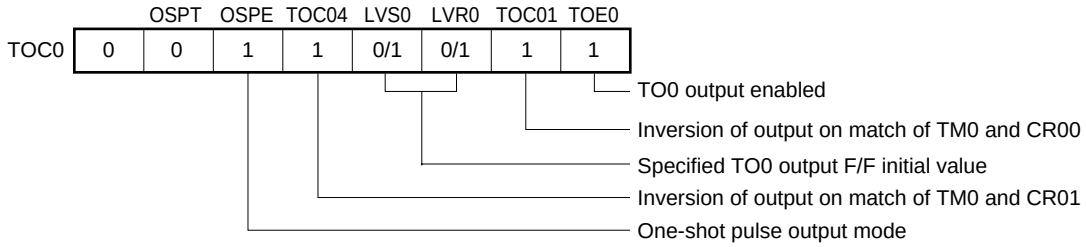
(a) 16-bit timer mode control register (TMC0)



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



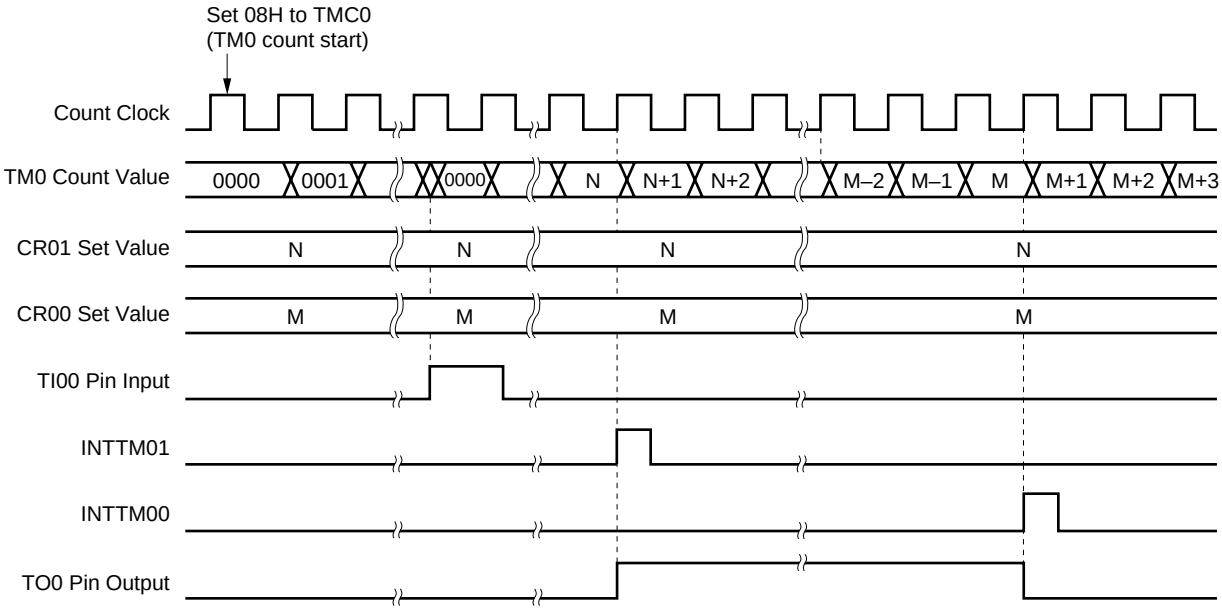
(c) 16-bit timer output control register (TOC0)



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

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

Figure 6-34. Timing of One-Shot Pulse Output Operation Using External Trigger (With Rising Edge Specified)



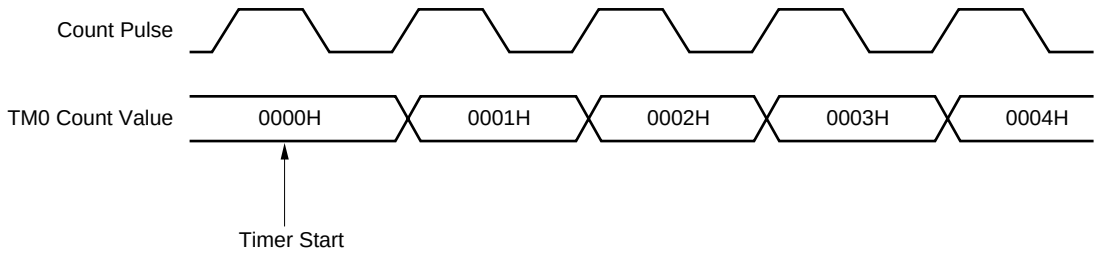
Caution The 16-bit timer register starts operation at the moment a value other than 0, 0, 0 (operation stop mode) is set to TMC01 to TMC03, respectively.

6.6 16-Bit Timer/Event Counter Operating Precautions

(1) Timer start errors

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

Figure 6-35. 16-Bit Timer Register Start Timing



(2) 16-bit compare register setting

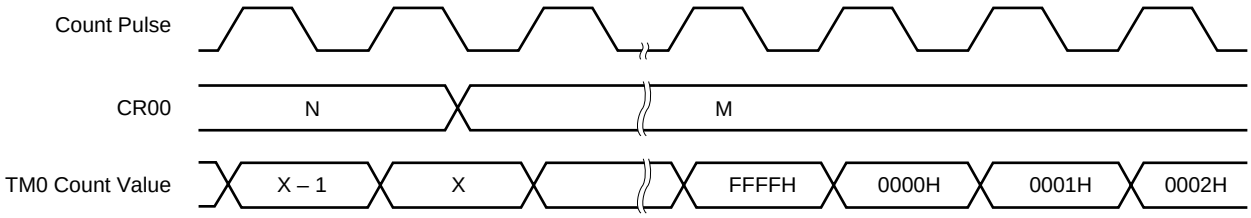
Set a value other than 0000H to the 16-bit capture/compare register 00 (CR00).

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 6-36. Timings After Change of Compare Register During Timer Count Operation

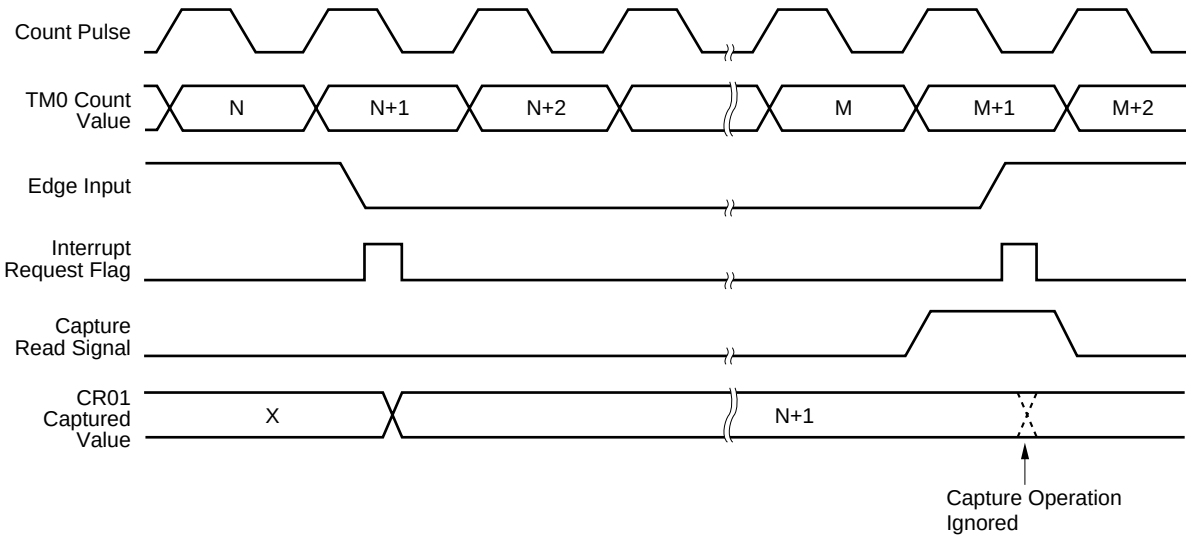


Remark N > X > M

(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 6-37. Capture Register Data Retention Timing



(5) Valid edge setting

Set the valid edge of the TI00/P00/INTP0 pin after setting bits 1 to 3 (TMC01 to TMC03) of the 16-bit timer mode control register (TMC0) to 0, 0 and 0, respectively, and then stopping timer operation. Valid edge setting is carried out with bits 2 and 3 (ES10 and ES11) of the external interrupt mode register 0 (INTM0).

(6) Re-trigger of one-shot pulse

(a) One-shot pulse output using software

When outputting one-shot pulse, do not set 1 in OSPT. When outputting one-shot pulse again, execute it after the INTTM00, or interrupt request 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.

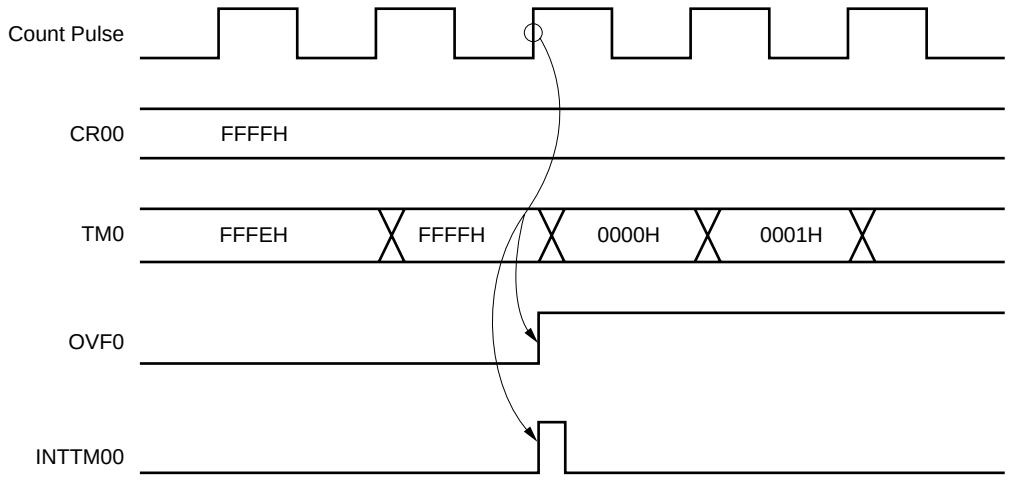
↓

CR00 is set to FFFFH.

↓

When TM0 is counted up from FFFFH to 0000H.

Figure 6-38. Operation Timing of OVF0 Flag



[MEMO]

CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 1 AND 2

7.1 8-Bit Timer/Event Counters 1 and 2 Functions

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

7.1.1 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 7-1. 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 : Oscillation mode selection register (OSMS) bit 0

3. Values in parentheses when operated at 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 7-2. 8-Bit Timer/Event Counters 1 and 2 Square-Wave Output Ranges

Minimum Pulse Width		Maximum Pulse Width		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
$2 \times 1/f_X$ (400 ns)	$2^2 \times 1/f_X$ (800 ns)	$2^9 \times 1/f_X$ (102.4 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 : Oscillation mode selection register (OSMS) bit 0

3. Values in parentheses when operated at f_X = 5.0 MHz.

7.1.2 16-bit timer/event counter mode

(1) 16-bit interval timer

Interrupt requests can be generated at the preset time intervals.

Table 7-3. Interval 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 : Oscillation mode selection register (OSMS) bit 0

3. Values in parentheses when operated at 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 7-4. Square-Wave Output Ranges when 8-Bit Timer/Event Counters 1 and 2 are Used as 16-Bit Timer/Event Counters

Minimum Pulse Width		Maximum Pulse Width		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
$2 \times 1/f_X$ (400 ns)	$2^2 \times 1/f_X$ (800 ns)	$2^{17} \times 1/f_X$ (26.2 ms)	$2^{18} \times 1/f_X$ (52.4 ms)	$2 \times 1/f_X$ (400 ns)	$2^2 \times 1/f_X$ (800 ns)
$2^2 \times 1/f_X$ (800 ns)	$2^3 \times 1/f_X$ (1.6 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 : Oscillation mode selection register (OSMS) bit 0

3. Values in parentheses when operated at f_X = 5.0 MHz.

7.2 8-Bit Timer/Event Counters 1 and 2 Configurations

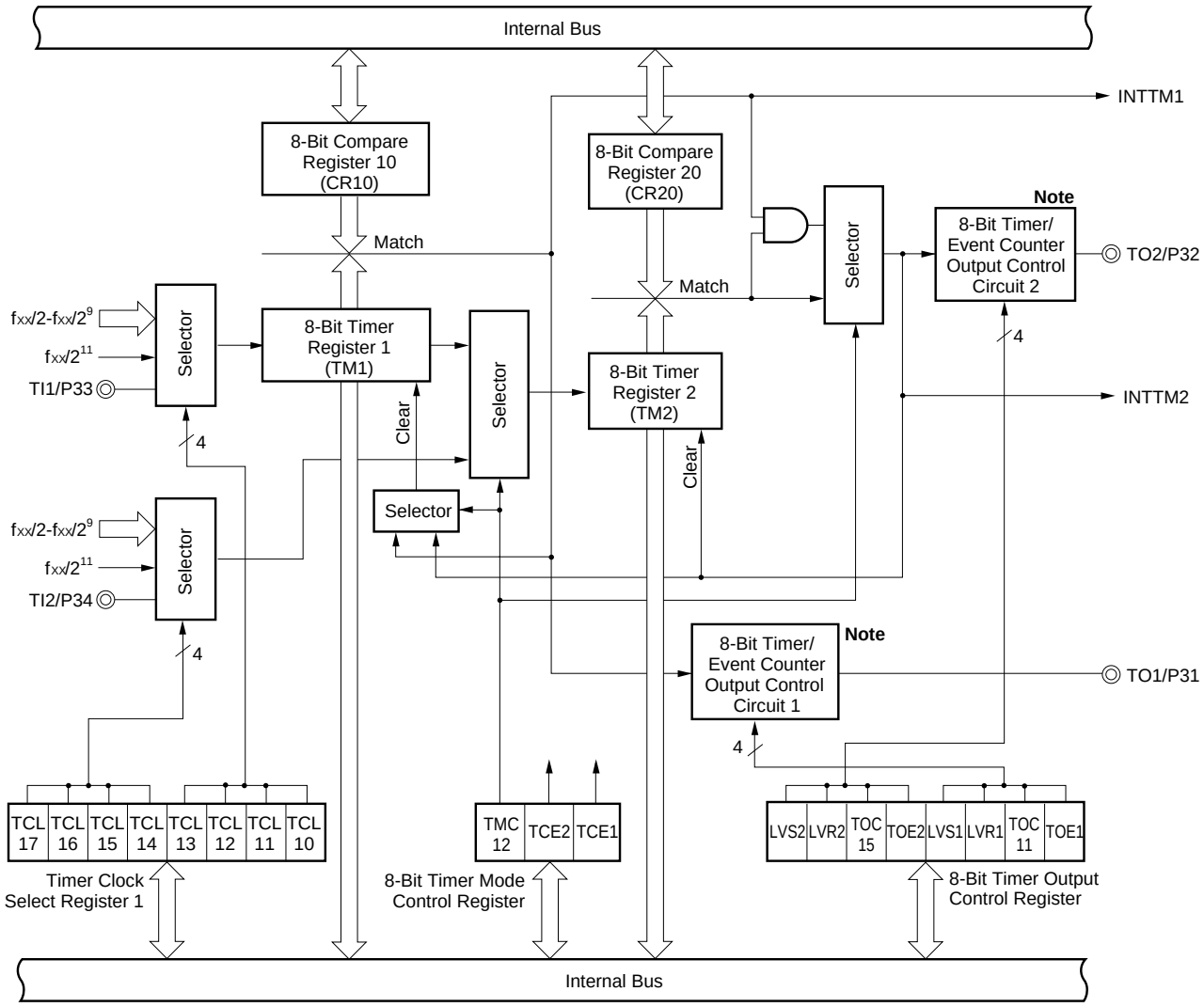
The 8-bit timer/event counters 1 and 2 consist of the following hardware.

Table 7-5. 8-Bit Timer/Event Counters 1 and 2 Configurations

Item	Configuration
Timer register	8 bits × 2 (TM1, TM2)
Register	Compare register: 8 bits × 2 (CR10, CR20)
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**

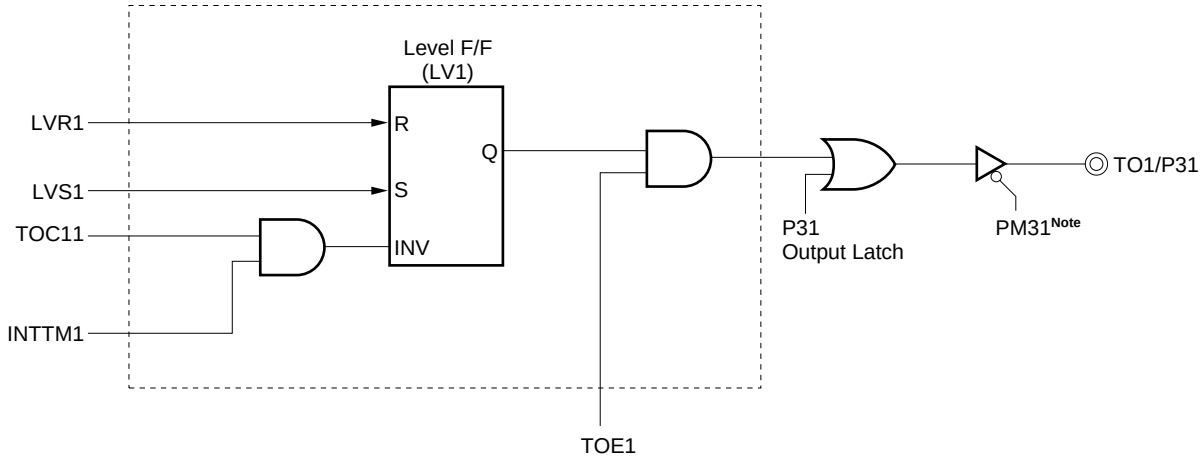
Note Refer to **Figure 4-7 Block Diagram of P30 to P37**

Figure 7-1. 8-Bit Timer/Event Counters 1 and 2 Block Diagram



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

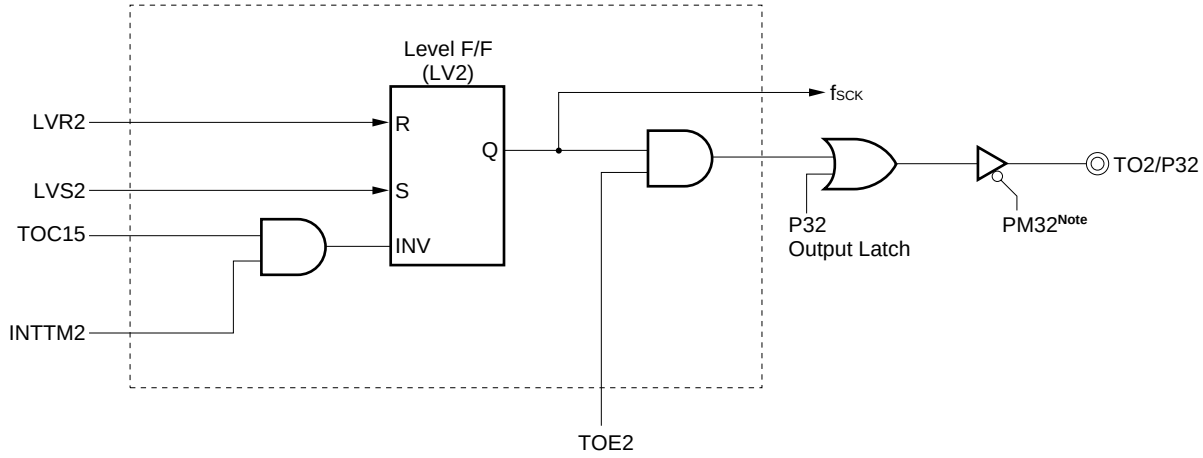
Figure 7-2. Block Diagram of 8-Bit Timer/Event Counter Output Control Circuit 1



Note Bit 1 of the port mode register 3 (PM3)

Remark The section in the broken line is an output control circuit.

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



Note Bit 2 of the 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. If the new values of CR10 and CR20 are less than the values of the 8-bit timer registers (TM1 and TM2), TM1 and TM2 continue counting, overflow, and restart counting from 0. If the new values of CR10 and CR20 are less than the old values, the timers must be restarted after changing 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 (TMS) is read with a 16-bit memory manipulation instruction.

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

7.3 8-Bit Timer/Event Counters 1 and 2 Control Registers

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

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

(1) Timer clock select register 1 (TCL1)

This register sets count clocks of 8-bit timer registers 1 and 2.

TCL1 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TCL1 to 00H.

Figure 7-4. 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 : Oscillation mode selection register (OSMS) bit 0
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 2.

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

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

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

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

TCE1	8-Bit Timer Register 1 Operation Control
0	Operation stop (TM1 clear to 0)
1	Operation enable

TCE2	8-Bit Timer Register 2 Operation Control
0	Operation stop (TM2 clear to 0)
1	Operation enable

TMC12	Operating Mode Selection
0	8-Bit timer register × 2 channel mode (TM1, TM2)
1	16-Bit timer register × 1 channel mode (TMS)

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

2. When used as 16-bit timer register, TCE1 should be used for operation enable/stop.

(3) 8-bit timer output control register (TOC1)

This register controls operation of 8-bit timer/event counter output control circuits 1 and 2.

It sets/resets the R-S flip-flops (LV1 and LV2) and enables/disables inversion and 8-bit timer output of 8-bit timer registers 1 and 2.

TOC1 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TOC1 to 00H.

Figure 7-6. 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 reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TOE2	8-Bit Timer/Event Counter 2 Output Control
0	Output 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 reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

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

2. After data setting, 0 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 7-7. 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)

7.4 8-Bit Timer/Event Counters 1 and 2 Operations

7.4.1 8-bit timer/event counter mode

(1) Interval timer operations

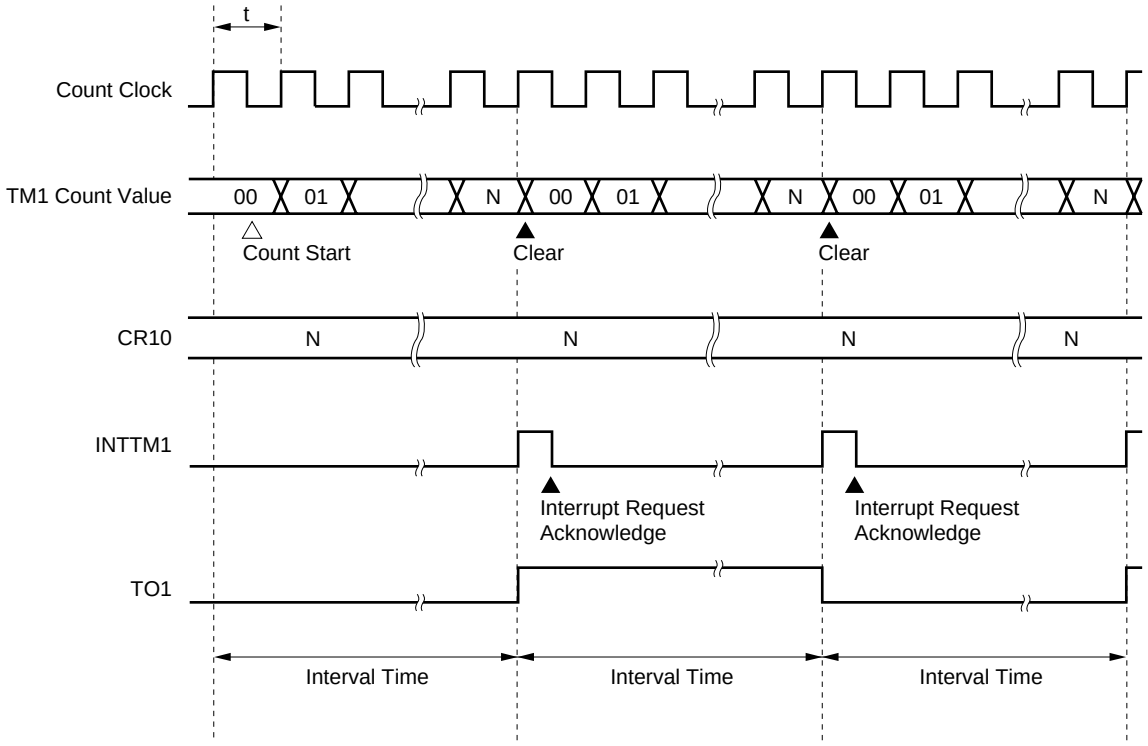
The 8-bit timer/event counters 1 and 2 operate as interval timers which generate interrupt requests repeatedly at intervals of the count value preset to 8-bit compare registers 10 and 20 (CR10 and CR20).

When the count values of the 8-bit timer registers 1 and 2 (TM1 and TM2) match the values set to CR10 and CR20, counting continues with the TM1 and TM2 values cleared to 0 and the interrupt request signals (INTTM1 and INTTM2) are generated.

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

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

Figure 7-8. Interval Timer Operation Timings



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

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

TCL13	TCL12	TCL11	TCL10	Minimum Interval Time		Maximum Interval Time		Resolution	
				MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
0	0	0	0	TI1 input cycle		$2^8 \times$ TI1 input cycle		TI1 input edge cycle	
0	0	0	1	TI1 input cycle		$2^8 \times$ TI1 input cycle		TI1 input edge cycle	
0	1	1	0	$2 \times 1/f_X$ (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 : Oscillation mode selection register (OSMS) bit 0

3. TCL10-TCL13: Bits 1 through 3 of the timer clock select register (TCL1)

4. Values in parentheses when operated at f_X = 5.0 MHz.

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

TCL17	TCL16	TCL15	TCL14	Minimum Interval Time		Maximum Interval Time		Resolution	
				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 : Oscillation mode selection register (OSMS) bit 0

3. TCL14-TCL17: Bits 4 through 7 of the timer clock select register (TCL1)

4. Values in parentheses when operated at f_X = 5.0 MHz

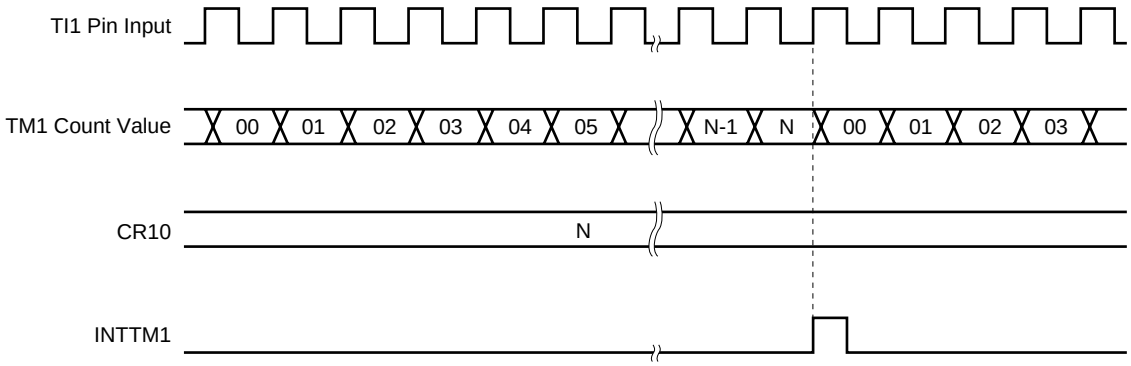
(2) External event counter operation

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 (CR10 and CR20), TM1 and TM2 are cleared to 0 and the interrupt request signals (INTTM1 and INTTM2) are generated.

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



Remark N = 00H to FFH

(3) Square-wave output

This operates as a square wave output with any selected frequency at intervals of the value preset to 8-bit compare registers 10 and 20 (CR10 and CR20).

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

Table 7-8. 8-Bit Timer/Event Counters 1 and 2 Square-Wave Output Ranges

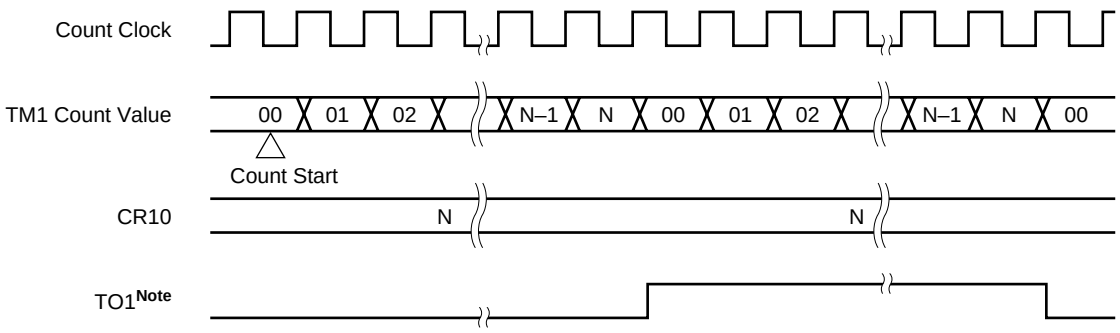
Minimum Pulse Width		Maximum Pulse Width		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
$2 \times 1/f_X$ (400 ns)	$2^2 \times 1/f_X$ (800 ns)	$2^9 \times 1/f_X$ (102.4 μs)	$2^{10} \times 1/f_X$ (204.8 μs)	$2 \times 1/f_X$ (400 ns)	$2^2 \times 1/f_X$ (800 ns)
$2^2 \times 1/f_X$ (800 ns)	$2^3 \times 1/f_X$ (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 : Oscillation mode selection register (OSMS) bit 0

3. Values in parentheses when operated at f_X = 5.0 MHz.

Figure 7-10. Timing of Square Wave Output Operation



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

7.4.2 16-bit timer/event counter mode

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

In this mode, the count clock is selected 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.

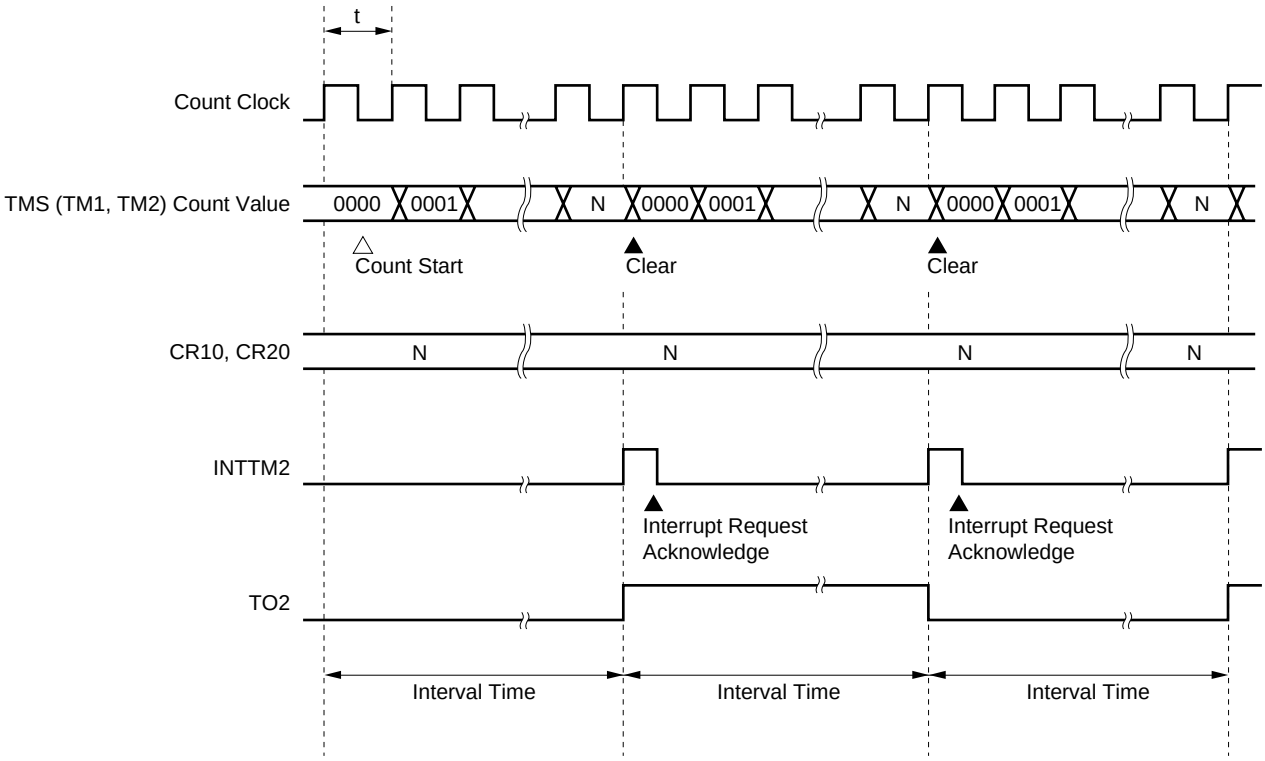
(1) Operation as interval timer

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 7-9**.

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 7-11**.

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 7-11. 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 (TMS) count value, use the 16-bit memory manipulation instruction.

Table 7-9. 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		$2^8 \times$ TI1 input cycle		TI1 input edge cycle	
0	0	0	1	TI1 input cycle		$2^8 \times$ TI1 input cycle		TI1 input edge cycle	
0	1	1	0	$2 \times 1/f_X$ (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 : Oscillation mode selection register (OSMS) bit 0

3. TCL10-TCL13: Bits 0 through 3 of the timer clock select register 1 (TCL1)

4. Values in parentheses when operated at f_X = 5.0 MHz.

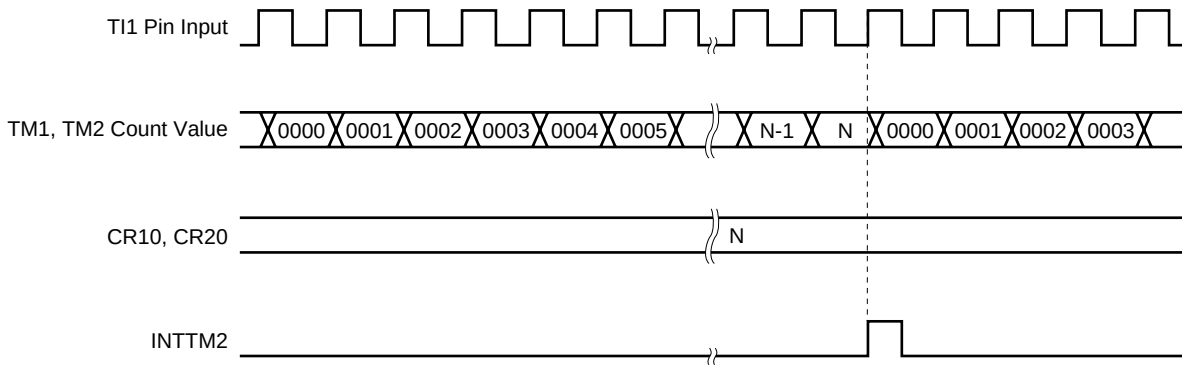
(2) External event counter operations

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

TM1 is incremented each time the valid edge specified with the timer clock select register 1 (TCL1) is input. When TM1 overflows, TM2 is incremented taking the overflow signal as 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 7-12. 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 (TMS) count value, use the 16-bit memory manipulation instruction.

(3) Square-wave output operation

This operates as a square wave output with any selected frequency at intervals of the value preset to 8-bit compare registers 10 and 20 (CR10 and CR20). When setting the count value, set the high-order 8 bits to CR20, and 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 7-10. Square-Wave Output Ranges when 2-Channel 8-Bit Timer/Event Counters (TM1 and TM2) are Used as 16-Bit Timer/Event Counter

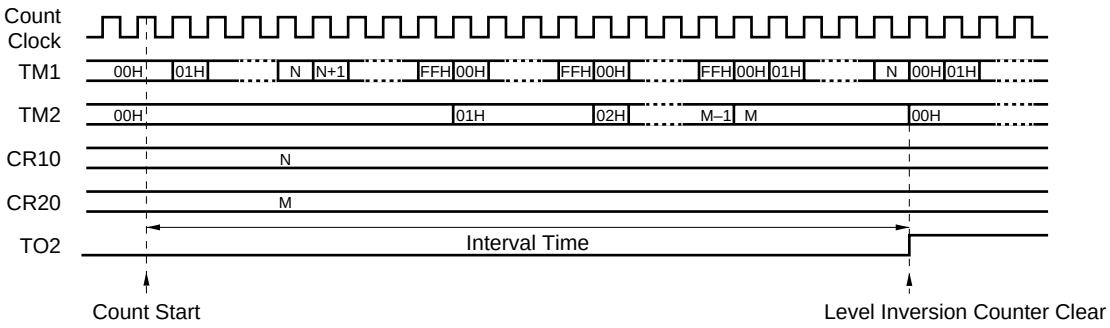
Minimum Pulse Width		Maximum Pulse Width		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
$2 \times 1/f_X$ (400 ns)	$2^2 \times 1/f_X$ (800 ns)	$2^{17} \times 1/f_X$ (26.2 ms)	$2^{18} \times 1/f_X$ (52.4 ms)	$2 \times 1/f_X$ (400 ns)	$2^2 \times 1/f_X$ (800 ns)
$2^2 \times 1/f_X$ (800 ns)	$2^3 \times 1/f_X$ (1.6 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 : Oscillation mode selection register (OSMS) bit 0

3. Values in parentheses when operated at f_X = 5.0 MHz.

Figure 7-13. Timing of Square Wave Output Operation

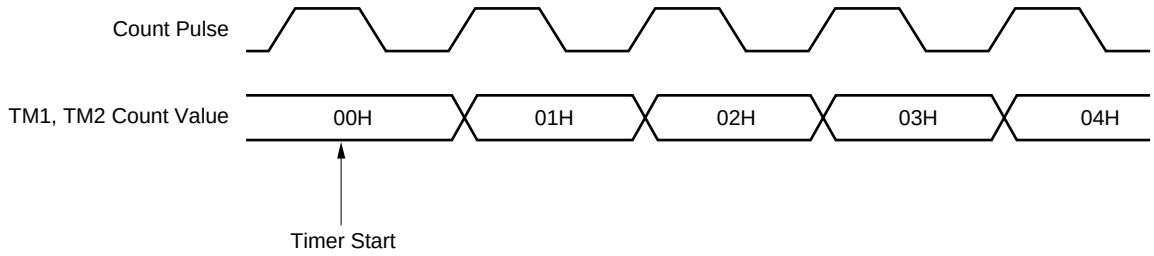


7.5 Cautions on 8-Bit Timer/Event Counters

(1) Timer start errors

An error with a maximum of one clock may occur concerning the time required for a match signal to be generated after timer start. This is because 8-bit timer registers 1 and 2 (TM1 and TM2) are started asynchronously with the count pulse.

Figure 7-14. 8-Bit Timer Registers 1 and 2 Start Timing



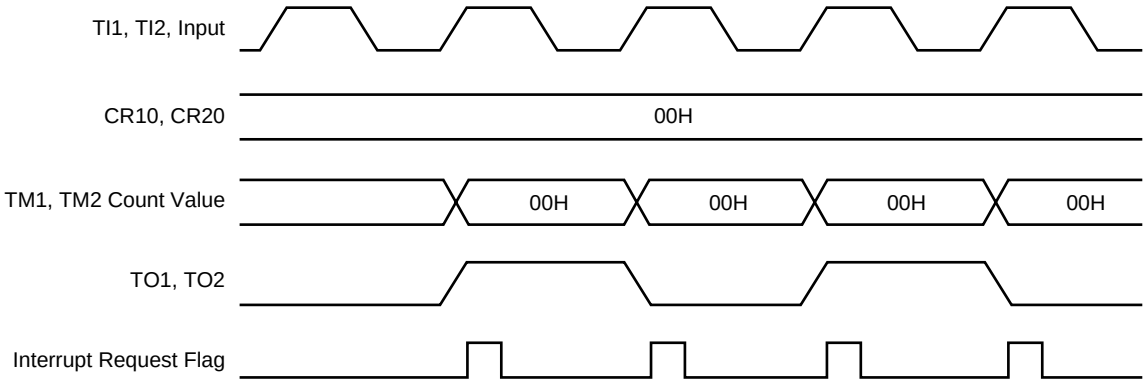
(2) 8-bit compare register 10 and 20 setting

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) to 0 and stopping timer operation.

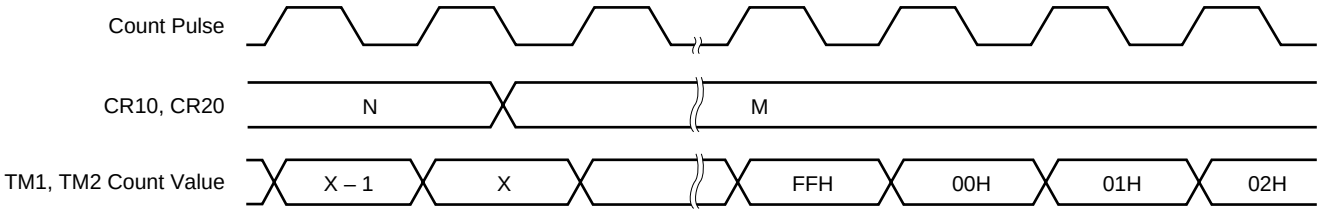
Figure 7-15. External Event Counter Operation Timing



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

If the values after the 8-bit compare registers 10 and 20 (CR10 and CR20) are changed are smaller than those of 8-bit timer registers 1 and 2 (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 7-16. Timing after Compare Register Change during Timer Count Operation



Remark N > X > M

CHAPTER 8 WATCH TIMER

8.1 Watch Timer Functions

The watch timer has the following functions.

- Watch timer
- Interval timer

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

(1) Watch timer

When the 32.768 kHz subsystem clock is used, a flag (WTIF) is set at 0.5 second or 0.25 second intervals. When the 4.19 MHz (4.194304 MHz TYP.) 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 8-1. 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})

8.2 Watch Timer Configuration

The watch timer consists of the following hardware.

Table 8-2. Watch Timer Configuration

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

8.3 Watch Timer Control Registers

The following two types of registers are used to control the watch timer.

- Timer clock select register 2 (TCL2)
- Watch timer mode control register (TMC2)

(1) Timer clock select register 2 (TCL2)

This register sets the watch timer count clock.

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

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

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

Figure 8-1. Watch Timer Block Diagram

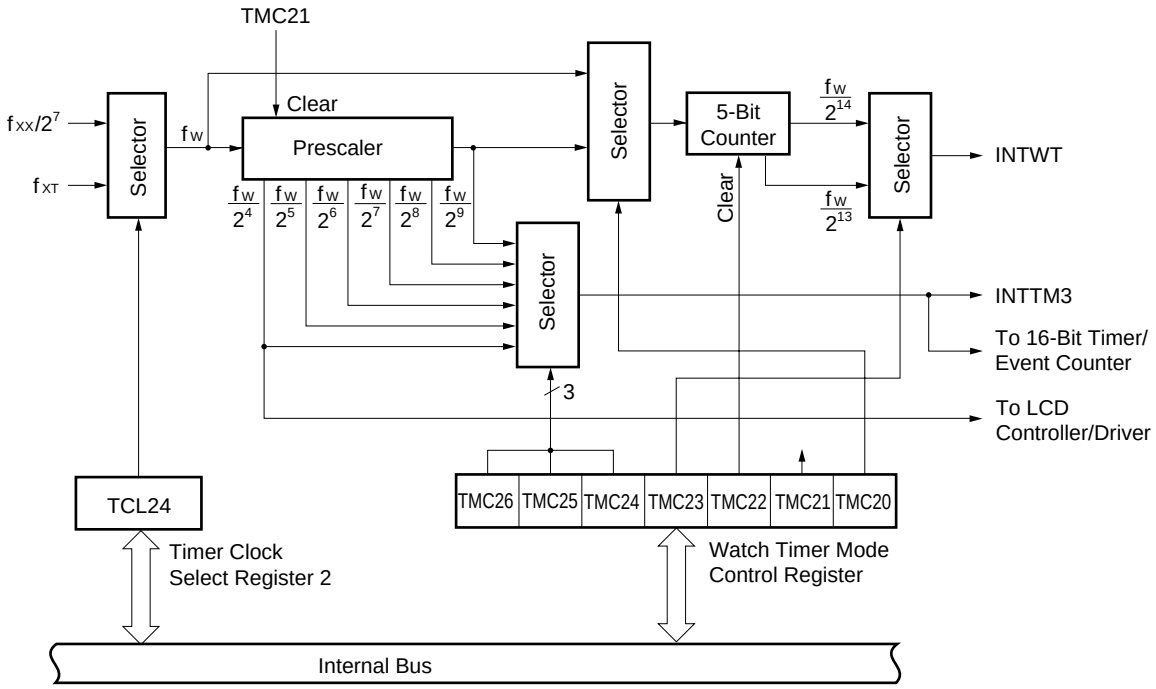


Figure 8-2. Timer Clock Select Register 2 Format

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 : Oscillation mode selection register (OSMS) bit 0

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

(2) Watch timer mode control register (TMC2)

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

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

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

Figure 8-3. 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.

Remark f_W : Watch timer clock frequency ($f_{XX}/2^7$ or f_{XT})
f_{XX} : Main system clock frequency (f_X or $f_X/2$)
f_X : Main system clock oscillation frequency
f_{XT} : Subsystem clock oscillation frequency

8.4 Watch Timer Operations

8.4.1 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 (TIMC22) 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).

8.4.2 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 8-3. 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})

CHAPTER 9 WATCHDOG TIMER

9.1 Watchdog Timer Functions

The watchdog timer has the following functions.

- Watchdog timer
- Interval timer

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

(1) Watchdog timer mode

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

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

Inadvertent Program Loop Detection Time	MCS = 1	MCS = 0
$2^{11} \times 1/f_{XX}$	$2^{11} \times 1/f_X$ (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 : Oscillation mode selection register (OSMS) bit 0

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

(2) Interval timer mode

Interrupt requests are generated at the preset time intervals.

Table 9-2. Interval Times

Interval Time	MCS = 1	CS = 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 : Oscillation mode selection register (OSMS) bit 0

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

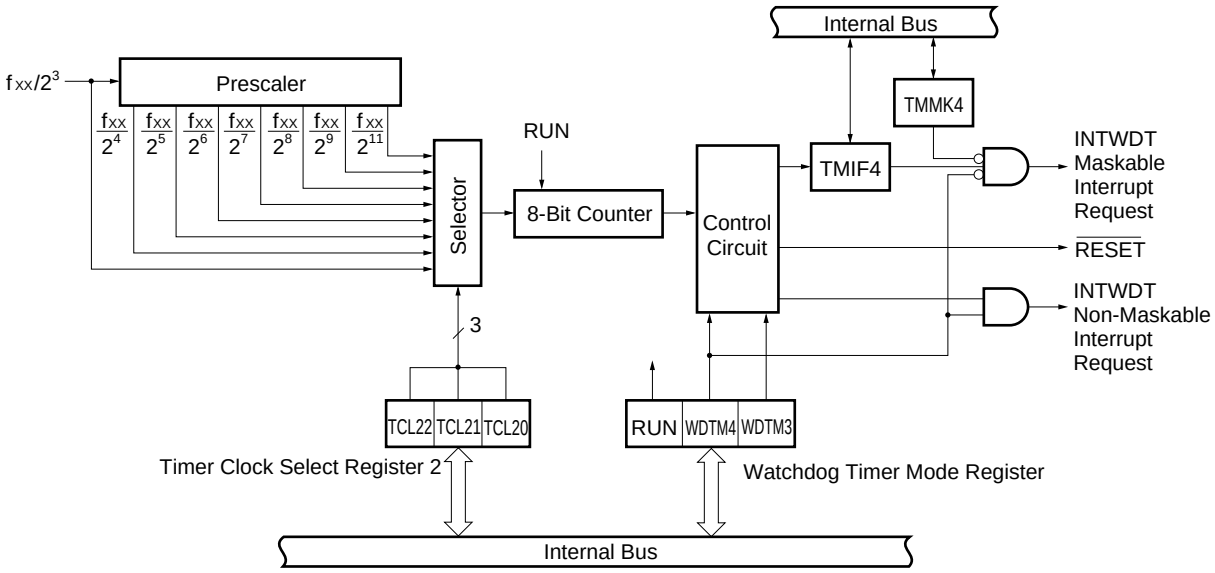
9.2 Watchdog Timer Configuration

The watchdog timer consists of the following hardware.

Table 9-3. Watchdog Timer Configuration

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

Figure 9-1. Watchdog Timer Block Diagram



9.3 Watchdog Timer Control Registers

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

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

(1) Timer clock select register 2 (TCL2)

This register sets the watchdog timer count clock.

TCL2 is set with 8-bit memory manipulation instruction.

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

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

Figure 9-2. Timer Clock Select Register 2 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL2	TCL27	TCL26	TCL25	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 : Oscillation mode selection register (OSMS) bit 0

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

(2) Watchdog timer mode register (WDTM)

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

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

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

Figure 9-3. Watchdog Timer Mode Register Format

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
WDTM	RUM	0	0	WDTM4	WDTM3	0	0	0	FFF9H	00H	R/W

WDTM4	WDTM3	Watchdog Timer Operation Mode Selection[Note 1]
0	×	Interval timer mode[Note 2] (Maskable interrupt occurs upon generation of an overflow.)
1	0	Watchdog timer mode 1 (Non-maskable interrupt occurs upon generation of an overflow.)
1	1	Watchdog timer mode 2 (Reset operation is activated upon generation of an overflow.)

RUN	Watchdog Timer Operation Mode Selection[Note 3]
0	Count stop
1	Counter is cleared and counting starts.

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

2. Interval timer operation is started at the time RUN is set to 1.

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

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

2. To use watchdog timer modes 1 and 2, confirm that the interrupt request flag (TMIF4) is 0, and then set WDTM4 to 1.
If WDTM4 is set to 1 while TMIF4 is 1, a non-maskable interrupt request occurs regardless of the contents of WDTM3.

Remark × : don't care

9.4 Watchdog Timer Operations

9.4.1 Watchdog timer operation

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

The watchdog timer count clock (inadvertent program loop detection time interval) can be selected with bits 0 to 2 (TCL20 to TCL22) of the timer clock select register 2 (TCL2).

Watchdog timer starts by setting bit 7 (RUN) of WDTM to 1. After the watchdog timer is started, set RUN to 1 within the set inadvertent program loop 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 inadvertent program loop detection time is past, system reset or a non-maskable interrupt request is generated according to the WDTM bit 3 (WDTM3) value.

The watchdog timer continues operating in the HALT mode but it stops in the STOP mode. Thus, set RUN to 1 before the STOP mode is set, clear the watchdog timer and then execute the STOP instruction.

Cautions 1. The actual inadvertent program loop detection time may be shorter than the set time by a maximum of 0.5 %.

2. When the subsystem clock is selected for CPU clock, watchdog timer count operation is stopped.

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

TCL22	TCL21	TCL20	Inadvertent Program Loop Detection Time	MCS = 1	MCS = 0
0	0	0	$2^{11} \times 1/f_{XX}$	$2^{11} \times 1/f_X$ (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 : Oscillation mode selection register (OSMS) bit 0

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

9.4.2 Interval timer operation

The watchdog timer operates as an interval timer which generates interrupt requests repeatedly at an interval of the preset count value when bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 1 and 0, respectively.

The count clock (interval time) can be selected by bits 0 to 2 (TCL20 to TCL22) of the timer clock select register 2 (TCL2). Interval timer operation is started by setting bit 7 (RUN) of WDTM to 1.

When the watchdog timer operated as interval timer, the interrupt mask flag (TMMK4) and priority specify flag (TMPR4) are validated and the maskable interrupt request (INTWDT) can be generated. Among maskable interrupt requests, the INTWDT default has the highest priority.

The interval timer continues operating in the HALT mode but it stops in STOP mode. Thus, set bit 7 of the WDTM (RUN) to 1 before the STOP mode is set, clear the interval timer and then execute the STOP instruction.

Cautions 1. Once bit 4 (WDTM4) of WDTM is set to 1 (with the watchdog timer mode selected), the interval timer mode is not set unless $\overline{\text{RESET}}$ 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 9-5. Interval Timer Interval Time

TCL22	TCL21	TCL20	Interval Time	MCS = 1	MCS = 0
0	0	0	$2^{11} \times 1/f_{XX}$	$2^{11} \times 1/f_X$ (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 : Oscillation mode selection register (OSMS) bit 0

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

CHAPTER 10 CLOCK OUTPUT CONTROL CIRCUIT

10.1 Clock Output Control Circuit Functions

The clock output control circuit is intended for carrier output during remote controlled transmission and clock output for supply to peripheral LSI. Clocks selected with the timer clock select register 0 (TCL0) are output from the PCL/P35 pin.

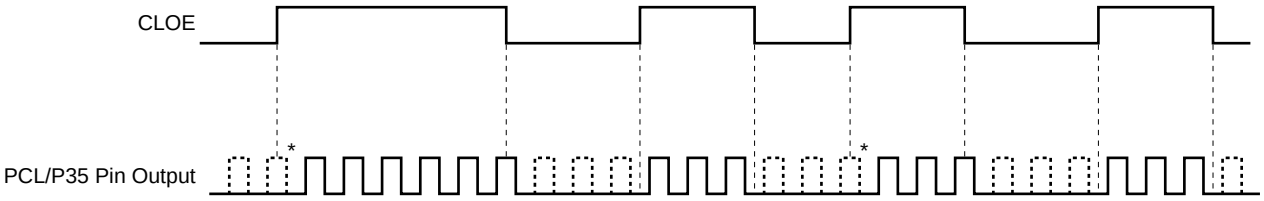
Follow the procedure below to output clock pulses.

(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 to 0 (set to output mode).
(4) Set bit 7 (CLOE) of 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 (Refer to the portions marked with * in **Figure 10-1**).

Figure 10-1. Remote Controlled Output Application Example



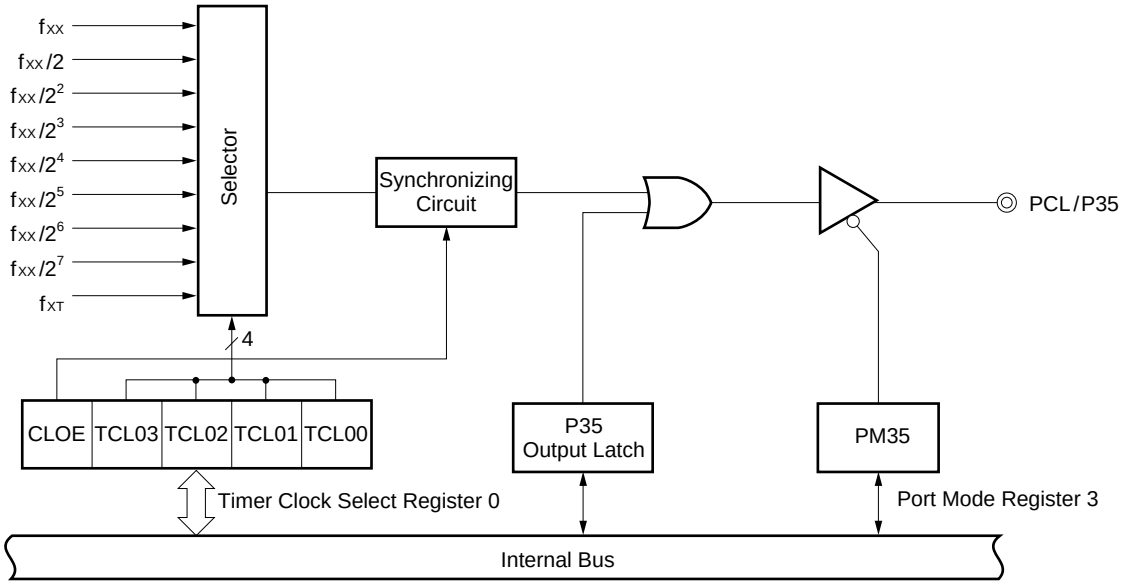
10.2 Clock Output Control Circuit Configuration

The clock output control circuit consists of the following hardware.

Table 10-1. Clock Output Control Circuit Configuration

Item	Configuration
Control register	Timer clock select register 0 (TCL0) Port mode register 3 (PM3)

Figure 10-2. Clock Output Control Circuit Block Diagram



10.3 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 sets TCL0 to 00H.

Remark Besides setting PCL output clock, TCL0 sets the 16-bit timer register count clock.

Cautions

1. **Setting of the TI00/P00/INTP0 pin valid edge is performed by external interrupt mode register 0 (INTM0), and selection of the sampling clock frequency is performed by the sampling clock selection register (SCS).**
2. **When enabling PCL output, set TCL00 to TCL03, then set 1 in CLOE with a 1-bit memory manipulation instruction.**
3. **To read the count value when TI00 has been specified as the TM0 count clock, the value should be read from TM0, not from capture/compare register 01 (CR01).**
4. **When rewriting TCL0 to other data, stop the timer operation beforehand.**

Figure 10-3. 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

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 : Oscillation mode selection register (OSMS) bit 0

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 10-4. 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)

[MEMO]

CHAPTER 11 BUZZER OUTPUT CONTROL CIRCUIT

11.1 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 to 0 (Set to output mode).

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

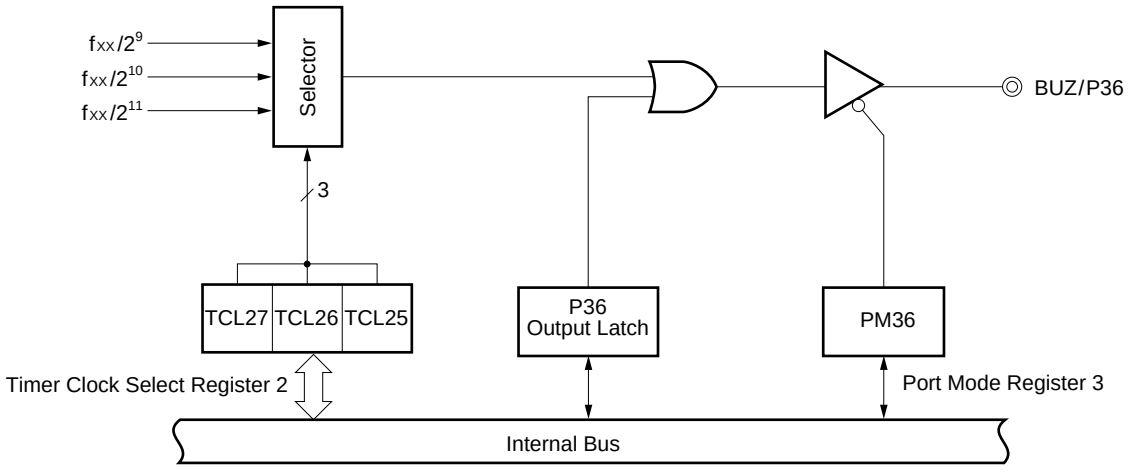
11.2 Buzzer Output Control Circuit Configuration

The buzzer output control circuit consists of the following hardware.

Table 11-1. Buzzer Output Control Circuit Configuration

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

Figure 11-1. Buzzer Output Control Circuit Block Diagram



11.3 Buzzer Output Function Control Registers

The following two types of registers are used to control the buzzer output function.

- Timer clock select register 2 (TCL2)
- Port mode register 3 (PM3)

(1) Timer clock select register 2 (TCL2)

This register sets the buzzer output frequency.

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

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

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

Figure 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 : Oscillation mode selection register (OSMS) bit 0

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 11-3. 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 12 A/D CONVERTER

12.1 A/D Converter Functions

The A/D converter converts an analog input into a digital value. It consists of 8 channels (ANI0 to ANI7) with an 8-bit resolution.

The conversion method is based on successive approximation and the conversion result is held in the 8-bit A/D conversion result register (ADCR).

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).

Select one channel of analog input from ANI0 to ANI7 to carry out A/D conversion. In the case of hardware start, A/D conversion operation stops when an A/D conversion operation ends and an interrupt request (INTAD) is generated. In the case of software start, the A/D conversion operation is repeated. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

Caution Do not perform the following operation on the pins shared with port pins (refer to (1) Port pins in 2.1.1 Normal operating mode pins) during A/D conversion operation; otherwise, the specifications of the absolute accuracy during A/D conversion cannot be satisfied (except the pins shared with LCD segment output pins).

(1) Rewriting the output latch of an output pin used as a port pin

(2) Changing the output level of an output pin even when it is not used as a port pin

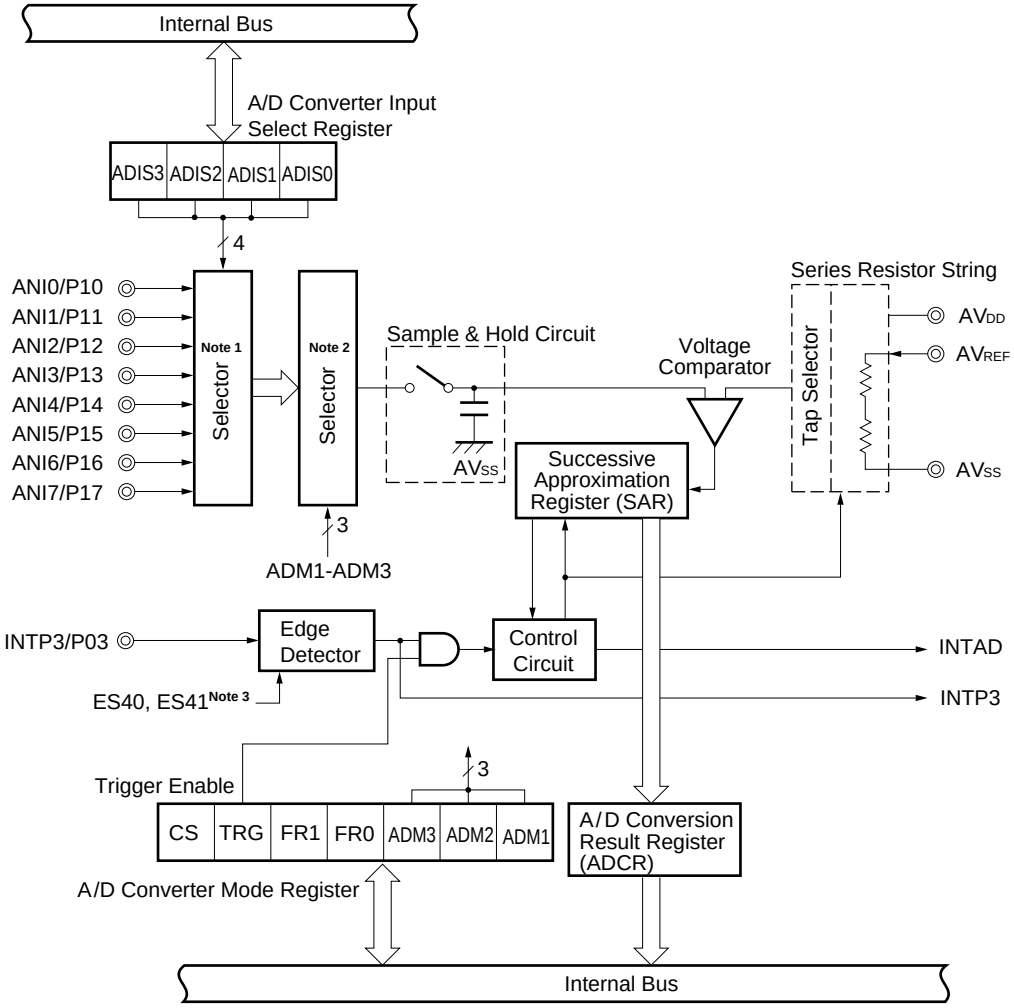
12.2 A/D Converter Configuration

The A/D converter consists of the following hardware.

Table 12-1. A/D Converter Configuration

Item	Configuration
Analog input	8 channels (ANI0 to ANI7)
Control 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 12-1. A/D Converter Block Diagram



Notes 1. Selector to select the number of channels to be used for analog input.

2. Selector to select the channel for A/D conversion.

3. Bits 0, 1 of the 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 retained (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_{REF} to AV_{SS} and generates a voltage to be compared to the analog input.

(6) ANI0 to ANI7 pins

These are 8-channel analog input pins to input analog signals to undergo A/D conversion to the A/D converter. Pins other than those selected as analog input by the A/D converter input select register (ADIS) can be used as input/output ports.

Cautions 1. Use ANI0 to ANI7 input voltages within the specified range. If a voltage higher than AV_{REF} 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 (PORT1) pins. When A/D conversion is performed with any of pins ANI0 to ANI7 selected, be sure not to execute a PORT1 input instruction while conversion is in progress, as this may reduce the conversion resolution.

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

(7) AV_{REF} pin

This pin inputs the A/D converter reference voltage.

It converts signals input to ANI0 to ANI7 into digital signals according to the voltage applied between AV_{REF} and AV_{SS}.

The current flowing in the series resistor string can be reduced by setting the voltage to be input to the AV_{REF} pin to AV_{SS} level in standby mode.

Caution A series resistor string of about 10 kΩ is connected between the AV_{REF} and AV_{SS} pins. Therefore, if the output impedance of the reference voltage source is high, the impedance is virtually connected in parallel with the series resistor string between the AV_{REF} and AV_{SS} pin, increasing the error of the reference voltage.

(8) AV_{SS} pin

This is a GND potential pin of the A/D converter. Keep it at the same potential as the V_{SS} pin when not using the A/D converter.

(9) AV_{DD} pin

This is an A/D converter analog power supply pin. Keep it at the same potential as the V_{SS} pin when not using the A/D converter.

12.3 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 12-2. 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)
1	1	1	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.

Remark f_X : Main system clock oscillation frequency

MCS : Oscillation mode selection register (OSMS) bit 0

(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 sets ADIS to 00H.

Cautions 1. Set the analog input channel in the following order.

(1) Set the number of analog input channels with ADIS.

(2) Using 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.

Figure 12-3. 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-P17)
0	0	0	1	1 channel (ANI0, P11-P17)
0	0	1	0	2 channel (ANI0, ANI1, P12-P17)
0	0	1	1	3 channel (ANI0-ANI2, P13-P17)
0	1	0	0	4 channel (ANI0-ANI3, P14-P17)
0	1	0	1	5 channel (ANI0-ANI4, P15-P17)
0	1	1	0	6 channel (ANI0-ANI5, P16, P17)
0	1	1	1	7 channel (ANI0-ANI6, P17)
1	0	0	0	8 channel (ANI0-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 sets INTM1 to 00H.

Figure 12-4. 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

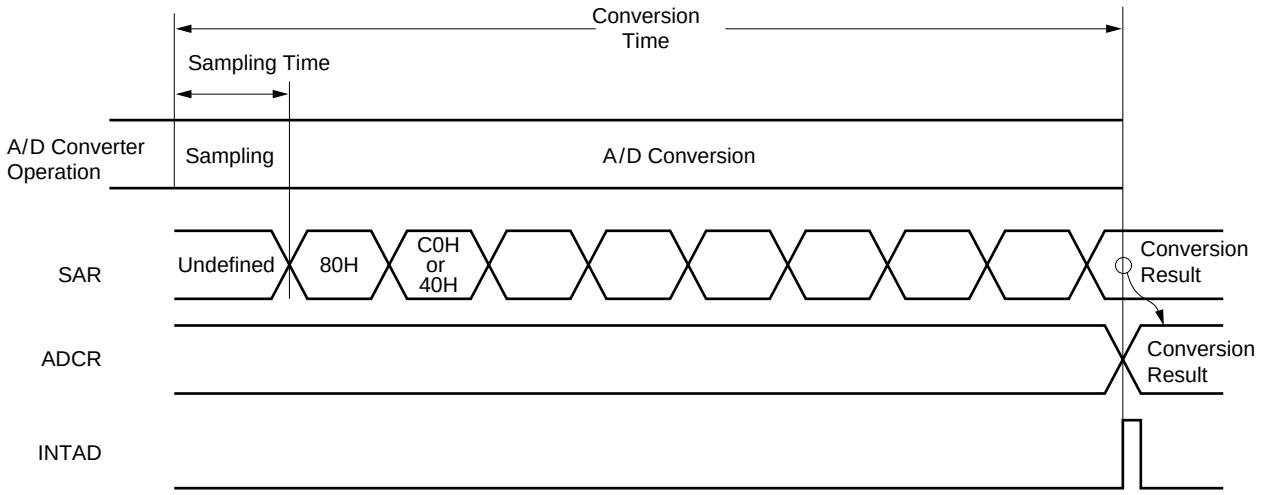
12.4 A/D Converter Operations

12.4.1 Basic operations of A/D converter

(1) Set the number of analog input channels with A/D converter input select register (ADIS).
(2) From among the analog input channels set with ADIS, select one channel for A/D conversion with A/D converter mode register (ADM).
(3) Sample the voltage input to the selected analog input channel with the sample & hold circuit.
(4) Sampling for the specified period of time sets the sample & hold circuit to the hold state so that the circuit holds the input analog voltage until termination of A/D conversion.
(5) Bit 7 of the successive approximation register (SAR) is set and the tap selector sets the series resistor string voltage tap to (1/2) AV_{REF}.
(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_{REF}, the MSB of SAR remains set. If the input is smaller than (1/2) AV_{REF}, 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_{REF}
    - Bit 7 = 0 : (1/4) AV_{REF}

    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 12-5. A/D Converter Basic Operation



A/D conversion operations are performed continuously until the bit 7 (CS) of A/D converter mode register (ADM) is reset (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 (1), conversion starts again from the beginning.

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

12.4.2 Input voltage and conversion results

The relation between the analog input voltage input to the analog input pins (ANI0 to ANI7) and the A/D conversion result (the value stored in A/D conversion result register (ADCR)) is shown by the following expression.

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

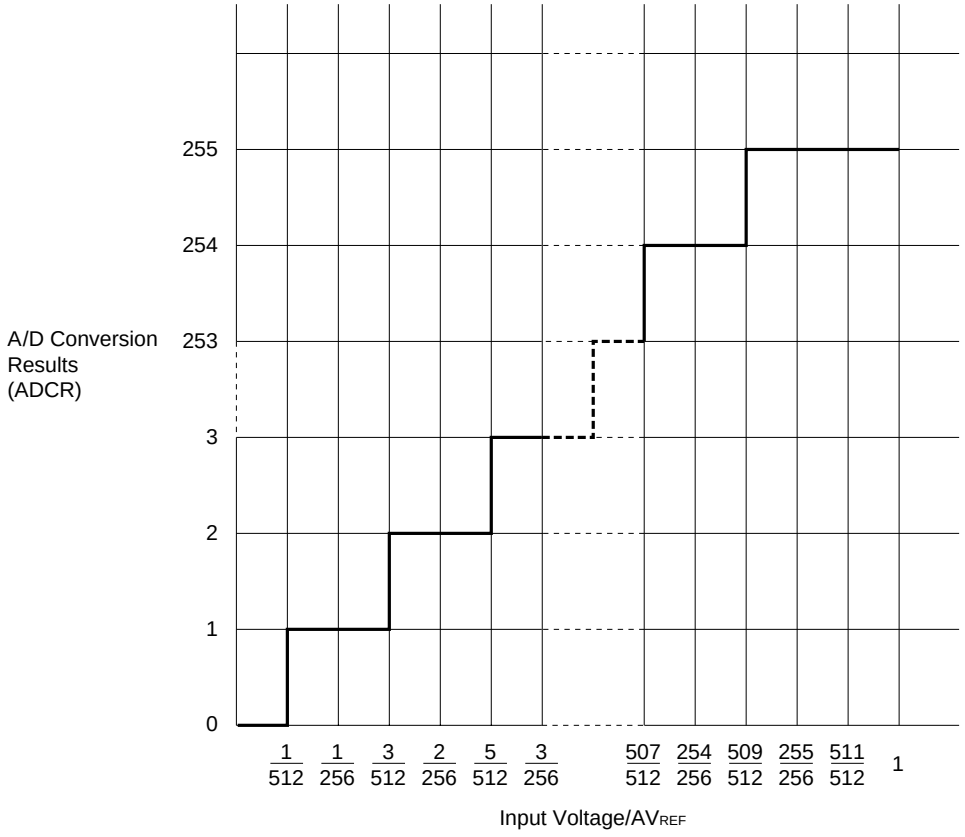
or

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

INT( ) : Function which returns integer parts of value in parentheses.
V_{IN} : Analog input voltage
AV_{REF} : AV_{REF} pin voltage
ADCR : A/D conversion result register (ADCR) register value

Figure 12-6 shows the relation between the analog input voltage and the A/D conversion result.

Figure 12-6. Relations between Analog Input Voltage and A/D Conversion Result



12.4.3 A/D converter operating mode

Select one analog input channel from among ANI0 to ANI7 with the A/D converter input select register (ADIS) and A/D converter mode register (ADM) and execute A/D conversion.

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

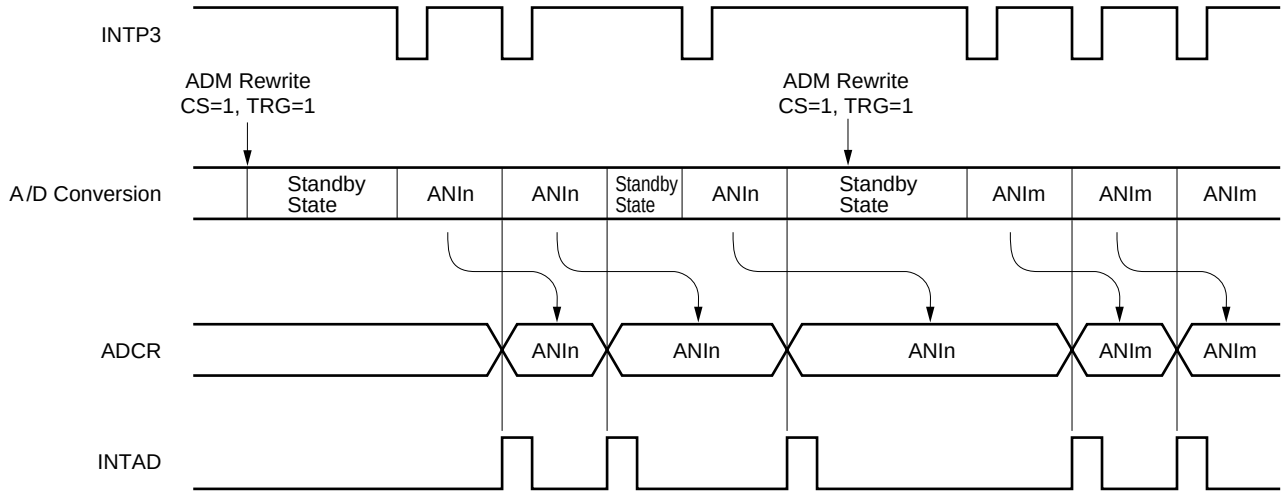
When bit 6 (TRG) and bit 7 (CS) of ADM are set to 1, the A/D conversion standby state is set. When the external trigger signal (INTP3) is input, the A/D conversion starts on the voltage applied to the analog input pins specified with bits 1 to 3 (ADM1 to ADM3) of A/D converter mode register (ADM).

Upon termination of the A/D conversion, the conversion result is stored in the A/D conversion result register (ADCR) and the interrupt request signal (INTAD) is generated. After one A/D conversion operation is started and terminated, another operation is not started until a new external trigger signal is input.

If data with CS set to 1 is written to ADM again during A/D conversion, the converter suspends its A/D conversion operation and waits for a new external trigger signal to be input. When the external trigger input signal is reinput, A/D conversion is carried out from the beginning.

If data with CS set to 0 is written to ADM during A/D conversion, the A/D conversion operation stops immediately.

Figure 12-7. A/D Conversion by Hardware Start



Remark n = 0, 1, ... , 7
m = 0, 1, ... , 7

(2) A/D conversion operation in software start

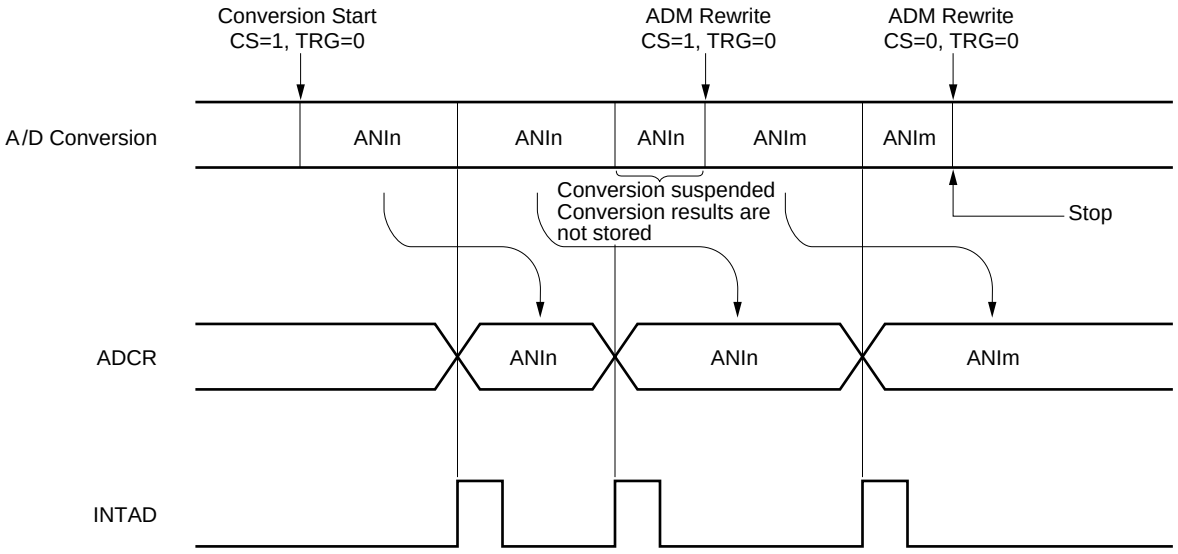
When bit 6 (TRG) and bit 7 (CS) of A/D converter mode register (ADM) are set to 0 and 1, respectively, the A/D conversion starts on the voltage applied to the analog input pins specified with bits 1 to 3 (ADM1 to ADM3) of ADM.

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 12-8. A/D Conversion by Software Start



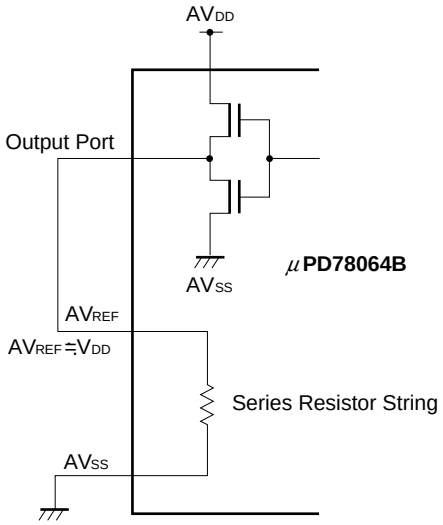
Remark n = 0, 1, ... , 7
m = 0, 1, ... , 7

12.5 A/D Converter Cautions

(1) Current consumption in standby mode

The A/D converter operates on the main system clock. Therefore, its operation stops in STOP mode or in HALT mode with the subsystem clock. As a current still flows in the AV_{REF} pin at this time, this current must be cut in order to minimize the overall system power dissipation. In Figure 12-9, 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_{REF} voltage, and therefore the conversion values themselves lack precision and can only be used for relative comparison.

Figure 12-9. Example of Method of Reducing Current Consumption in Standby Mode



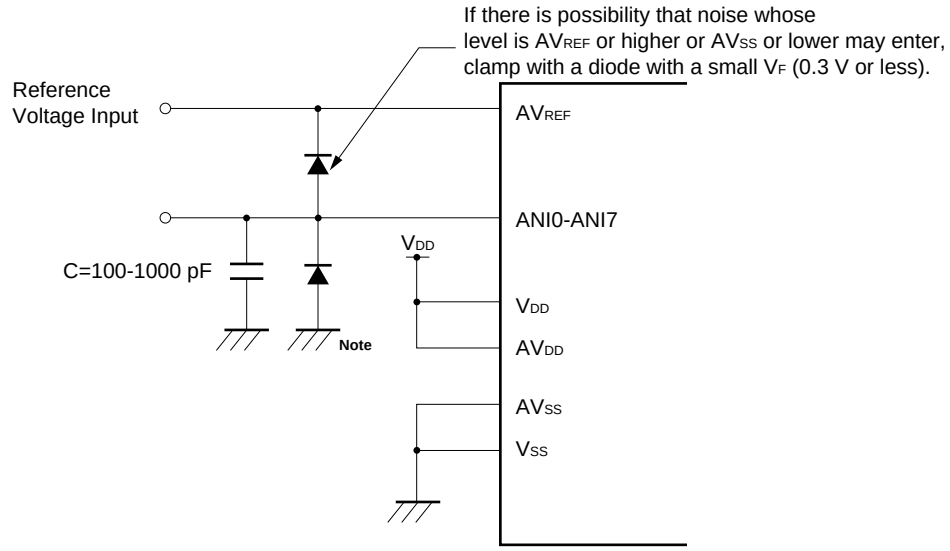
(2) Input range of ANI0 to ANI7

The input voltages of ANI0 to ANI7 should be within the specification range. In particular, if a voltage above AV_{REF} or below AV_{SS} is input (even if within the absolute maximum rating range), the conversion value for that channel will be indeterminate. The conversion values of the other channels may also be affected.

(3) Noise countermeasures

In order to maintain 8-bit resolution, attention must be paid to noise on pins AV_{REF} and ANI0 to ANI7. Since the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally as shown in Figure 12-10 in order to reduce noise.

Figure 12-10. Analog Input Pin Disposition



Note To reduce EMI noise, supply separate power to V_{DD} and AV_{DD} from separate sources, and separately ground V_{SS} and AV_{SS}.

(4) Pins ANI0/P10 to ANI7/P17

The analog input pins ANI0 to ANI7 also function as input/output port (PORT1) pins. When A/D conversion is performed with any of pins ANI0 to ANI7 selected, be sure not to execute a PORT1 input instruction while conversion is in progress, as this may reduce the conversion resolution.

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

(5) AV_{REF} pin input impedance

A series resistor string of approximately 10 kΩ is connected between the AV_{REF} pin and the AV_{SS} pin.

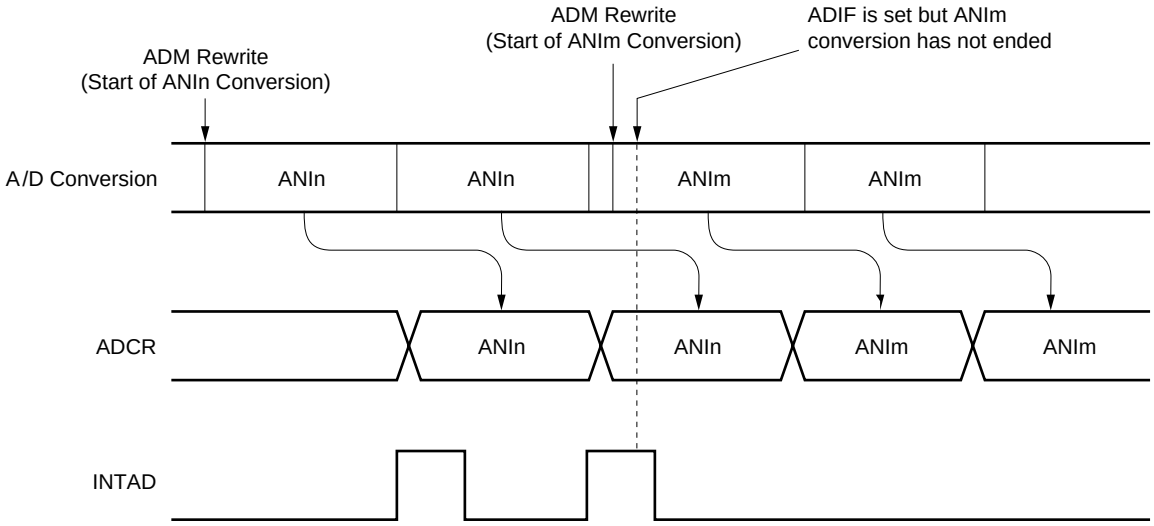
Therefore, if the output impedance of the reference voltage source is high, this will result in parallel connection to the series resistor string between the AV_{REF} pin and the AV_{SS} pin, and there will be a large reference voltage error.

(6) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the A/D converter mode register (ADM) is changed. Caution is therefore required since, if a change of analog input pin is performed during A/D conversion, the A/D conversion result and ADIF for the pre-change analog input may be set just before the ADM rewrite, and when ADIF is read immediately after the ADM rewrite, ADIF may be set despite the fact that the A/D conversion for the post-change analog input has not ended.

When the A/D conversion is stopped and then resumed, clear the ADIF before it is resumed.

Figure 12-11. A/D Conversion End Interrupt Generation Timing

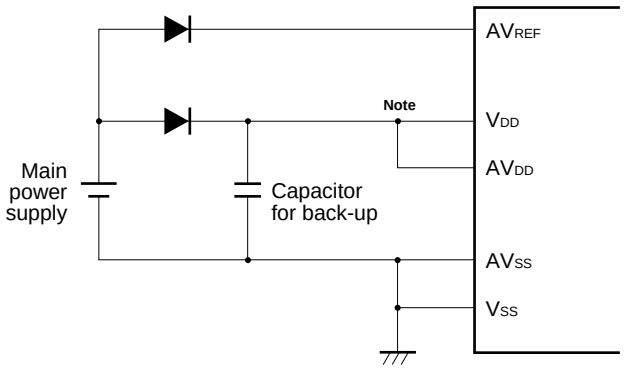


(7) AV_{DD} pin

The AV_{DD} pin is the analog circuit power supply pin, and supplies power to the input circuits of ANI0/P10 to ANI7/P17.

Therefore, be sure to apply the same voltage as V_{DD} to this pin even when the application circuit is designed so as to switch to a backup battery.

Figure 12-12. Handling of AV_{DD} Pin



Note To reduce EMI noise, supply separate power to V_{DD} and AV_{DD} from separate sources, and separately ground V_{SS} and AV_{SS}.

(8) Port operation during A/D converter operation

Do not perform the following operation on the pins shared with port pins (refer to **(1) Port pins** in **2.1.1 Normal operating mode pins**) during A/D conversion operation; otherwise, the specifications of the absolute accuracy during A/D conversion cannot be satisfied (except the pins shared with LCD segment output pins).

(1) Rewriting the output latch of an output pin used as a port pin

(2) Changing the output level of an output pin even when it is not used as a port pin

CHAPTER 13 SERIAL INTERFACE CHANNEL 0

The μPD78064B subseries incorporates two channels of serial interfaces. Differences between channels 0 and 2 are as follows (Refer to **CHAPTER 14 SERIAL INTERFACE CHANNEL 2** for details of the serial interface channel 2).

Table 13-1. Differences between Channels 0 and 2

Serial Transfer Mode		Channel 0	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	External clock, baud rate generator output
	Transfer method	MSB/LSB switchable as the start bit	MSB/LSB switchable as the start bit
	Transfer end flag	Serial transfer end interrupt request flag (CSIIF0)	Serial transfer end interrupt request flag (SRIF)
SBI (serial bus interface)		Use possible	None
2-wire serial I/O			
UART (Asynchronous serial interface)		None	Use possible

13.1 Serial Interface Channel 0 Functions

Serial interface channel 0 employs the following four modes.

- Operation stop mode
- 3-wire serial I/O mode
- SBI (serial bus interface) mode
- 2-wire serial I/O mode

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

(1) Operation stop mode

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

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

This mode is used for 8-bit data transfer using three lines, one each for serial clock ($\overline{\text{SCK0}}$), 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).

In the SBI mode, transfer data is classified into “addresses”, “commands”, and “data” in conformance with NEC serial bus format, and is transmitted or received.

- Address : Data to select target device for serial communication
- Command : Data that gives direction to target device
- Data : Data actually transferred

Actual transfer takes place when the master device outputs an “address” onto the serial bus to select a slave device from multiple devices with which the master is to communicate. “Commands” and “data” are then transmitted between the master and selected slave device. In this way, serial communication is implemented.

The receiver can automatically identify the received data as “address”, “command”, or “data in hardware.

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

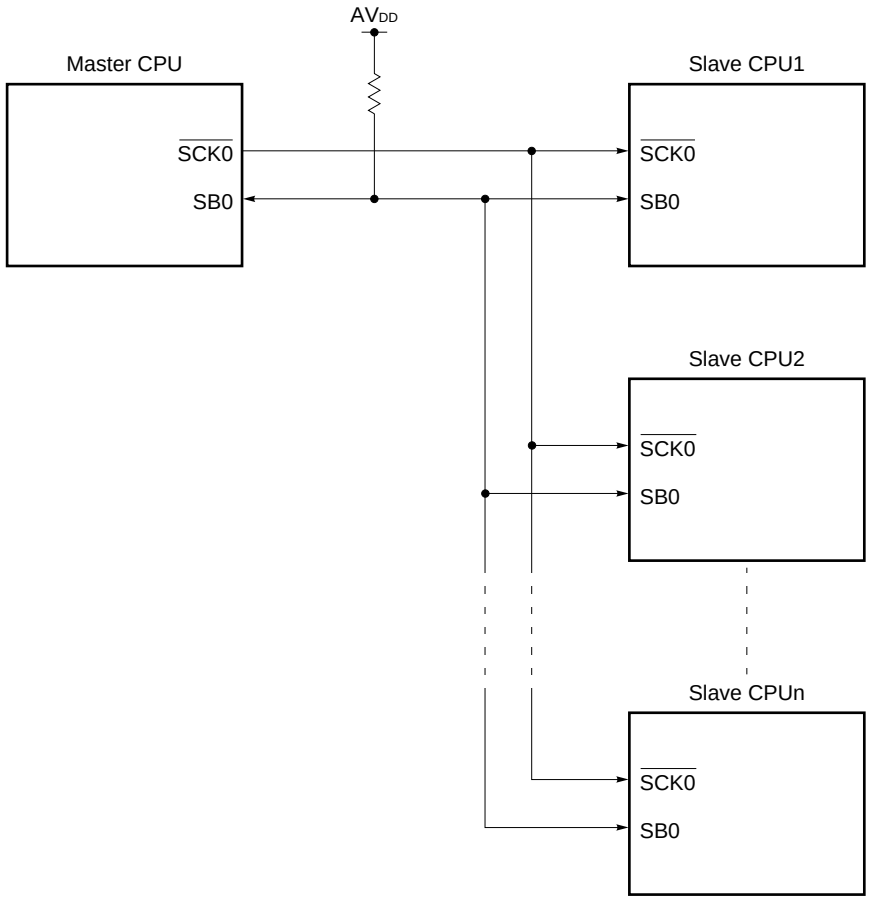
In this mode, the wake-up function for handshake and the output function of acknowledge and busy signals can also be used.

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

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

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

Figure 13-1. Serial Bus Interface (SBI) System Configuration Example



13.2 Serial Interface Channel 0 Configuration

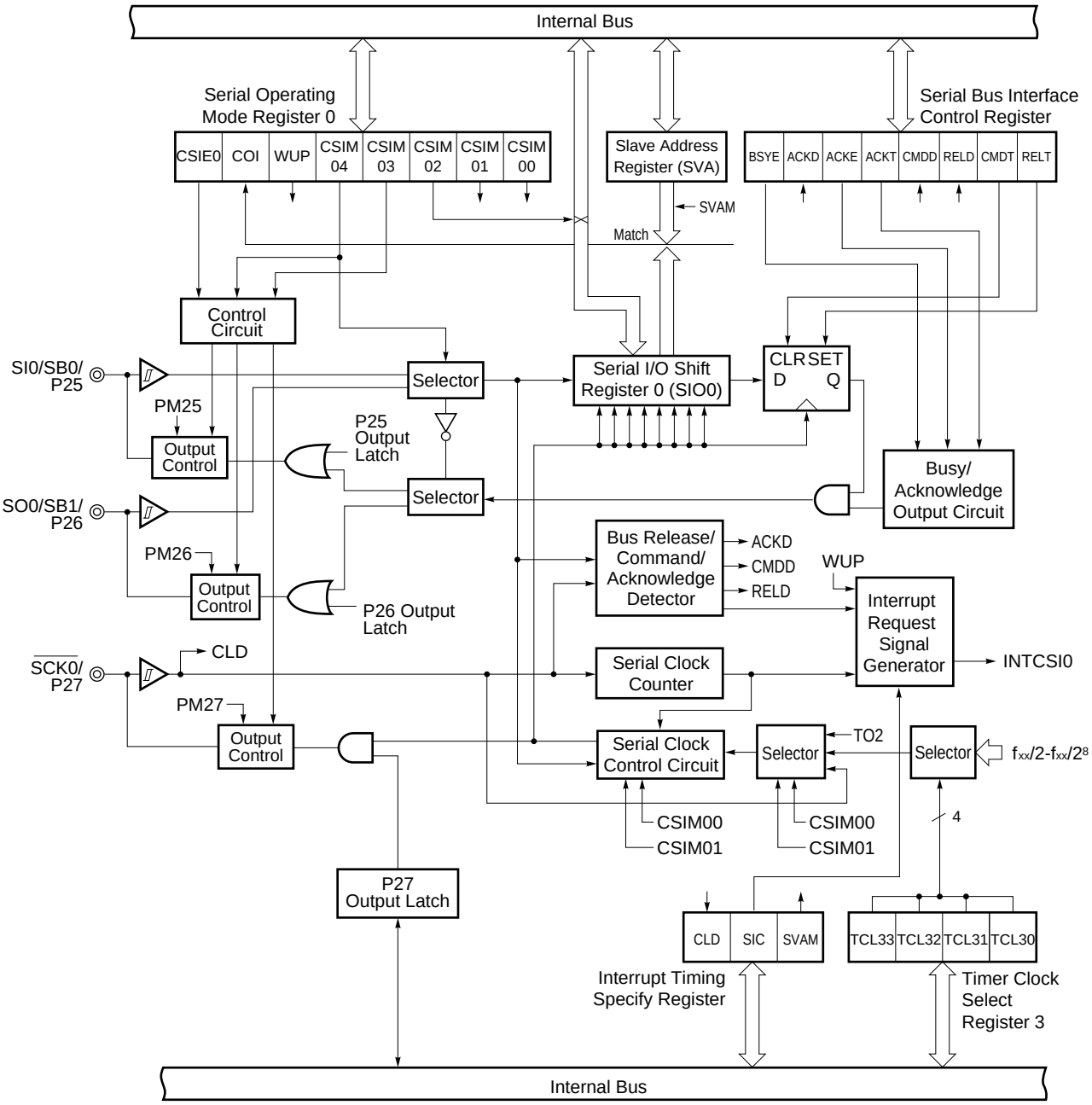
Serial interface channel 0 consists of the following hardware.

Table 13-2. 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 4-5 Block Diagram of P25 and P26**, **Figure 4-6 Block Diagram of P27**.

Figure 13-2. 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 SBI mode or 2-wire serial I/O mode, the bus pin must serve for both input and output. Thus, in the case of a device for reception, write FFH to SIO0 in advance (except when address reception is carried out by setting bit 5 (WUP) of CSIM0 to 1).

In the SBI mode, the busy state can be cleared by writing data to SIO0. In this case, bit 7 (BSYE) of the serial bus interface control register (SBIC) is not cleared to 0.

$\overline{\text{RESET}}$ input makes SIO0 undefined.

(2) Slave address register (SVA)

This is an 8-bit register to set the slave address value for connection of a slave device to the serial bus.

SVA is set with an 8-bit memory manipulation instruction. It is not used in the 3-wire serial I/O mode.

The master device outputs a slave address for selection of a particular slave device to the connected slave device. These two data (the slave address output from the master device and the SVA value) are compared with an address comparator. If they match, the slave device has been selected. In that case, bit 6 (COI) of serial operating mode register 0 (CSIM0) becomes 1.

Address can also be compared on the data of LSB-masked high-order 7 bits by setting (1) bit 4 (SVAM) of the interrupt timing specify register (SINT).

If no matching is detected in address reception, bit 2 (RELD) of the serial bus interface control register (SBIC) is cleared to 0. In the SBI mode, the wake-up function can be used by setting bit 5 (WUP) of CSIM0 to 1. In this case, an interrupt request signal (INTCSI0) is generated only when the slave address output by the master coincides with the value of SVA. This interrupt indicates that the master has issued a request for communication. While bit 5 (SIC) of the interrupt timing specification register (SINT) is set to 1, the wake-up function does not operate even if WUP is set to 1 (the interrupt signal is generated when bus release is detected). Clear SIC to 0 to use the wake-up function.

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

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

(3) SO0 latch

This latch holds SI0/SB0/P25 and SO0/SB1/P26 pin levels. It can be directly controlled by software. In the SBI mode, this latch is set upon termination of the 8th serial clock.

(4) Serial clock counter

This counter counts the serial clocks to be output and input during transmission/reception and to check whether 8-bit data has been transmitted/received.

(5) Serial clock control circuit

This circuit controls serial clock supply to the serial I/O shift register 0 (SIO0). When the internal system clock is used, the circuit also controls clock output to the $\overline{\text{SCK0}}$/P27 pin.

(6) Interrupt request signal generator

This circuit controls interrupt request signal generation. It generates the interrupt request signal in the following cases.

- In the 3-wire serial I/O mode and 2-wire serial I/O mode
  This circuit generates an interrupt request signal every eight serial clocks.

- In the SBI mode
  When WUP**Note** is 0..... Generates an interrupt request signal every eight serial clocks.
  When WUP**Note** is 1..... Generates an interrupt request signal when the serial I/O shift register 0 (SIO0) value matches the slave address register (SVA) value after address reception.

Note WUP is wake-up function specify bit. It is bit 5 of serial operating mode register 0 (CSIM0). To use the wake-up function (WUP = 1), clear bit 5 (SIC) of the interrupt timing specification register (SINT) to 0.

(7) Busy/acknowledge output circuit and bus release/command/acknowledge detector

These two circuits output and detect various control signals in the SBI mode.
These do not operate in the 3-wire serial I/O mode and 2-wire serial I/O mode.

13.3 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 13-3. Timer Clock Select Register 3 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL3	1	0	0	0	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		

Cautions 1. Set bits 4 to 6 to 0, and bit 7 to 1.

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

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

(2) Serial operating mode register 0 (CSIM0)

This register sets serial interface channel 0 serial clock, operating mode, operation enable/stop wake-up function and displays the address comparator match signal.

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM0 to 00H.

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

Figure 13-4. 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 $\overline{SCK0}$ 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	$\overline{SCK0}$/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)	$\overline{SCK0}$ (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)	$\overline{SCK0}$ (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)	$\overline{SCK0}$ (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)	

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

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

3. Can be used freely as port function.

Remark × : don't care

PM×× : port mode register

P×× : output latch of port

Figure 13-4. 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 data in SBI mode

R	COI	Slave Address Comparison Result Flag[Note 2]
	0	Slave address register not equal to serial I/O shift register 0 data
	1	Slave address register equal to serial I/O shift register 0 data

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

Notes 1. To use the wake-up function (WUP = 1), clear bit 5 (SIC) of the interrupt timing specification 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 sets SBIC to 00H.

Figure 13-5. 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		
R/W	RELT	Used for bus release signal output. When RELT = 1, SO latch is set to 1. After SO latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W		
R/W	CMDT	Used for command signal output. When CMDT = 1, SO latch is cleared to (0). After SO latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R	RELD	Bus Release Detection	
	Clear Conditions (RELD = 0)		Set Conditions (RELD =1)
	• When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception • When $\overline{\text{CSIE0}}$ = 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 $\overline{\text{CSIE0}}$ = 0 • When $\overline{\text{RESET}}$ input is applied		• When command signal (CMD) is detected

R/W		
R/W	ACKT	Acknowledge signal is output in synchronization with the falling edge clock of $\overline{\text{SCK0}}$ just after execution of the instruction to be set to 1, and after acknowledge signal output, automatically cleared to 0. Used as ACKE=0. Also cleared to 0 upon start of serial interface transfer or when CSIE0 = 0.

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

Remarks 1. Bits 0, 1, and 4 (RELD, CMDT, and ACKT) are 0 when read after data setting.

2. CSIE0: Bit 7 of the serial operation mode register 0 (CSIM0)

Figure 13-5. Serial Bus Interface Control Register Format (2/2)

R/W

ACKE	Acknowledge Signal Output Control	
0	Acknowledge signal automatic output disable (output with ACKT enable)	
1	Before completion of transfer	Acknowledge signal is output in synchronization with the 9th clock falling edge of $\overline{SCK0}$ (automatically output when ACKE = 1).
	After completion of transfer	Acknowledge signal is output in synchronization with the falling edge of $\overline{SCK0}$ just after execution of the instruction to be set to 1 (automatically output when ACKE = 1). However, not automatically cleared to 0 after acknowledge signal output.

R

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

R/W

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

Note The busy mode can be canceled by start of serial interface transfer. However, the BSYE flag is not cleared to 0.

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

(4) Interrupt timing specify register (SINT)

This register sets the bus release interrupt and address mask functions and displays the $\overline{SCK0}$ pin level status.

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

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

Figure 13-6. 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 Cause Selection[Note 2]
0	CSIIF0 is set upon termination of serial interface channel 0 transfer
1	CSIIF0 is set upon bus release detection or termination of serial interface channel 0 transfer

R

CLD	$\overline{SCK0}$ Pin Level[Note 3]
0	Low level
1	High level

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

2. When using 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 operation mode register 0 (CSIM0)

13.4 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

13.4.1 Operation stop mode

Serial transfer is not carried out in the operation stop mode. Thus, power consumption can be reduced. The serial I/O shift register 0 (SIO0) does not carry out shift operation either and thus it can be used as ordinary 8-bit register.

In the operation stop mode, the P25/SI0/SB0, P26/SO0/SB1 and P27/$\overline{\text{SCK0}}$ pins can be used as ordinary input/output ports.

(1) Register setting

The operation stop mode is set with the serial operating mode register 0 (CSIM0).

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM0 to 00H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W

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

13.4.2 3-wire serial I/O mode operation

The 3-wire serial I/O mode is valid for connection of peripheral I/O units and display controllers which incorporate a conventional synchronous clocked serial interface as is the case with the 75X/XL, 78K, and 17K series.

Communication is carried out with three lines of serial clock ($\overline{SCK0}$), serial output (SO0), and serial input (SI0).

(1) Register setting

The 3-wire serial I/O mode is set with the serial operating mode register 0 (CSIM0) and serial bus interface control register (SBIC).

(a) Serial operating mode register 0 (CSIM0)

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input sets CSIM0 to 00H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	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 $\overline{SCK0}$ 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	$\overline{SCK0}$/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)	$\overline{SCK0}$ (CMOS input/output)
		1								LSB			
1	0	SBI mode (Refer to **13.4.3 SBI mode operation**.)											
1	1	2-wire serial I/O mode (Refer to **13.4.4 2-wire serial I/O mode operation**.)											

R/W

WUP	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×× : output latch of port

(b) Serial bus interface control register (SBIC)

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

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

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

R/W		
	RELT	When RELT = 1, SO latch is set to 1. After SO latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W		
	CMDT	When CMDT = 1, SO latch is cleared to 0. After SO latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

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

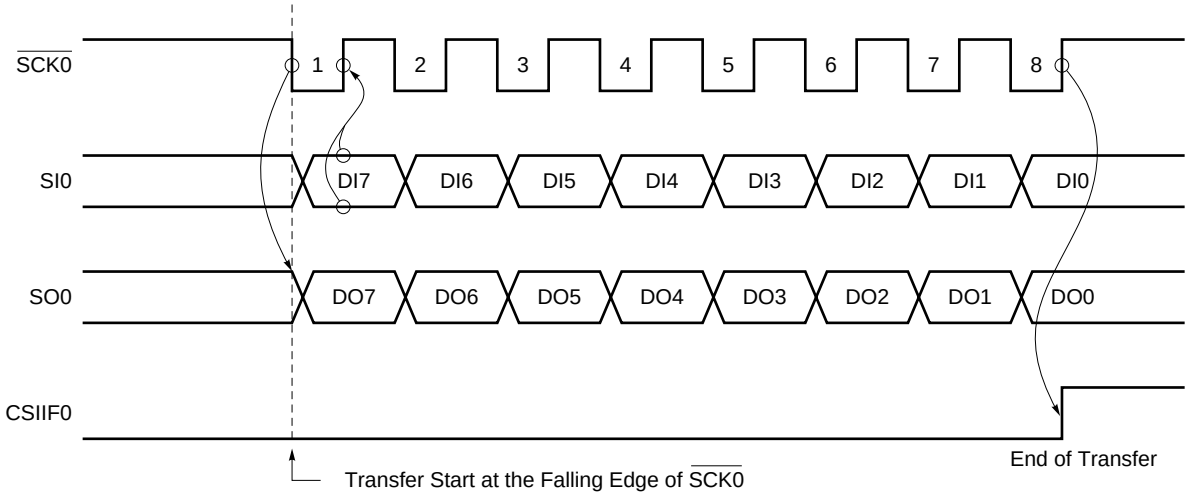
(2) Communication operation

The 3-wire serial I/O mode is used for data transmission/reception in 8-bit units. Bit-wise data transmission/reception is carried out in synchronization with the serial clock.

Shift operation of the serial I/O shift register 0 (SIO0) is carried out at the falling edge of the serial clock ($\overline{SCK0}$). The transmitted data is held in the SO0 latch and is output from the SO0 pin. The received data input to the SI0 pin is latched in SIO0 at the rising edge of $\overline{SCK0}$.

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

Figure 13-7. 3-Wire Serial I/O Mode Timings



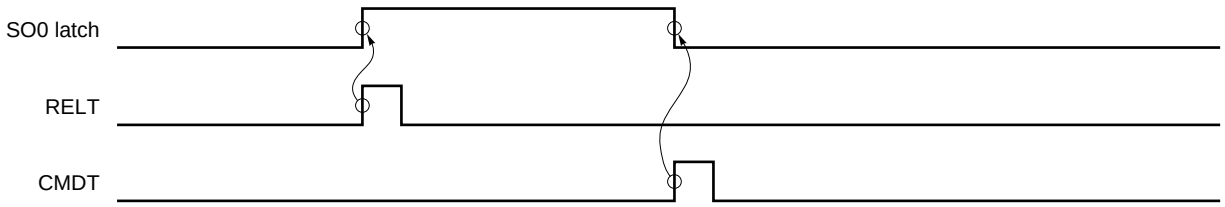
The SO0 pin is a CMOS output pin and outputs current SO0 latch statuses. Thus, the SO0 pin output status can be manipulated by setting the bit 0 (RELT) and bit 1 (CMDT) of the serial bus interface control register (SBIC). However, do not carry out this manipulation during serial transfer.

Control the $\overline{SCK0}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **13.4.5 $\overline{SCK0}$/P27 pin output manipulation**).

(3) Other signals

Figure 13-8 shows RELT and CMDT operations.

Figure 13-8. RELT and CMDT Operations



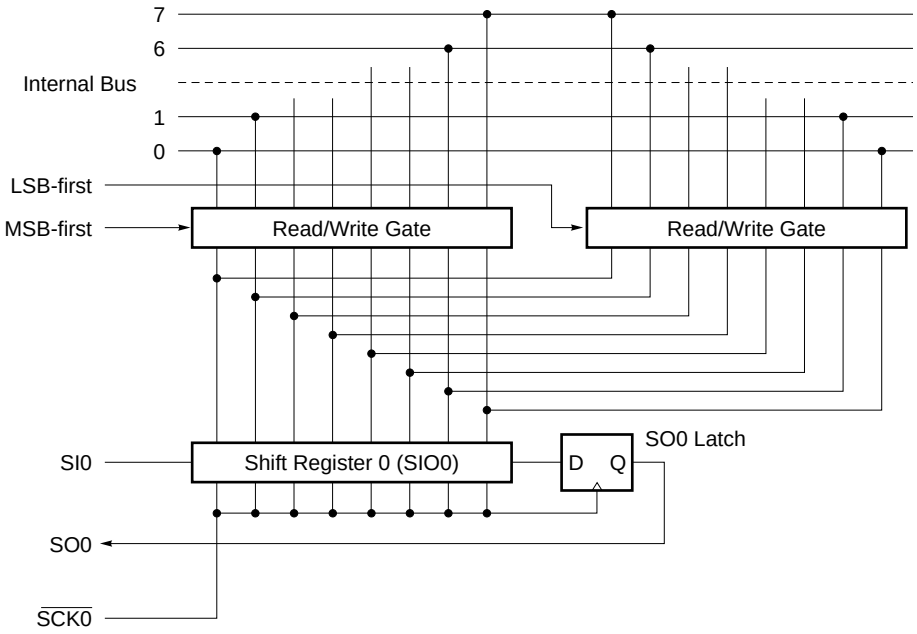
(4) MSB/LSB switching as the start bit

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

Figure 13-9 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 13-9. 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 the shift register.

(5) Transfer start

Serial transfer is started by setting transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied.

- Serial interface channel 0 operation control bit (CSIE0) = 1.
- Internal serial clock is stopped or $\overline{\text{SCK0}}$ is a high level after 8-bit serial transfer.

Caution If CSIE0 is set to "1" after data write to SIO0, transfer does not start.

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

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

13.4.3 SBI mode operation

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

SBI uses a single master device and employs the clocked serial I/O format with the addition of a bus configuration function. This function enables devices to communicate using only two lines. Thus, when making up a serial bus with two or more microcontrollers and peripheral ICs, the number of ports to be used and the number of wires on the board can be decreased.

The master device outputs three kinds of data to slave devices on the serial data bus: “addresses” to select a device to be communicated with, “commands” to instruct the selected device, and “data” which is actually required.

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

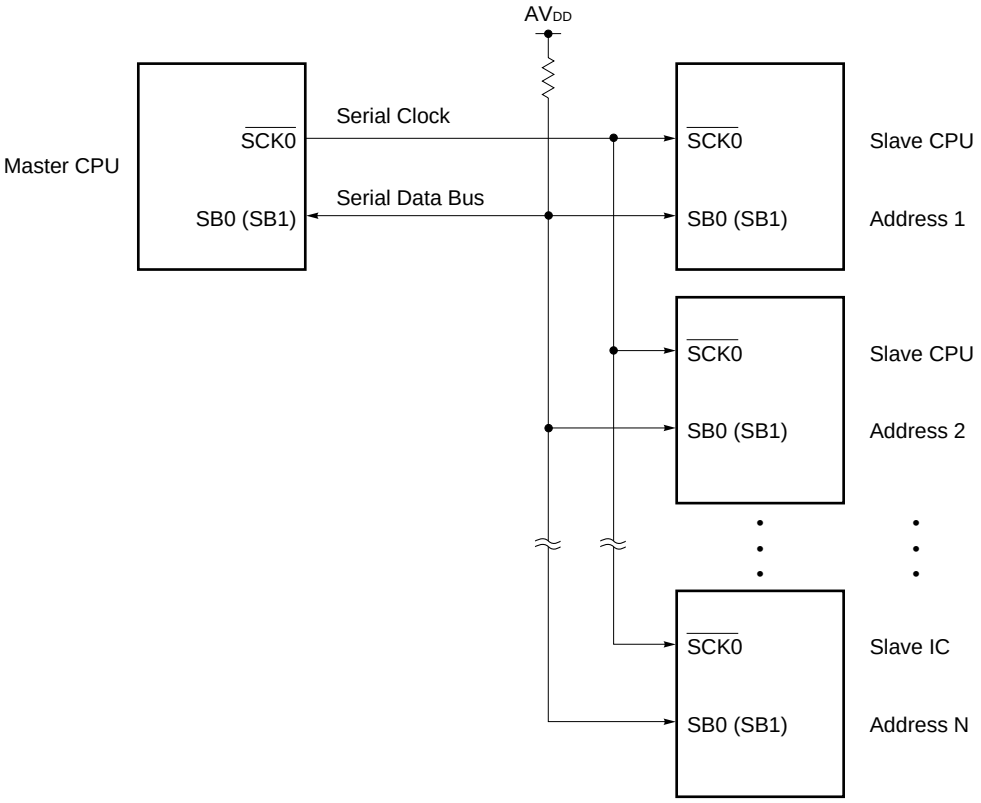
The SBI function is incorporated into various devices including 75X/XL series and 78K series devices.

Figure 13-10 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 13-10. 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 ($\overline{\text{SKC0}}$) input/output switching is carried out asynchronously between the master and slave CPUs.

(1) SBI functions

In the conventional serial I/O format, when a serial bus is configured by connecting two or more devices, many ports and wiring are necessary, to provide chip select signal to identify command and data, and to judge the busy state, because only the data transfer function is available. If these operations are to be controlled by software, the software must be heavily loaded.

In SBI, a serial bus can be configured with two signal lines of serial clock $\overline{\text{SCK0}}$ and serial data bus SB0 (SB1). Thus, use of SBI leads to reduction in the number of microcontroller ports and that of wirings and routings on the board.

The SBI functions are described below.

(a) Address/command/data identify function

Serial data is distinguished into addresses, commands, and data.

(b) Chip select function by address transmission

The master executes slave chip selection by address transmission.

(c) Wake-up function

The slave can easily judge address reception (chip select judgment) with the wake-up function (which can be set/reset by software).

When the wake-up function is set, the interrupt request signal (INTCSI0) is generated upon reception of a match address.

Thus, when communication is executed with two or more devices, the CPU except the selected slave devices can operate regardless of underway serial communications.

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

The acknowledge signal to check serial data reception is controlled.

(e) Busy signal ($\overline{\text{BUSY}}$) control function

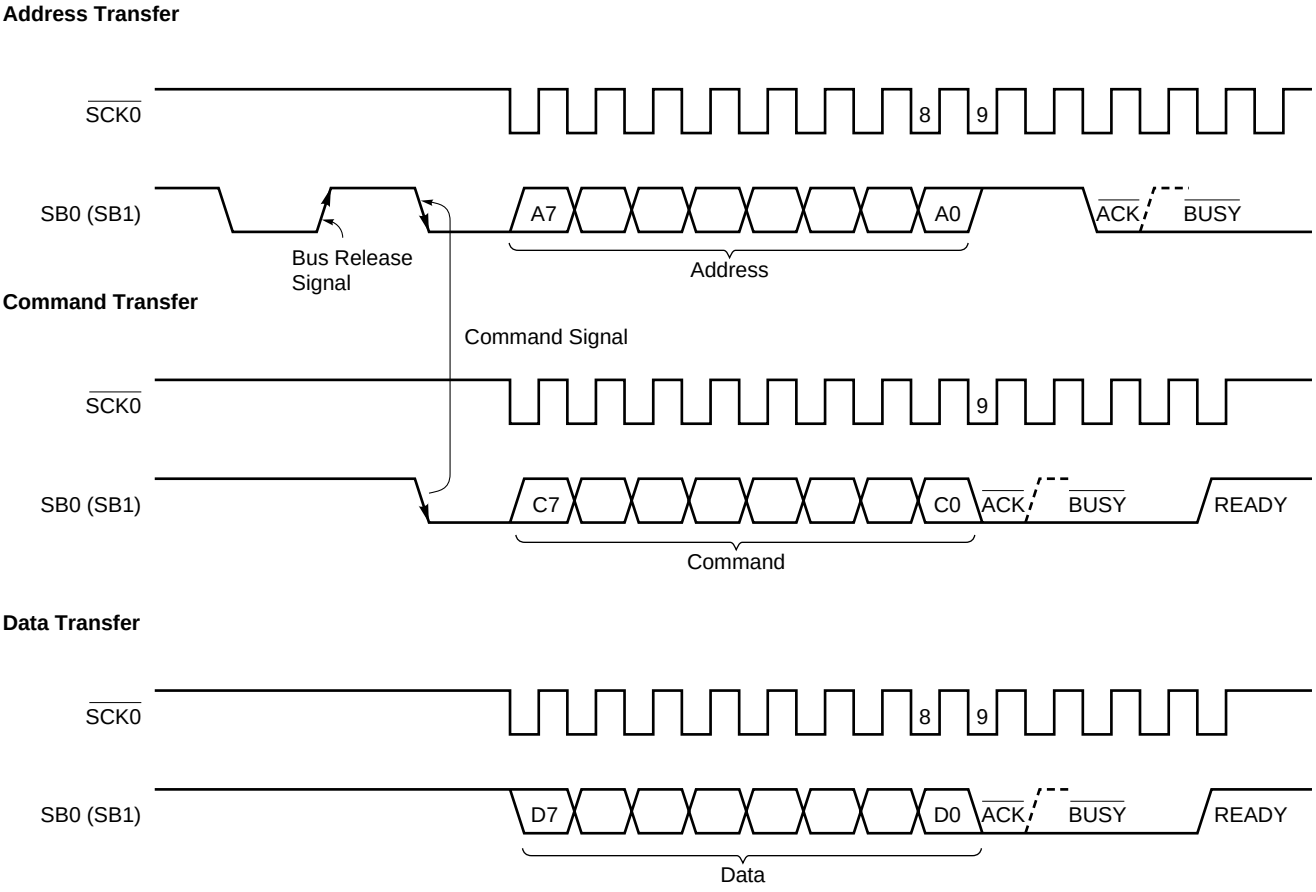
The busy signal to report the slave busy state is controlled.

(2) SBI definition

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

Serial data to be transferred with SBI consists of three kinds of data: “address”, “command”, and “data”. Figure 13-11 shows the address, command, and data transfer timings.

Figure 13-11. SBI Transfer Timings



Remark The dotted line indicates the READY status.

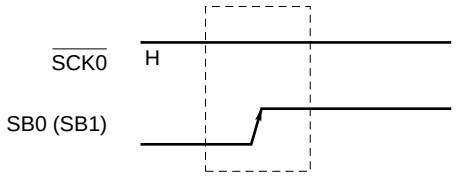
The bus release signal and the command signal are output by the master device. $\overline{\text{BUSY}}$ is output by the slave signal. $\overline{\text{ACK}}$ can be output by either the master or slave device (normally, the 8-bit data receiver outputs). Serial clocks continue to be output by the master device from 8-bit data transfer start to $\overline{\text{BUSY}}$ reset.

(a) Bus release signal (REL)

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

This signal is output by the master device.

Figure 13-12. Bus Release Signal



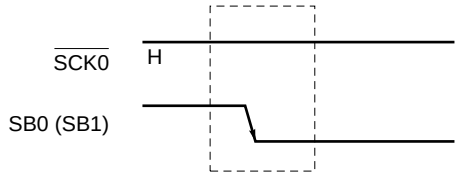
The bus release signal indicates that the master device is going to transmit an address to the slave device. The slave device incorporates hardware to detect the bus release signal.

★ **Caution The positive transition of the SB0 (SB1) line (transition from low to high) while the $\overline{SCK0}$ line is high is recognized as a bus release signal. If the timing of changes on the bus shifts due to the influence of board capacitance, etc., a bus release signal may be detected even while data is being transmitted. Exercise care when routing the wiring.**

(b) Command signal (CMD)

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

Figure 13-13. Command Signal



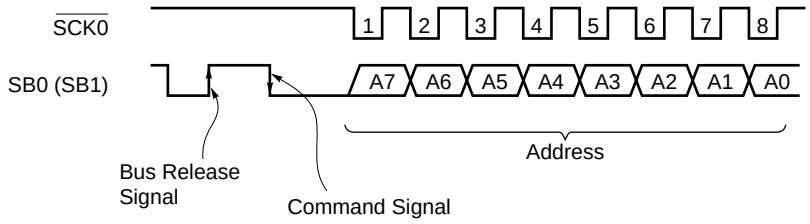
The command signal indicates that the master is going to transmit a command to the slave (however, the command signal following the bus release signal indicates that an address is to be transmitted). The slave device incorporates hardware to detect the command signal.

★ **Caution The positive transition of the SB0 (SB1) line (transition from low to high) while the $\overline{SCK0}$ line is high is recognized as a command signal. If the timing of changes on the bus shifts due to the influence of board capacitance, etc., a command signal may be detected even while data is being transmitted. Exercise care when routing the wiring.**

(c) Address

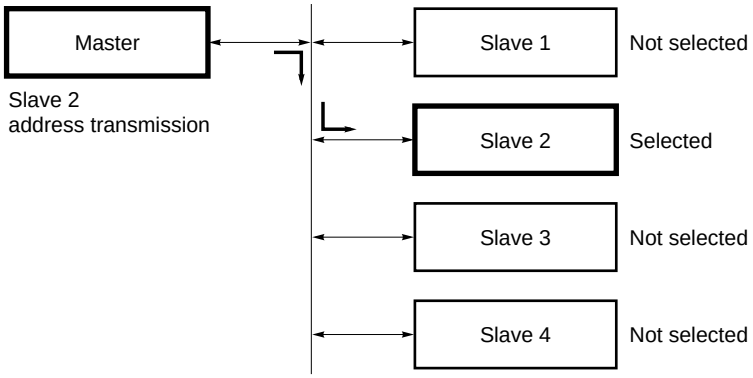
An address is 8-bit data which the master device outputs to the slave device connected to the bus line in order to select a particular slave device.

Figure 13-14. 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 13-15. 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 13-16. Commands

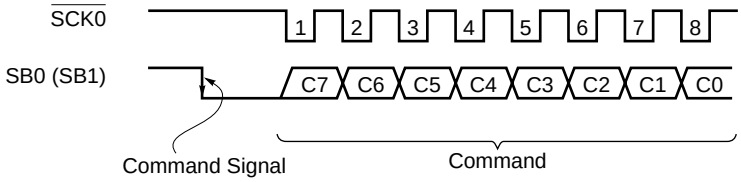
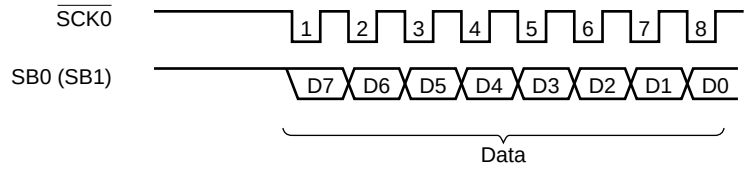


Figure 13-17. Data



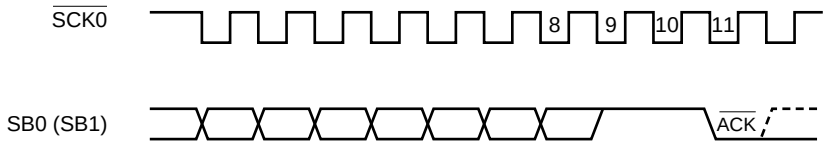
8-bit data following a command signal is defined as “command” data. 8-bit data without command signal is defined as “data”. Command and data operation procedures are allowed to determine by user according to communications specifications.

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

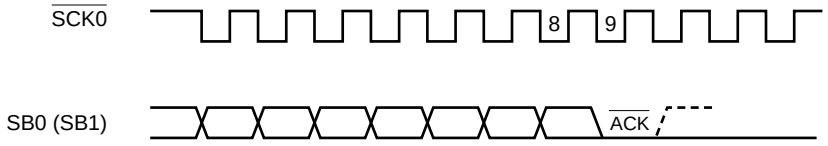
The acknowledge signal is used to check serial data reception between transmitter and receiver.

Figure 13-18. Acknowledge Signal

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



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



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

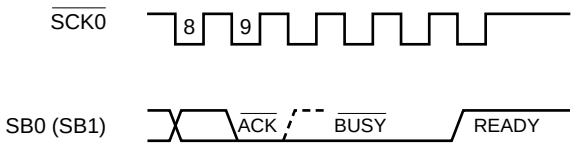
After 8-bit data transmission, the transmitter checks whether the receiver has returned the acknowledge signal. If the acknowledge signal is not returned for the preset period of time after data transmission, it can be judged that data reception has not been carried out correctly.

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

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

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

Figure 13-19. $\overline{\text{BUSY}}$ and READY Signals



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

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

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

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

(3) Register setting

The SBI mode is set with the serial operating mode register 0 (CSIM0), the serial bus interface control register (SBIC), and the interrupt timing specify register (SINT).

(a) Serial operating mode register 0 (CSIM0)

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM0 to 00H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W Note 1

R/W	CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection
	0	×	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 (refer to **13.4.2 3-wire serial I/O mode operation**.)											
	1	0	0	Note 2 ×	Note 2 ×	0	0	0	1	SBI mode	MSB	P25 (CMOS 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 (refer to **13.4.4 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	COI	Slave Address Comparison Result Flag Note 4
	0	Slave address register not equal to serial I/O shift register 0 data
	1	Slave address register equal to serial I/O shift register 0 data

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

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

2. Can be used as a port.

3. To use the wake-up function (WUP = 1), clear bit 5 (SIC) of the interrupt timing specification register (SINT) to 0.

4. When CSIE0=0, COI becomes 0.

Remark × : don't care

PM×× : port mode register

P×× : output latch of port

(b) Serial bus interface control register (SBIC)

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

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

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

R/W

RELT	Used for bus release signal output. When RELT = 1, SO latch is set to (1). After SO latch setting, automatically cleared to (0). Also cleared to 0 when CSIE0 = 0.

R/W

CMDT	Used for command signal output. When CMDT = 1, SO latch is cleared to (0). After SO latch clearance, automatically cleared to (0). Also cleared to 0 when CSIE0 = 0.

R

RELD	Bus Release Detection	
Clear Conditions (RELD = 0)		Set Conditions (RELD = 1)
• When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception • When $\overline{\text{CSIE0}}$ = 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 $\overline{\text{CSIE0}}$ = 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 clock of $\overline{\text{SCK0}}$ just after execution of the instruction to be set to (1) and, after acknowledge signal output, automatically cleared to (0). Used as ACKE=0. Also cleared to (0) upon start of serial interface transfer or when CSIE0 = 0.

R/W

ACKE		
0	Acknowledge signal automatic output disable (output with ACKT enable)	
1	Before completion of transfer	Acknowledge signal is output in synchronization with the 9th clock falling edge of $\overline{\text{SCK0}}$ (automatically output when ACKE = 1).
	After completion of transfer	Acknowledge signal is output in synchronization with falling edge clock of $\overline{\text{SCK0}}$ just after execution of the instruction to be set to 1 (automatically output when ACKE = 1). However, not automatically cleared to 0 after acknowledge signal output.

(Continued)

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

Remarks 1. Bits 0, 1, and 4 (RELT, CMDT, and ACKT) are 0 when read after data setting.

2. CSIE0: Bit 7 of the serial operation mode register 0 (CSIM0)

R

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

R/W

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

Note Busy mode can be cleared by start of serial interface transfer. However, BSYE flag is not cleared to 0.

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

(c) Interrupt timing specify register (SINT)

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

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

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

R/W

SVAM	SVA Bit to be Used as Slave Address
0	Bits 0 to 7
1	Bits 1 to 7

R/W

SIC	INTCSI0 Interrupt Factor Selection[Note 2]
0	CSIIF0 is set upon termination of serial interface channel 0 transfer
1	CSIIF0 is set upon bus release detection or termination of serial interface channel 0 transfer

R

CLD	$\overline{\text{SCK0}}$ Pin Level[Note 3]
0	Low level
1	High level

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

2. When using wake-up function in the SBI mode, set SIC to 0.

3. When CSIE0 = 0, CLD becomes 0.

Caution Be sure to set bits 0 to 3 to 0.

Remark SVA : Slave address register

CSIIF0: Interrupt request flag corresponding to INTCSI0

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

(4) Various signals

Figures 13-20 to 13-25 show various signals and flag operations in SBI. Table 13-3 lists various signals in SBI.

Figure 13-20. RELT, CMDT, RELD, and CMDD Operations (Master)

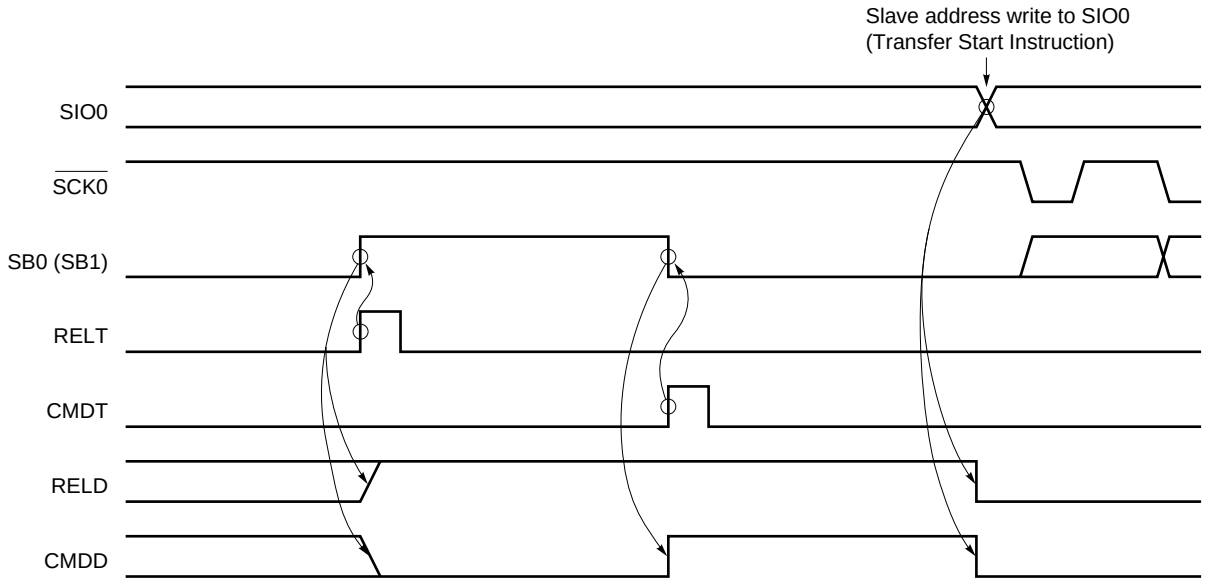


Figure 13-21. RELD and CMDD Operations (Slave)

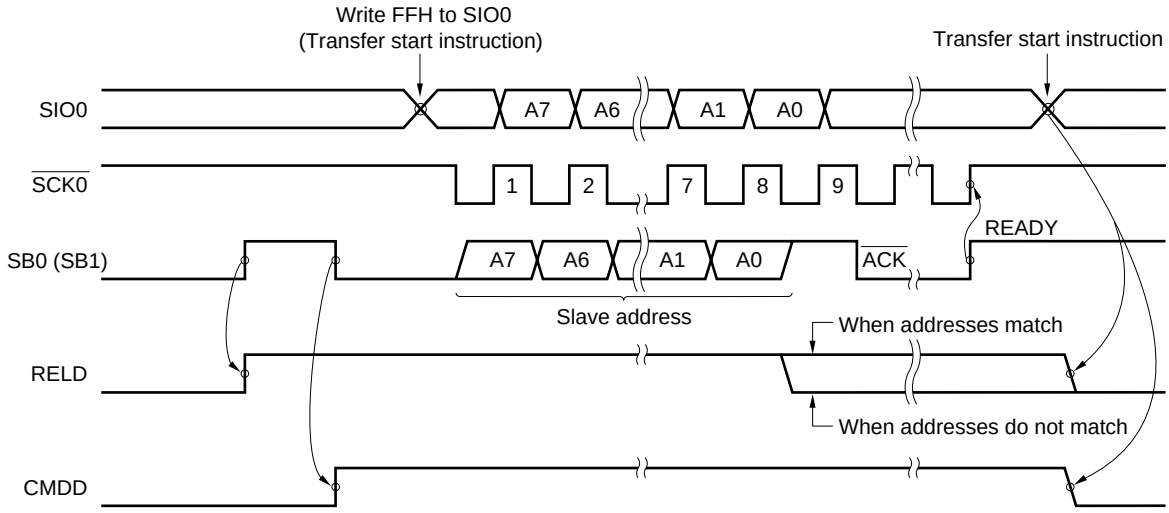
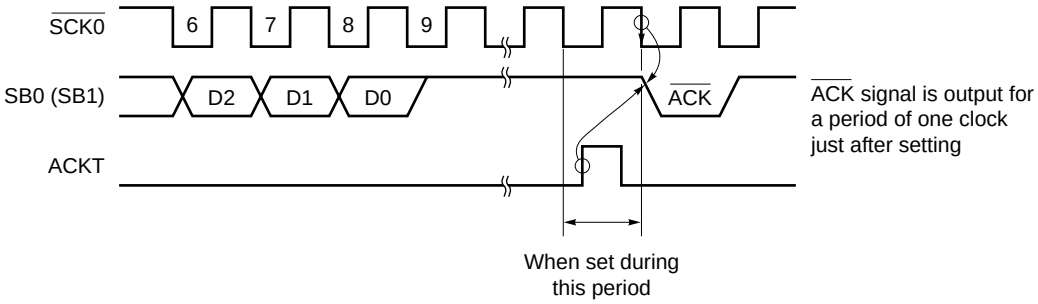


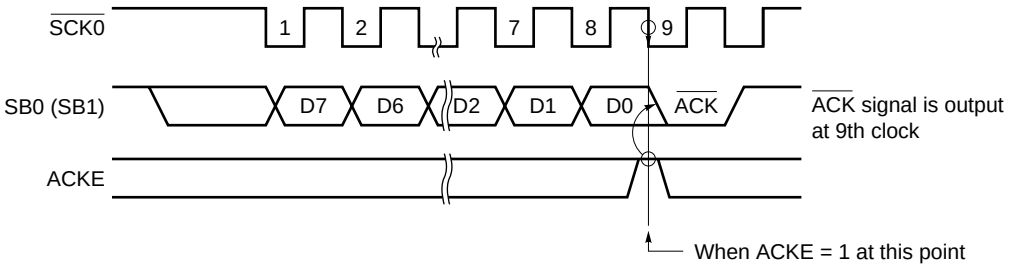
Figure 13-22. ACKT Operations



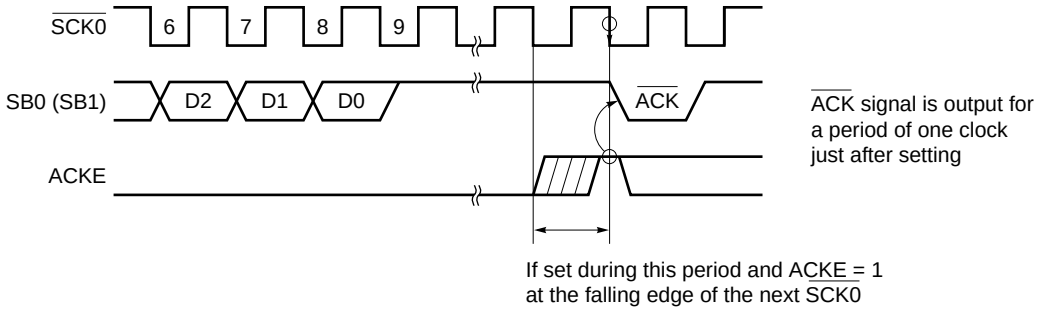
Caution Do not set ACKT before termination of transfer.

Figure 13-23. ACKE Operations

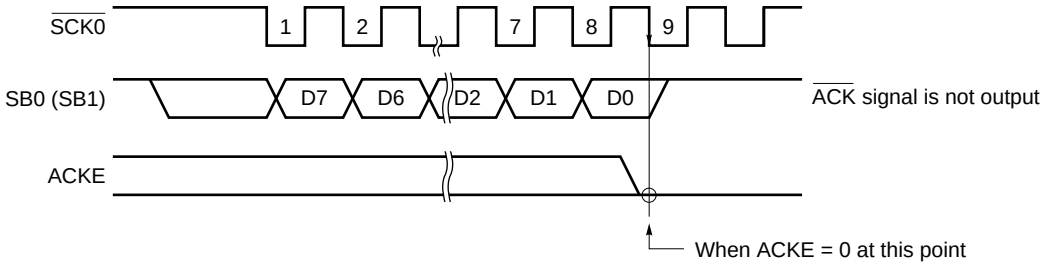
(a) When ACKE = 1 upon completion of transfer



(b) When set after completion of transfer



(c) When ACKE = 0 upon completion of transfer



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

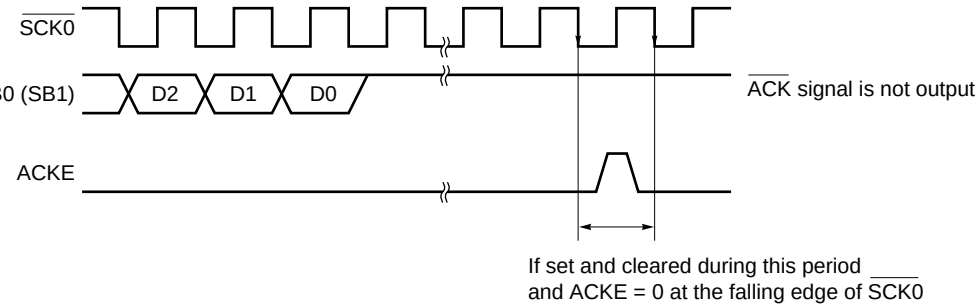
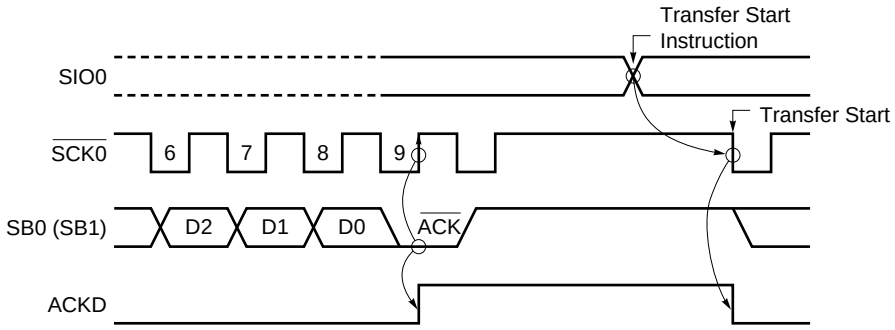
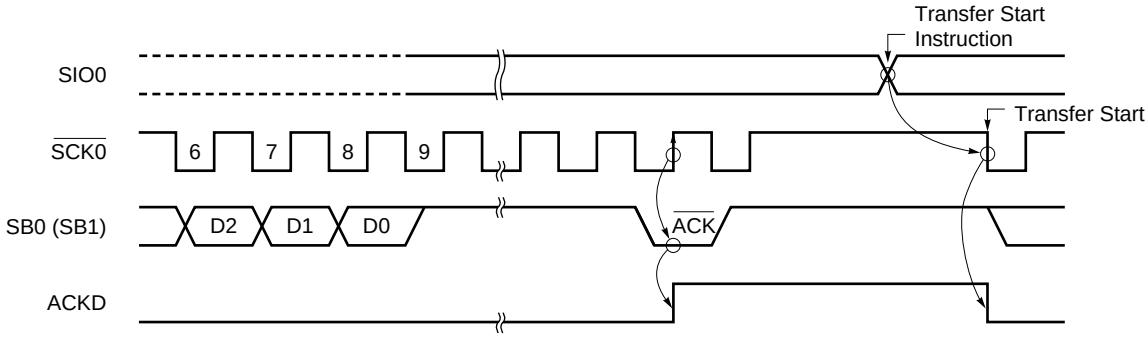


Figure 13-24. ACKD Operations

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



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



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

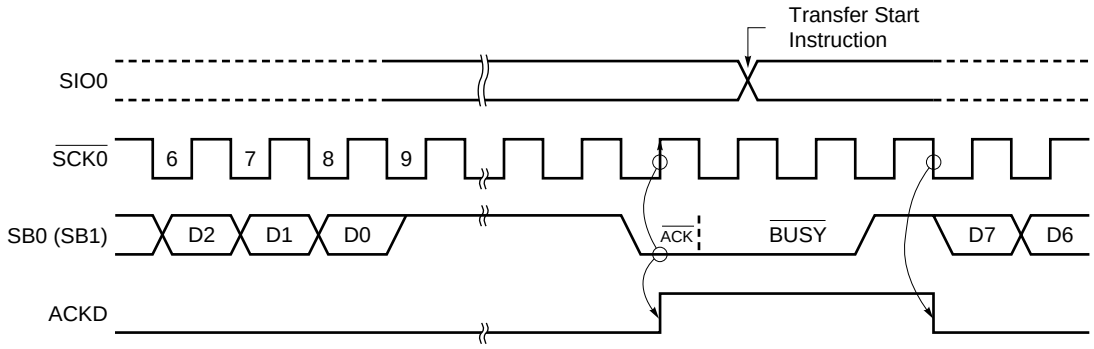


Figure 13-25. BSYE Operation

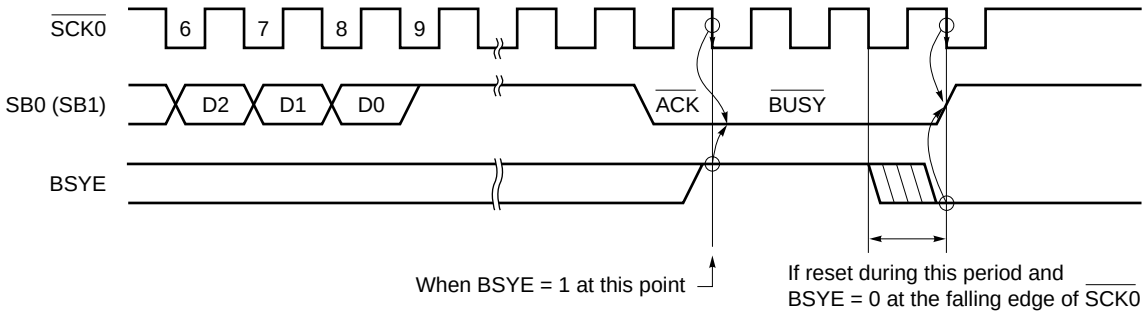


Table 13-3. Various Signals in SBI Mode (1/2)

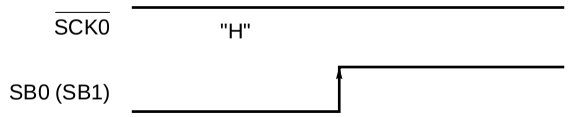
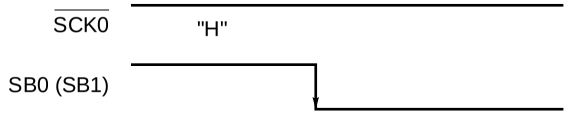
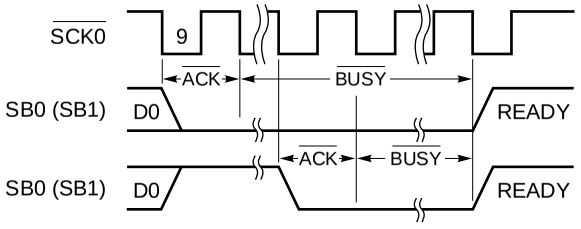
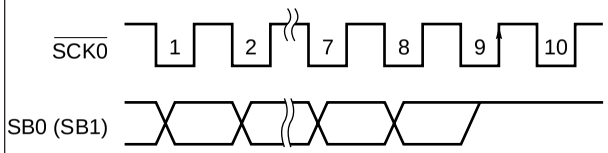
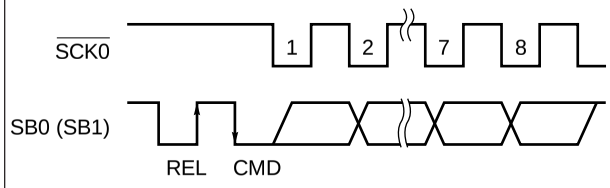
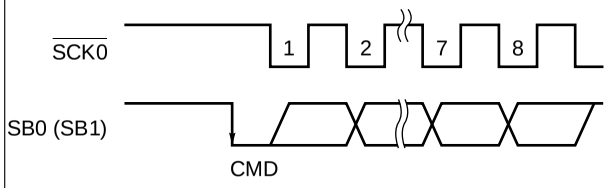
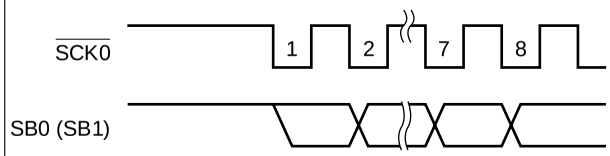
Signal Name	Output Device	Definition	Timing Chart	Output Condition	Effects on Flag	Meaning of Signal
Bus release signal (REL)	Master	SB0 (SB1) rising edge when $\overline{SCK0}$ = 1		• RELT set	• RELD set • CMDD clear	CMD signal is output to indicate that transmit data is an address.
Command signal (CMD)	Master	SB0 (SB1) falling edge when $\overline{SCK0}$ = 1		• CMDT set	• CMDD set	i) Transmit data is an address after REL signal output. ii) REL signal is not output and transmit data is an command.
Acknowledge signal ($\overline{ACK}$)	Master/ slave	Low-level signal to be output to SB0 (SB1) during one-clock period of $\overline{SCK0}$ after completion of serial reception	[Synchronous BUSY output]	(1) ACKE = 1 (2) ACKT set	• ACKD set	Completion of reception
Busy signal ($\overline{BUSY}$)	Slave	[Synchronous BUSY signal] Low-level signal to be output to SB0 (SB1) following Acknowledge signal		• BSYE = 1	—	Serial receive disable because of processing
Ready signal (READY)	Slave	High-level signal to be output to SB0 (SB1) before serial transfer start and after completion of serial transfer		(1) BSYE = 0 (2) Execution of instruction for data write to SIO0 (transfer start instruction)	—	Serial receive enable

Table 13-3. Various Signals in SBI Mode (2/2)

Signal Name	Output Device	Definition	Timing Chart	Output Condition	Effects on Flag	Meaning of Signal
Serial clock ($\overline{SCK0}$)	Master	Synchronous clock to output address/command/ data, $\overline{ACK}$ signal, synchro- nous $\overline{BUSY}$ signal, etc. Address/command/data are transferred with the first eight synchronous clocks.		When CSIE0 = 1, execution of instruction for data write to SIO0 (serial transfer start instruction)**Note 2**	CSIIF0 set (rising edge of 9th clock of $\overline{SCK0}$)**Note 1**	Timing of signal output to serial data bus
Address (A7 to A0)	Master	8-bit data to be transferred in synchronization with $\overline{SCK0}$ after output of REL and CMD signals				Address value of slave device on the serial bus
Commands (C7 to C0)	Master	8-bit data to be transferred in synchronization with $\overline{SCK0}$ after output of only CMD signal without REL signal output				Instructions and messages to the slave device
Data (D7 to D0)	Master/ slave	8-bit data to be transferred in synchronization with $\overline{SCK0}$ without output of REL and CMD signals				Numeric values to be processed with slave or master device

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

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

(5) Pin configuration

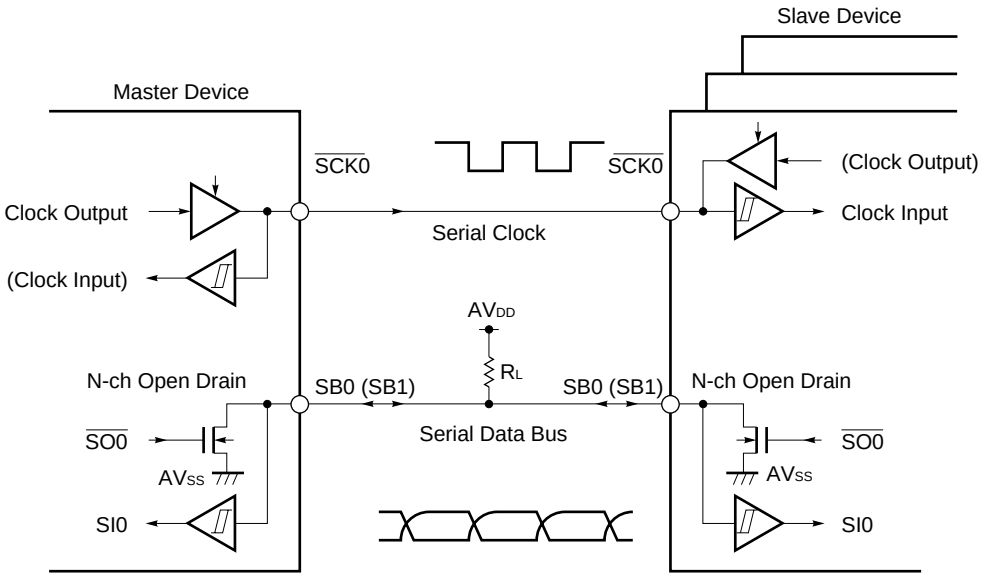
The serial clock pin $\overline{SCK0}$ and serial data bus pin SB0 (SB1) have the following configurations.

(a) $\overline{SCK0}$ ..... Serial clock input/output pin
<1> Master ... CMOS and push-pull output
<2> Slave..... Schmitt input

(b) SB0 (SB1) .... Serial data input/output dual-function pin
Both master and slave devices have an N-ch open drain output and a Schmitt input.

Because the serial data bus line has an N-ch open-drain output, an external pull-up resistor is necessary.

Figure 13-26. Pin Configuration



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

(6) Address match detection method

In the SBI mode, a particular slave device is selected by slave address transmission by the master device. Address coincidence is automatically detected by hardware. When the wake-up function specification bit (WUP) = 1, CSIIF0 is set if the slave address transmitted from the master coincides with the address set to the slave address register (SVA).

While bit 5 (SIC) of the interrupt timing specification register (SINT) is set (1), the wake-up function does not operate even if WUP is set to 1 (instead, an interrupt request signal is generated on detection of bus release). Clear SIC to 0 when the wake-up function is used.

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 13-27 to 13-30 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.

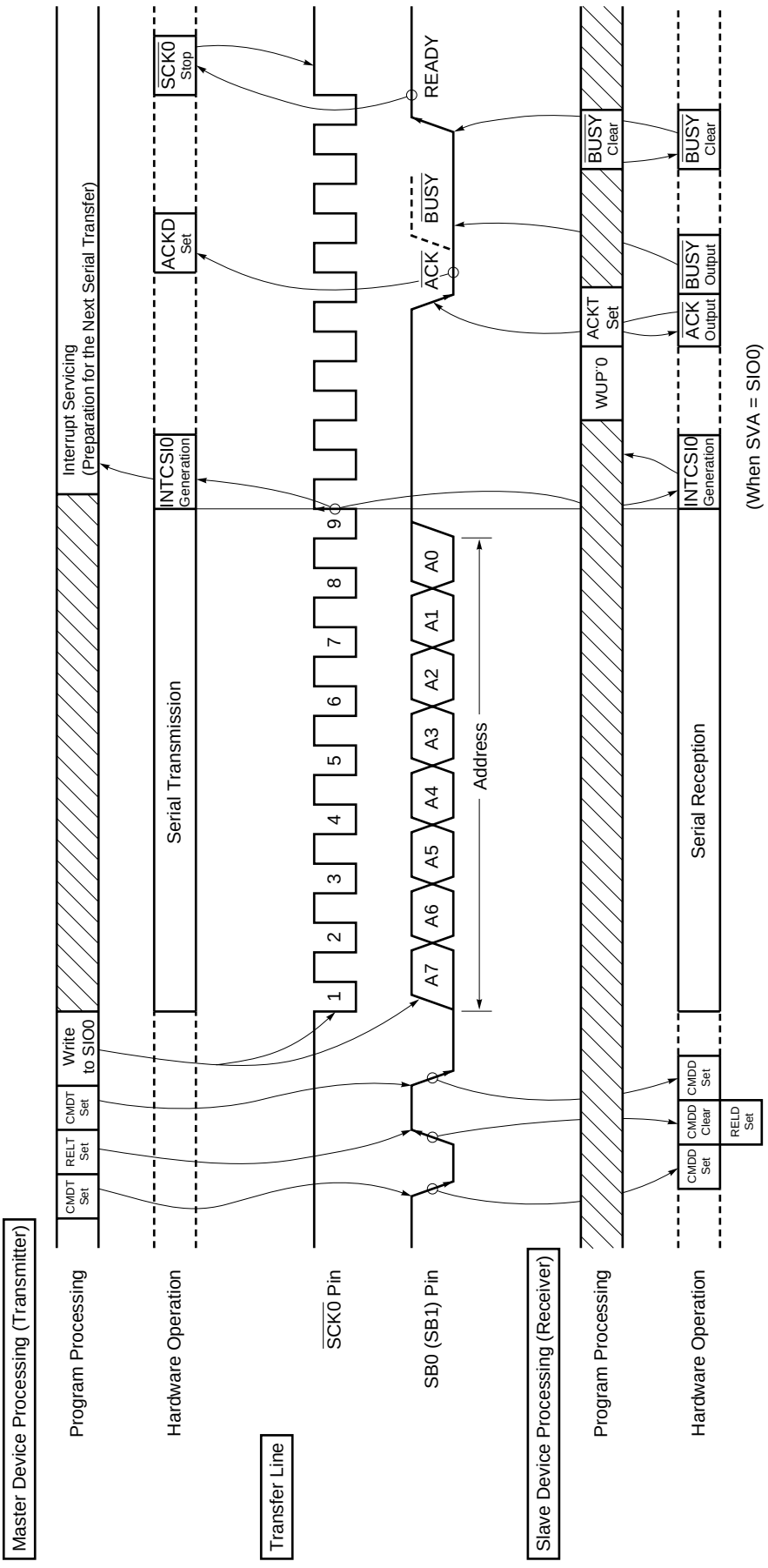
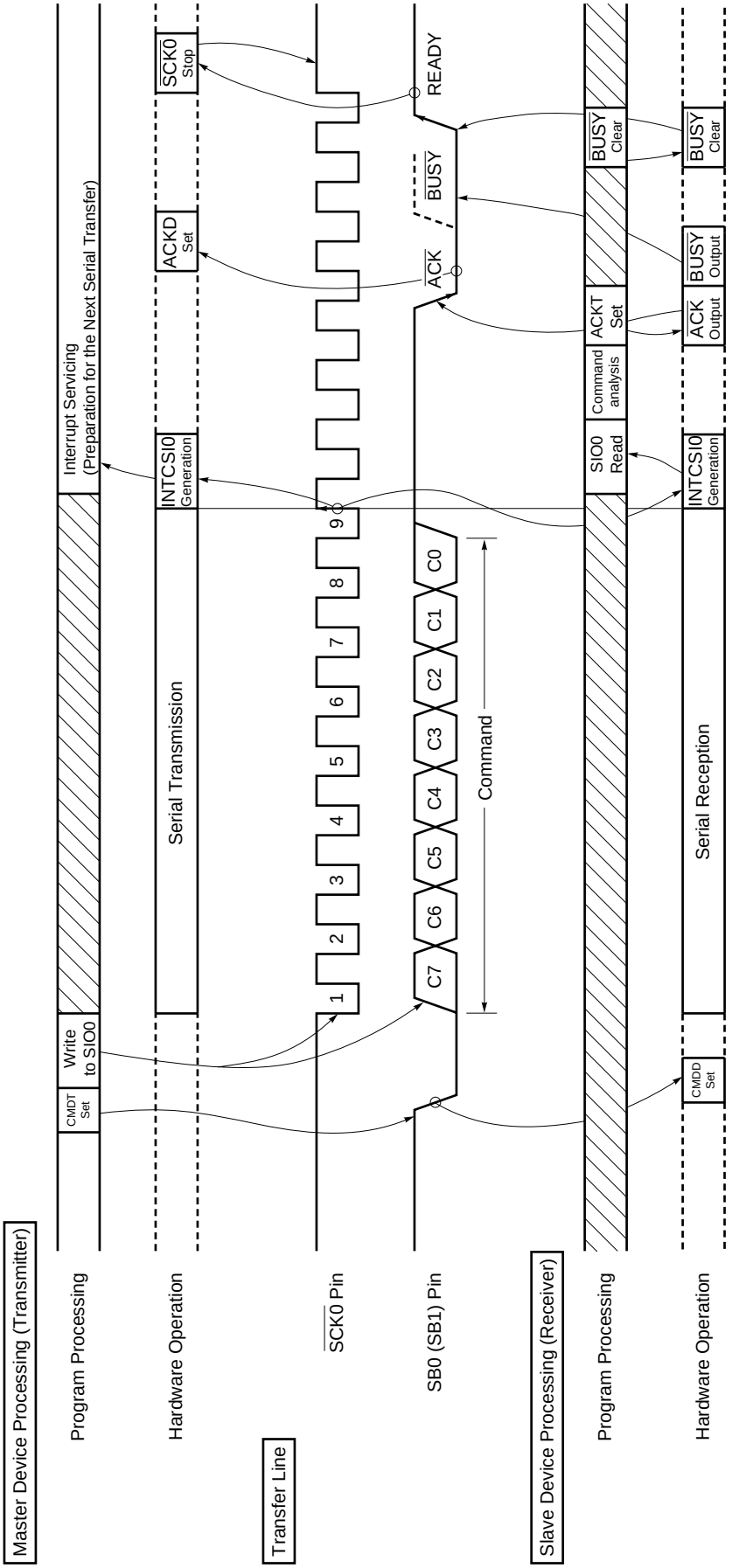


Figure 13-27. Address Transmission from Master Device to Slave Device (WUP = 1)

Figure 13-28. Command Transmission from Master Device to Slave Device



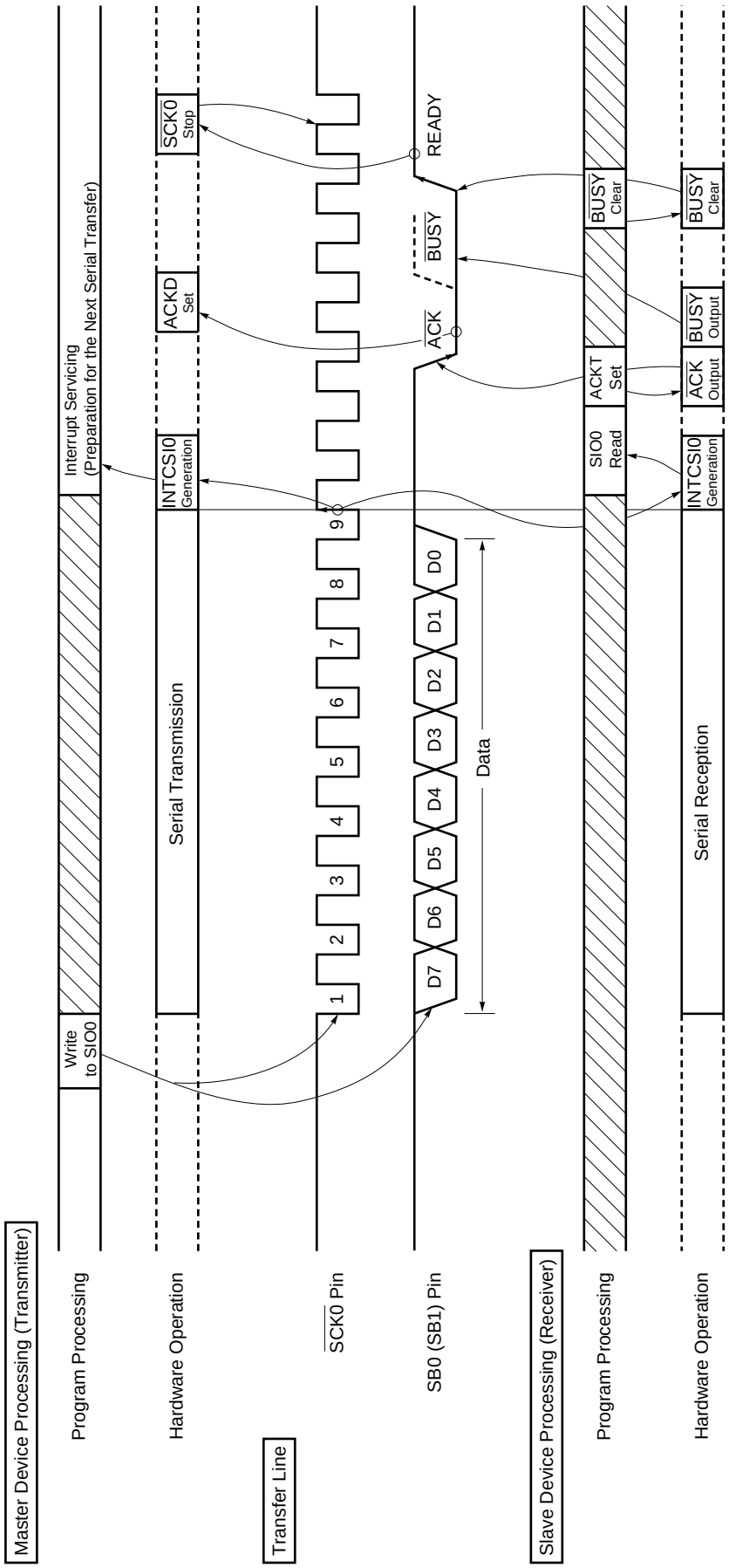


Figure 13-29. Data Transmission from Master Device to Slave Device

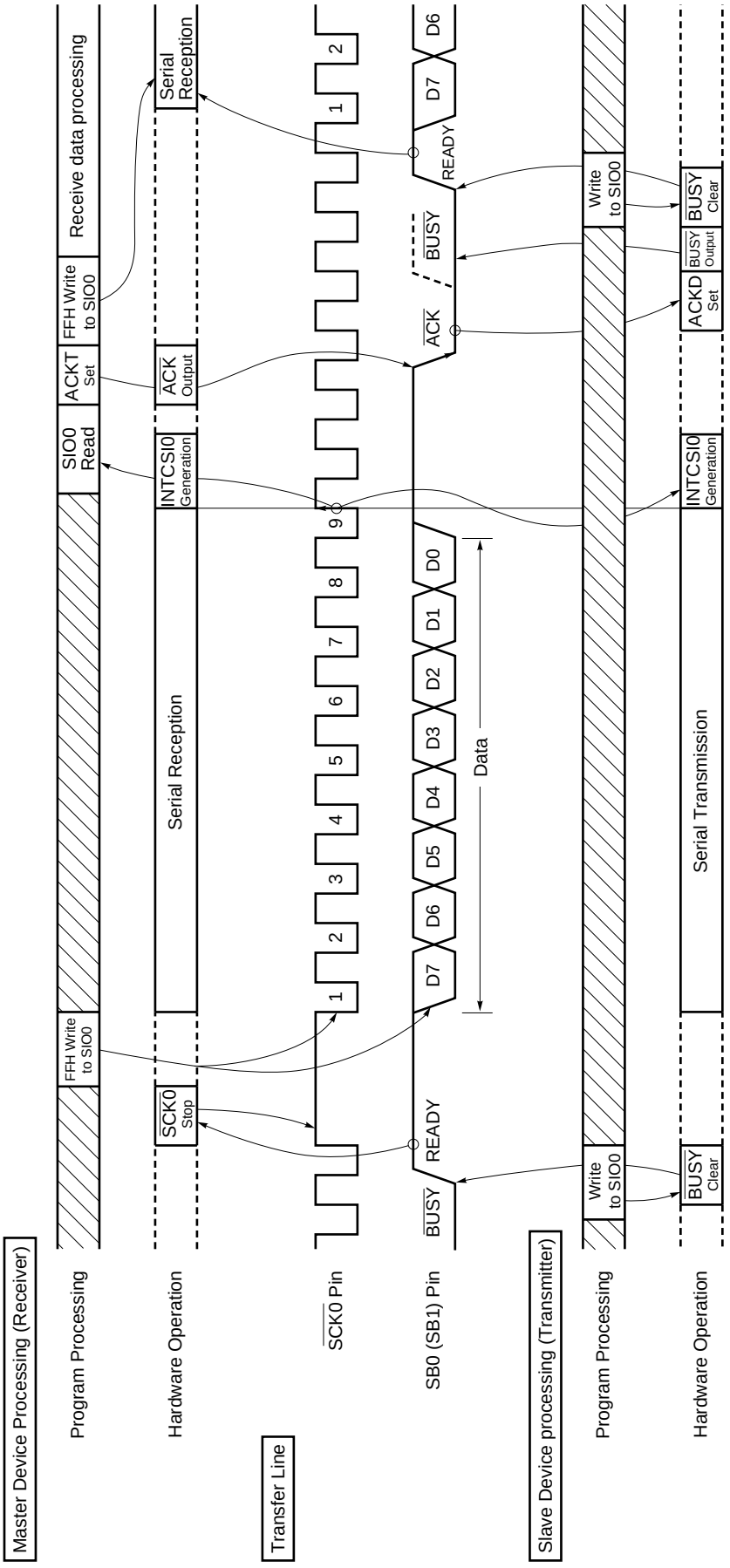


Figure 13-30. Data Transmission from Slave Device to Master Device

(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 $\overline{SCK0}$ is at high level after 8-bit serial transfer.

Cautions 1. If CSIE0 is set to "1" after data write to SIO0, transfer does not start.

2. Because the N-ch open-drain output must be made high-impedance state for data reception, write FFH to SIO0 in advance.
However, when the make-up function specify bit (WUP) = 1, the N-ch open-drain output is always at high-impedance state. Thus, it is not necessary to write FFH to SIO0.

3. If data is written to SIO0 when the slave is busy, the data is not lost.
When the busy state is cleared and SB0 (or SB1) input is set to the high level (READY) state, transfer starts.

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

★ Be sure to perform the following setting to the pin (SB0 or SB1) that is used to input/output data before serial transfer of 1 byte after the $\overline{RESET}$ signal has been input:

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

(10) Method to judge busy state of a slave

Check whether a slave is in the busy status from the device in the master mode, in the following procedure:

<1> Detect generation of the acknowledge signal ($\overline{ACK}$) or interrupt request signal.
<2> Set the port mode register PM25 (or PM26) of the SB0/P25 (or SB1/P26) pin in the input mode.
<3> Read the status of the pin (if the pin is high, it is in the ready status).

After detecting the ready status, set 0 to the port mode register, to restore 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) SBI outputs the $\overline{\text{BUSY}}$ signal after the $\overline{\text{BUSY}}$ has been cleared and until the next serial clock falls. If WUP is set to 1 by mistake during this period, $\overline{\text{BUSY}}$ will not be cleared. To set WUP to 1, therefore, clear $\overline{\text{BUSY}}$, and make sure that the SB0 (SB1) pin has gone high.

(d) For pins which are to be used for data input/output, be sure to carry out the following settings before serial transfer of the 1st byte after $\overline{\text{RESET}}$ 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.

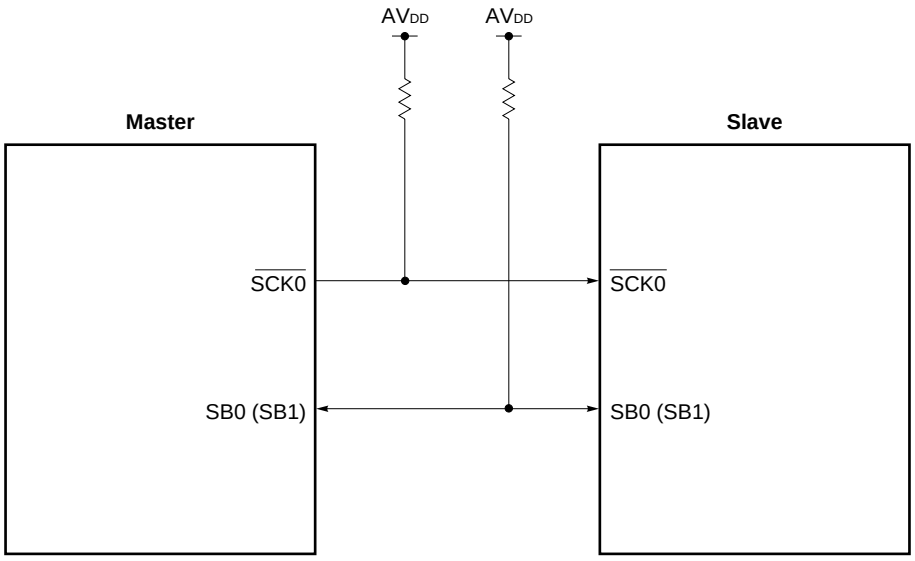
★ (e) The positive transition of the SB0 (SB1) line (transition from low to high or high to low) while the $\overline{\text{SCK}}$ line is high is recognized as a bus release signal or a command signal. If the timing of changes on the bus shifts due to the influence of board capacitance, etc., a bus release signal (or a command signal) may detected even while data is being transmitted. Exercise care when routing the wiring.

13.4.4 2-wire serial I/O mode operation

The 2-wire serial I/O mode can cope with any communication format by program.

Communication is basically carried out with two lines of serial clock ($\overline{SCK0}$) and serial data input/output (SB0 or SB1).

Figure 13-31. Serial Bus Configuration Example Using 2-Wire Serial I/O Mode



(1) Register setting

The 2-wire serial I/O mode is set with the serial operating mode register 0 (CSIM0), the serial bus interface control register (SBIC), and the interrupt timing specify register (SINT).

(a) Serial operating mode register 0 (CSIM0)

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM0 to 00H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W Note 1

R/W

CSIM01	CSIM00	Serial Interface Channel 0 Clock Selection
0	×	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 (Refer to **13.4.2 3-wire serial I/O mode operation**)											
1	0	SBI mode (Refer to **13.4.3 SBI mode operation**)											
1	1	0	Note 2 ×	Note 2 ×	0	0	0	1	2-wire serial I/O mode	MSB	P25 (CMOS 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 data in SBI mode

R

COI	Slave Address Comparison Result Flag Note4
0	Slave address register not equal to serial I/O shift register 0 data
1	Slave address register equal to serial I/O shift register 0 data

R/W

CSIE0	Serial Interface Channel 0 Operation Control
0	Operation stopped
1	Operation enabled

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

2. Can be used freely as port function.

3. Be sure to set WUP to 0 when the 2-wire serial I/O mode.

4. When CSIE0=0, COI becomes 0.

Remark × : don't care

PM×× : port mode register

P×× : output latch of port

(b) Serial bus interface control register (SBIC)

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

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

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

R/W

RELT	When RELT = 1, SO latch is set to 1. After SO latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W

CMDT	When CMDT = 1, SO latch is cleared to 0. After SO latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

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

(c) Interrupt timing specify register (SINT)

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

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

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

R/W

SIC	INTCSI0 Interrupt Factor Selection
0	CSIIF0 is set upon termination of serial interface channel 0 transfer
1	CSIIF0 is set upon bus release detection or termination of serial interface channel 0 transfer

R

CLD	$\overline{\text{SCK0}}$ 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 bit 0 to bit 3 to 0.

Remark CSIIF0: Interrupt request flag corresponding to INTCSI0

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

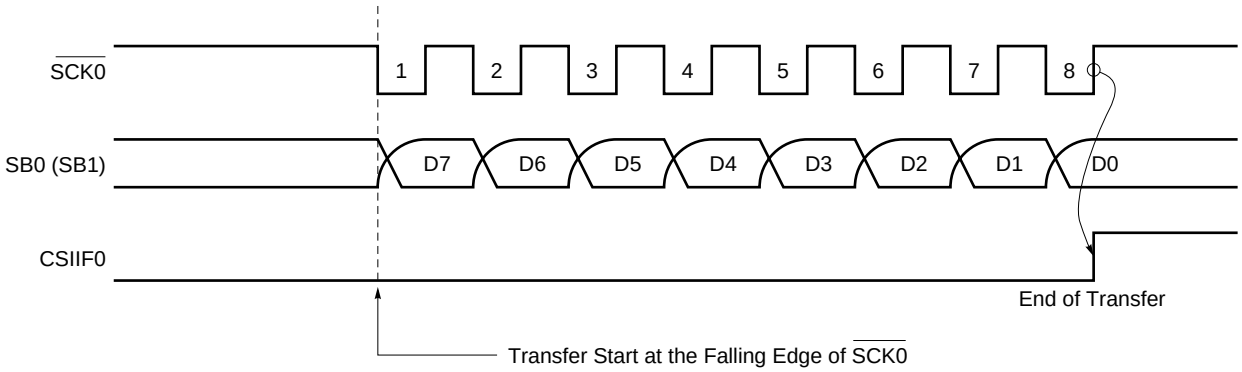
(2) Communication operation

The 2-wire serial I/O mode is used for data transmission/reception in 8-bit units. Data transmission/reception is carried out bit-wise in synchronization with the serial clock.

Shift operation of the serial I/O shift register 0 (SIO0) is carried out in synchronization with the falling edge of the serial clock ($\overline{SCK0}$). The transmit data is held in the SO0 latch and is output from the SB0/P25 (or SB1/P26) pin on an MSB-first basis. The receive data input from the SB0 (or SB1) pin is latched into the shift register at the rising edge of $\overline{SCK0}$.

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

Figure 13-32. 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 it is necessary to made high-impedance state the N-ch open-drain output for data reception, write FFH to SIO0 in advance.

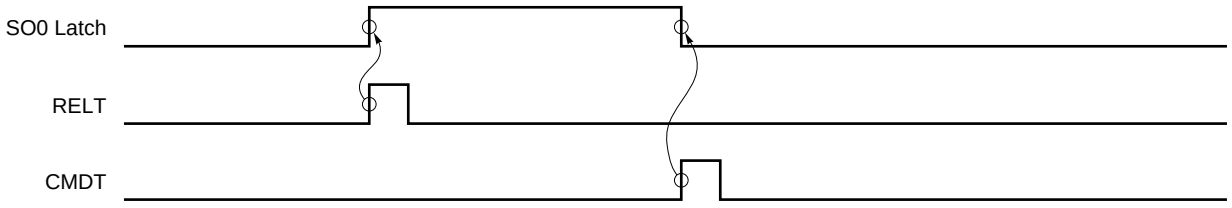
The SB0 (or SB1) pin generates the SO0 latch status and thus the SB0 (or SB1) pin output status can be manipulated by setting the bit 0 (RELT), bit 1 (CMDT) of the serial bus interface control register (SBIC). However, do not carry out this manipulation during serial transfer.

Control the $\overline{SCK0}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **13.4.5 $\overline{SCK0}$/P27 pin output manipulation**).

(3) Other signals

Figure 13-33 shows RELT and CMDT operations.

Figure 13-33. 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 be made high-impedance state for data reception, write FFH to SIO0 in advance.

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

(5) Error detection

In the 2-wire serial I/O mode, the serial bus SB0 (SB1) status being transmitted is fetched into the destination device, that is, serial I/O shift register 0 (SIO0). Thus, transmit error can be detected in the following way.

(a) Method of comparing SIO0 data before transmission to that after transmission

In this case, if two data differ from each other, a transmit error is judged to have occurred.

(b) Method of using the slave address register (SVA)

Transmit data is set to both SIO0 and SVA and is transmitted. After termination of transmission, COI bit (match signal coming from the address comparator) of the serial operating mode register 0 (CSIM0) is tested. If "1", normal transmission is judged to have been carried out. If "0", a transmit error is judged to have occurred.

13.4.5 $\overline{\text{SCK0}}$/P27 pin output manipulation

Because the $\overline{\text{SCK0}}$/P27 pin incorporates an output latch, static output is also possible by software in addition to normal serial clock output.

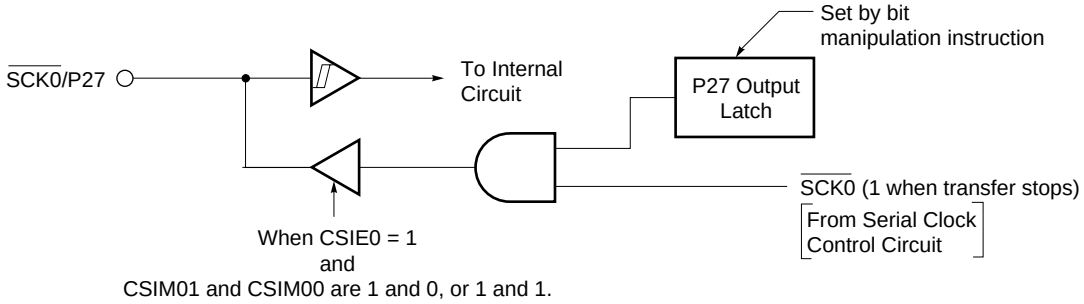
P27 output latch manipulation enables any number of $\overline{\text{SCK0}}$ to be set by software. (SI0/SB0 and SO0/SB1 pin to be controlled with the bit 0 (RELT) and bit 1 (CMDT) of the serial bus interface control register (SBIC).)

$\overline{\text{SCK0}}$/P27 pin output manipulating procedure is described below.

① Set the serial operating mode register 0 (CSIM0) ($\overline{\text{SCK0}}$ pin is set in the output mode and serial operation is enabled). While serial transfer is suspended, $\overline{\text{SCK0}}$ is set to 1.

② Manipulate the content of the P27 output latch by executing the bit manipulation instruction.

Figure 13-34. $\overline{\text{SCK0}}$/P27 Pin Configuration



CHAPTER 14 SERIAL INTERFACE CHANNEL 2

14.1 Serial Interface Channel 2 Functions

Serial interface channel 2 has the following three modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode
- 3-wire serial I/O mode

(1) Operation stop mode

This mode is used when serial transfer is not carried out to reduce power consumption.

(2) Asynchronous serial interface (UART) mode

In this mode, one byte of data is transmitted/received following the start bit, and full-duplex operation is possible.

A dedicated UART baud rate generator is incorporated, allowing communication over a wide range of baud rates. In addition, the baud rate can be defined by scaling the input clock to the ASCK pin.

The MIDI standard baud rate (31.25 kbps) can be used by employing the dedicated UART baud rate generator.

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

In this mode, 8-bit data transfer is performed using three lines: the serial clock ($\overline{SCK2}$), 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.

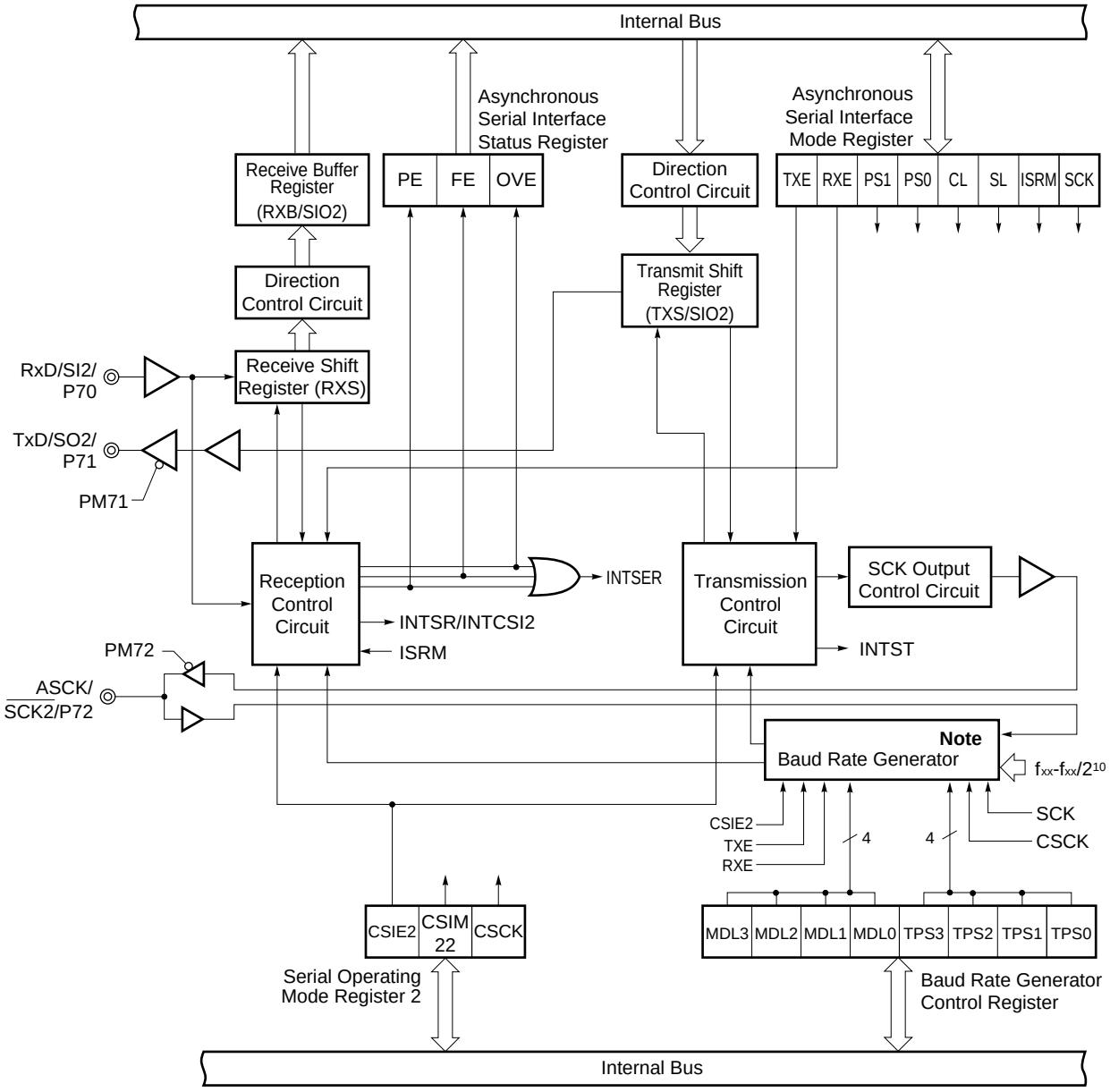
14.2 Serial Interface Channel 2 Configuration

Serial interface channel 2 consists of the following hardware.

Table 14-1. Serial Interface Channel 2 Configuration

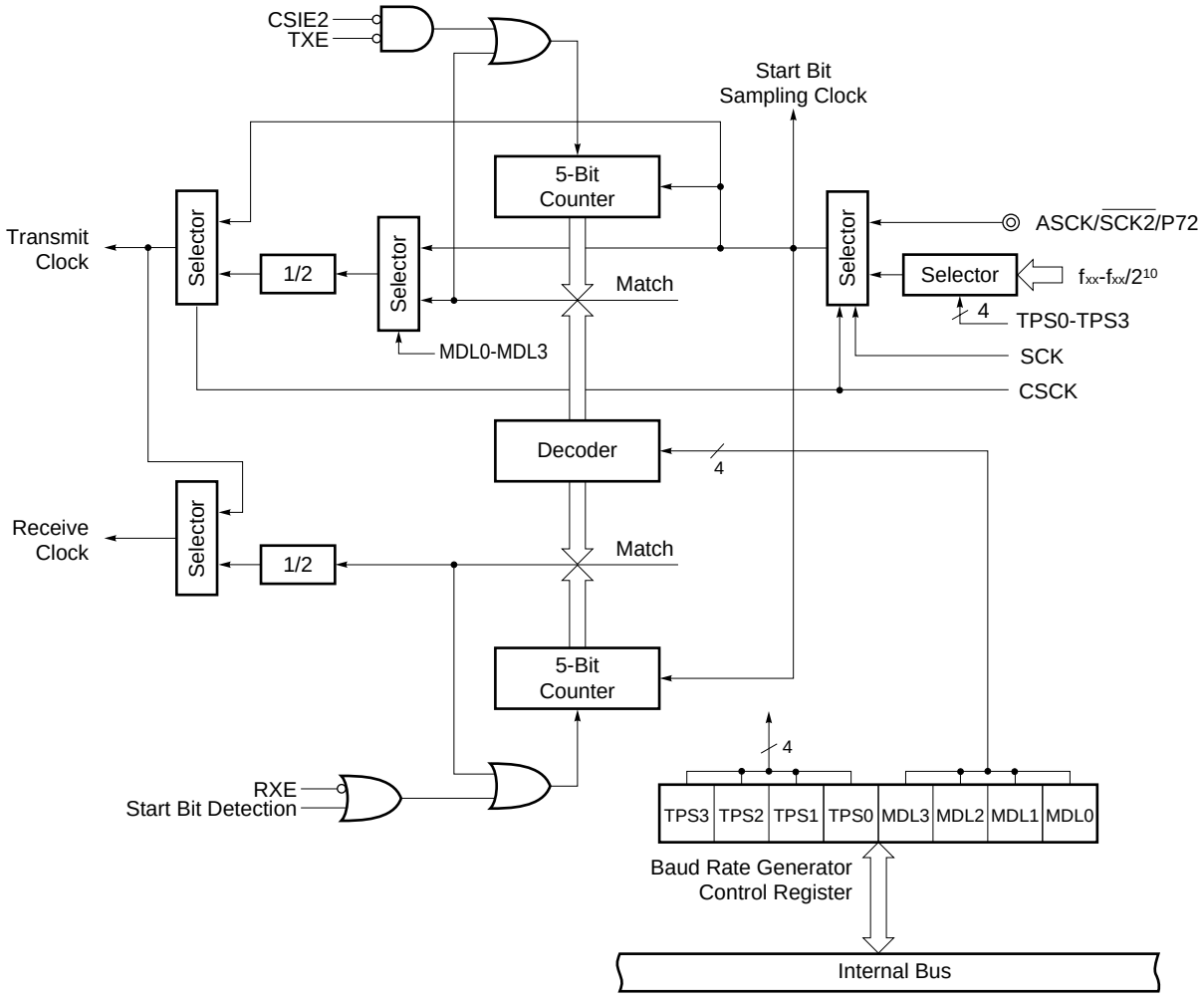
Item	Configuration
Register	Transmit shift register (TXS) Receive shift register (RXS) Receive buffer register (RXB)
Control register	Serial operating mode register 2 (CSIM2) Asynchronous serial interface mode register (ASIM) Asynchronous serial interface status register (ASIS) Baud rate generator control register (BRGC)

Figure 14-1. Serial Interface Channel 2 Block Diagram



Note Refer to Figure 14-2 for the baud rate generator configuration.

Figure 14-2. Baud Rate Generator Block Diagram



(1) Transmit shift register (TXS)

This register is used to set the transmit data. The data written in TXS is transmitted as serial data.

If the data length is specified as 7 bits, bits 0 to 6 of the data written in TXS are transferred as transmit data.

Writing data to TXS starts the transmit operation.

TXS is written to with an 8-bit memory manipulation instruction. It cannot be read.

TXS value is FFH after $\overline{\text{RESET}}$ input.

Caution TXS must not be written to during a transmit operation. TXS and the receive buffer register (RXB) are allocated to the same address, and when a read is performed, the value of RXB is read.

(2) Receive shift register (RXS)

This register is used to convert serial data input to the RxD pin to parallel data. When one byte of data is received, the receive data is transferred to the receive buffer register (RXB).

RXS cannot be directly manipulated by a program.

(3) Receive buffer register (RXB)

This register holds receive data. Each time one byte of data is received, new receive data is transferred from the receive shift register (RXS).

If the data length is specified as 7 bits, the receive data is transferred to bits 0 to 6 of RXB, and the MSB of RXB is always set to 0.

RXB is read with an 8-bit memory manipulation instruction. It cannot be written to.

RXB value is FFH after $\overline{\text{RESET}}$ input.

Caution RXB and the transmit shift register (TXS) are allocated to the same address, and when a write is performed, the value is written to TXS.

(4) Transmission control circuit

This circuit performs transmit operation control such as the addition of a start bit, parity bit and stop bit to data written in the transmit shift register (TXS) in accordance with the contents set in the asynchronous serial interface mode register (ASIM).

(5) Reception control circuit

This circuit controls receive operations in accordance with the contents set in the asynchronous serial interface mode register (ASIM). It performs error checks for parity errors, etc., during a receive operation, and if an error is detected, sets a value in the asynchronous serial interface status register (ASIS) in accordance with the error contents.

14.3 Serial Interface Channel 2 Control Registers

Serial interface channel 2 is controlled by the following four registers.

- Serial Operating Mode Register 2 (CSIM2)
- Asynchronous Serial Interface Mode Register (ASIM)
- Asynchronous Serial Interface Status Register (ASIS)
- Baud Rate Generator Control Register (BRGC)

(1) Serial operating mode register 2 (CSIM2)

This register is set when serial interface channel 2 is used in the 3-wire serial I/O mode.

CSIM2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM2 to 00H.

Figure 14-3. 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	CSIM22	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. Be sure to set bit 0 and bits 3 to 6 to 0.

2. When UART mode is selected, CSIM2 should be set to 00H.

(2) Asynchronous serial interface mode register (ASIM)

This register is set when serial interface channel 2 is used in the asynchronous serial interface mode.

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

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

Figure 14-4. 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 14-2. Serial Interface Channel 2 Operating Mode Settings

(1) Operation Stop Mode

ASIM			CSIM2			PM70	P70	PM71	P71	PM72	P72	Start Bit	Shift Clock	P70/SI2/ RxD Pin Functions	P71/SO2/ TxD Pin Functions	P72/$\overline{SCK2}$/ ASCK Pin Functions
TXE	RXE	SCK	CSIE2	CSIM22	CSCK											
0	0	×	0	×	×	×Note1	×Note1	×Note1	×Note1	×Note1	×Note1	—	—	P70	P71	P72
Other than above												Setting prohibited				

(2) 3-wire Serial I/O Mode

ASIM			CSIM2			PM70	P70	PM71	P71	PM72	P72	Start Bit	Shift Clock	P70/SI2/ RxD Pin Functions	P71/SO2/ TxD Pin Functions	P72/$\overline{SCK2}$/ ASCK Pin Functions
TXE	RXE	SCK	CSIE2	CSIM22	CSCK											
0	0	0	1	0	0	1Note2	×Note2	0	1	1	×	MSB	External clock	SI2Note2	SO2 (CMOS output)	$\overline{SCK2}$ input
					1					0	1		Internal clock			$\overline{SCK2}$ output
			1	1	0					1	×	LSB	External clock	SI2Note2	SO2 (CMOS output)	$\overline{SCK2}$ input
					1					0	1		Internal clock			$\overline{SCK2}$ output
Other than above												Setting prohibited				

(3) Asynchronous Serial Interface Mode

ASIM			CSIM2			PM70	P70	PM71	P71	PM72	P72	Start Bit	Shift Clock	P70/SI2/ RxD Pin Functions	P71/SO2/ TxD Pin Functions	P72/$\overline{SCK2}$/ ASCK Pin Functions
TXE	RXE	SCK	CSIE2	CSIM22	CSCK											
1	0	0	0	0	0	×Note1	×Note1	0	1	1	×	LSB	External clock	P70	TxD (CMOS output)	ASCK input
		1								×Note1	×Note1		Internal clock			P72
0	1	0	0	0	0	1	×	×Note1	×Note1	1	×		External clock	RxD	P71	ASCK input
		1								×Note1	×Note1		Internal clock			P72
1	1	0	0	0	0	1	×	0	1	1	×		External clock		TxD (CMOS output)	ASCK input
		1								×Note1	×Note1		Internal clock			P72
Other than above												Setting prohibited				

Notes 1. Can be used freely as port function.

2. Can be used as P70 (CMOS input/output) when only transmitter is used.

Remark × : Don't care

PM×× : Port mode register

P×× : Output latch of port

(3) Asynchronous serial interface status register (ASIS)

This is a register which displays the type of error when a reception error is generated in the asynchronous serial interface mode.

ASIS is read with a 1-bit or 8-bit memory manipulation instruction.

In 3-wire serial I/O mode, the contents of the ASIS are undefined.

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

Figure 14-5. 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 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.

(4) Baud rate generator control register (BRGC)

This register sets the serial clock for serial interface channel 2.

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

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

Figure 14-6. 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 14-6. 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 : Oscillation mode selection register (OSMS) bit 0

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.

$$[\text{Baud rate}] = \frac{f_{XX}}{2^n \times (k+16)} [\text{Hz}]$$

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 14-3. Relation 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
1200	C0H	1.73	B0H	1.73	BBH	1.14	ABH	1.14
2400	B0H	1.73	A0H	1.73	ABH	1.14	9BH	1.14
4800	A0H	1.73	90H	1.73	9BH	1.14	8BH	1.14
9600	90H	1.73	80H	1.73	8BH	1.14	7BH	1.14
19200	80H	1.73	70H	1.73	7BH	1.14	6BH	1.14
31250	74H	0	64H	0	71H	–1.31	61H	–1.31
38400	70H	1.73	60H	1.73	6BH	1.14	5BH	1.14
76800	60H	1.73	50H	1.73	5BH	1.14	—	—

Remark MCS: Oscillation mode selection register (OSMS) bit 0

(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.

$$[\text{Baud rate}] = \frac{f_{ASCK}}{2 \times (k+16)} \text{ [Hz]}$$

f_{ASCK} : Frequency of clock input to ASCK pin
k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

Table 14-4. Relation between ASCK Pin Input Frequency and Baud Rate (When BRGC is set to 00H)

Baud Rate (bps)	ASCK Pin Input Frequency
75	2.4 kHz
110	3.52 kHz
150	4.8 kHz
300	9.6 kHz
600	19.2 kHz
1200	38.4 kHz
2400	76.8 kHz
4800	153.6 kHz
9600	307.2 kHz
19200	614.4 kHz
31250	1000.0 kHz
38400	1228.8 kHz

14.4 Serial Interface Channel 2 Operation

Serial interface channel 2 has the following three modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode
- 3-wire serial I/O mode

14.4.1 Operation stop mode

In the operation stop mode, serial transfer is not performed, and therefore power consumption can be reduced.

In the operation stop mode, the P70/SI2/RxD, P71/SO2/TxD and P72/$\overline{\text{SCK2}}$/ASCK pins can be used as normal input/output ports.

(1) Register setting

Operation stop mode settings are performed using serial operating mode register 2 (CSIM2) and the asynchronous serial interface mode register (ASIM).

(a) Serial operating mode register 2 (CSIM2)

CSIM2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM2 to 00H.

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
CSIM2	CSIE2	0	0	0	0	CSIM 22	CSCK	0	FF72H	00H	R/W

CSIE2	Operation Control in 3-wire Serial I/O Mode
0	Operation stopped
1	Operation enabled

Caution Be sure to set bit 0 and bits 3 to 6 to 0.

(b) Asynchronous serial interface mode register (ASIM)

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

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

Symbol	⑦	⑥	5	4	3	2	1	0	Address	After Reset	R/W
ASIM	TXE	RXE	PS1	PS0	CL	SL	ISRM	SCK	FF70H	00H	R/W

RXE	Receive Operation Control
0	Receive operation stopped
1	Receive operation enabled

TXE	Transmit Operation Control
0	Transmit operation stopped
1	Transmit operation enabled

14.4.2 Asynchronous serial interface (UART) mode

In this mode, one byte of data is transmitted/received following the start bit, and full-duplex operation is possible.

A dedicated UART baud rate generator is incorporated, allowing communication over a wide range of baud rates. In addition, the baud rate can be defined by scaling the input clock to the ASCK pin.

The MIDI standard baud rate (31.25 kbps) can be used by employing the dedicated UART baud rate generator.

(1) Register setting

UART mode settings are performed using serial operating mode register 2 (CSIM2), the asynchronous serial interface mode register (ASIM), the asynchronous serial interface status register (ASIS), and the baud rate generator control register (BRGC).

(a) Serial operating mode register 2 (CSIM2)

CSIM2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM2 to 00H.

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
CSIM2	CSIE2	0	0	0	0	CSIM22	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 Be sure to set bit 0 and bits 3 to 6 to 0.

(b) Asynchronous serial interface mode register (ASIM)

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

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

Symbol	⑦	⑥	5	4	3	2	1	0	Address	After Reset	R/W
ASIM	TXE	RXE	PS1	PS0	CL	SL	ISRM	SCK	FF70H	00H	R/W

SCK	Clock Selection in Asynchronous Serial Interface Mode
0	Input clock from off-chip to ASCK pin
1	Dedicated baud rate generator output[Note]

ISRM	Control of Reception Completion Interrupt 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 sets ASIS to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ASIS	0	0	0	0	0	PE	FE	OVE	FF71H	00H	R

OVE	Overrun Error Flag
0	Overrun error not generated
1	Overrun error generated[Note 1] (When next receive operation is completed before data from receive buffer register is read)

FE	Framing Error Flag
0	Framing error not generated
1	Framing error generated[Note 2] (When stop bit is not detected)

PE	Parity Error Flag
0	Parity error not generated
1	Parity error generated (When transmit data parity does not match)

Notes 1. The receive buffer register (RXB) must be read when an overrun error is generated. Overrun errors will continue to be generated until RXB is read.

2. Even if the stop bit length has been set as 2 bits by bit 2 (SL) of the asynchronous serial interface mode register (ASIM), only single stop bit detection is performed during reception.

(d) Baud rate generator control register (BRGC)

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

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

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
BRGC	TPS3	TPS2	TPS1	TPS0	MDL3	MDL2	MDL1	MDL0	FF73H	00H	R/W

MDL3	MDL2	MDL1	MDL0	Baud Rate Generator Input Clock Selection	k
0	0	0	0	$f_{SCK}/16$	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

(continued)

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 : Oscillation mode selection register (OSMS) bit 0

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.

$$[\text{Baud rate}] = \frac{f_{XX}}{2^n \times (k+16)} \text{ [Hz]}$$

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 14-5. Relation 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
1200	C0H	1.73	B0H	1.73	BBH	1.14	ABH	1.14
2400	B0H	1.73	A0H	1.73	ABH	1.14	9BH	1.14
4800	A0H	1.73	90H	1.73	9BH	1.14	8BH	1.14
9600	90H	1.73	80H	1.73	8BH	1.14	7BH	1.14
19200	80H	1.73	70H	1.73	7BH	1.14	6BH	1.14
31250	74H	0	64H	0	71H	–1.31	61H	–1.31
38400	70H	1.73	60H	1.73	6BH	1.14	5BH	1.14
76800	60H	1.73	50H	1.73	5BH	1.14	—	—

Remark MCS: Oscillation mode selection register (OSMS) bit 0

(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.

$$[\text{Baud rate}] = \frac{f_{ASCK}}{2 \times (k+16)} \text{ [Hz]}$$

f_{ASCK} : Frequency of clock input to ASCK pin

k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

Table 14-6. Relation between ASCK Pin Input Frequency and Baud Rate (When BRGC is set to 00H)

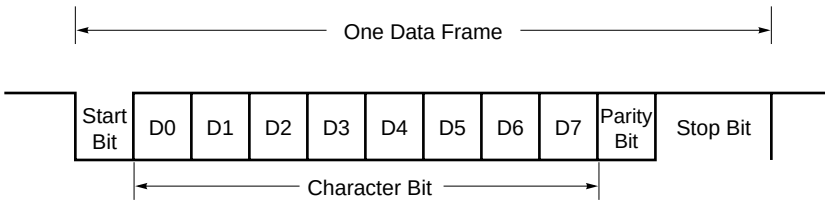
Baud Rate (bps)	ASCK Pin Input Frequency
75	2.4 kHz
110	3.52 kHz
150	4.8 kHz
300	9.6 kHz
600	19.2 kHz
1200	38.4 kHz
2400	76.8 kHz
4800	153.6 kHz
9600	307.2 kHz
19200	614.4 kHz
31250	1000.0 kHz
38400	1228.8 kHz

(2) Communication operation

(a) Data format

The transmit/receive data format is as shown in Figure 14-7.

Figure 14-7. 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 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 the ASIM and the baud rate generator control register (BRGC).

If a serial data receive error is generated, the receive error contents can be determined by reading the status of the asynchronous serial interface status register (ASIS).

(b) Parity types and operation

The parity bit is used to detect a bit error in the communication data. Normally, the same kind of parity bit is used on the transmitting side and the receiving side. With even parity and odd parity, a one-bit (odd number) error can be detected. With 0 parity and no parity, an error cannot be detected.

(i) Even parity

- **At transmission**

  Control is executed so that the number of bits with a value of "1" contained in the transmit data including parity bit is an even number. The parity bit value should be as follows.

  The number of bits with a value of "1" is an odd number in transmit data : 1
  The number of bits with a value of "1" is an even number in transmit data : 0

- **At reception**

  The number of bits with a value of "1" contained in the receive data including parity bit are counted, and if this is an odd number, a parity error is generated.

(ii) Odd parity

- **At transmission**

  Conversely to the situation with even parity, control is executed so that the number of bits with a value of "1" contained in the transmit data including parity bit is an odd number. The parity bit value should be as follows.

  The number of bits with a value of "1" is an odd number in transmit data : 0
  The number of bits with a value of "1" is an even number in transmit data : 1

- **At reception**

  The number of bits with a value of "1" contained in the receive data including parity bit are counted, and if this is an even number, a parity error is generated.

(iii) 0 Parity

When transmitting, the parity bit is set to "0" irrespective of the transmit data.
At reception, a parity bit check is not performed. Therefore, a parity error is not generated, irrespective of whether the parity bit is set to "0" or "1".

(iv) No parity

A parity bit is not added to the transmit data. At reception, data is received assuming that there is no parity bit. Since there is no parity bit, a parity error is not generated.

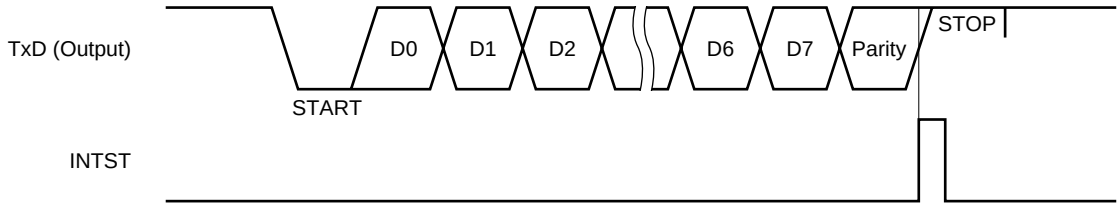
(c) Transmission

A transmit operation is started by writing transmit data to the transmit shift register (TXS). The start bit, parity bit and stop bit(s) are added automatically.

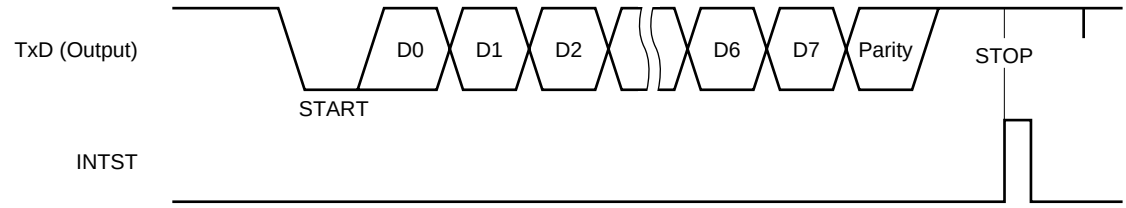
When the transmit operation starts, the data in the TXS is shifted out, and when the TXS is empty, a transmission completion interrupt request (INTST) is generated.

Figure 14-8. Asynchronous Serial Interface Transmission Completion Interrupt Request 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 register 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 (1), a receive operation is enabled and sampling of the RxD pin input is performed.

RxD pin input sampling is performed using the serial clock specified by ASIM.

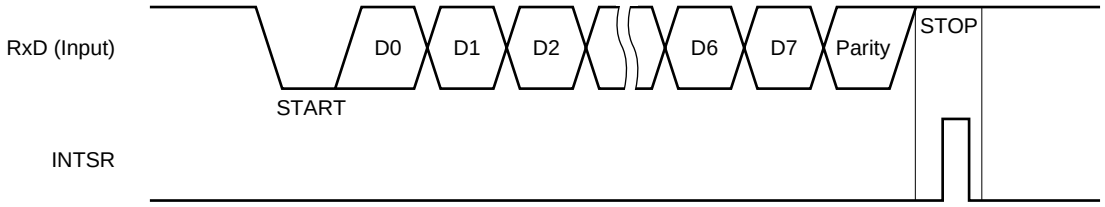
When the RxD pin input becomes low, the 5-bit counter of the baud rate generator (refer to **Figure 14-2**) starts counting, and at the time when the half time determined by specified baud rate has passed, the data sampling start timing signal is output. If the RxD pin input sampled again as a result of this start timing signal is low, it is identified as a start bit, the 5-bit counter is initialized and starts counting, and data sampling is performed. When character data, a parity bit and one stop bit are detected after the start bit, reception of one frame of data ends.

When one frame of data has been received, the receive data in the shift register is transferred to the receive buffer register (RXB), and a reception completion interrupt request (INTSR) is generated.

If an error is generated, the receive data in which the error was generated is still transferred to RXB, and INTSR is generated.

If the RXE bit is reset (0) during the receive operation, the receive operation is stopped immediately. In this case, the contents of RXB and ASIS are not changed, and INTSR and INTSER are not generated.

Figure 14-9. Asynchronous Serial Interface Reception Completion Interrupt Request 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 (INTSER) is generated. The reception error interrupt is generated before the reception completion interrupt (INTSR). Receive error causes are shown in Table 14-7.

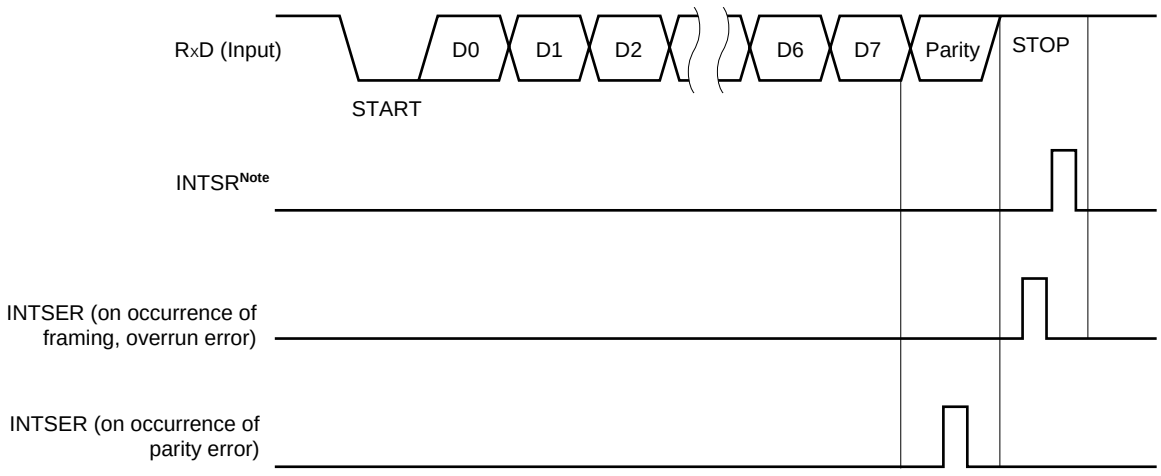
It is possible to determine what kind of error was generated during reception by reading the contents of the ASIS in the reception error interrupt servicing (INTSER) (refer to **Figures 14-9** and **14-10**).

The contents of ASIS are reset (0) by reading the receive buffer register (RXB) or receiving the next data (if there is an error in the next data, the corresponding error flag is set).

Table 14-7. 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 14-10. Receive Error Timing**



Cautions 1. The contents of the asynchronous serial interface status register (ASIS) are reset (0) by reading the receive buffer register (RXB) or receiving the next data. To ascertain the error contents, ASIS must be read before reading RXB.

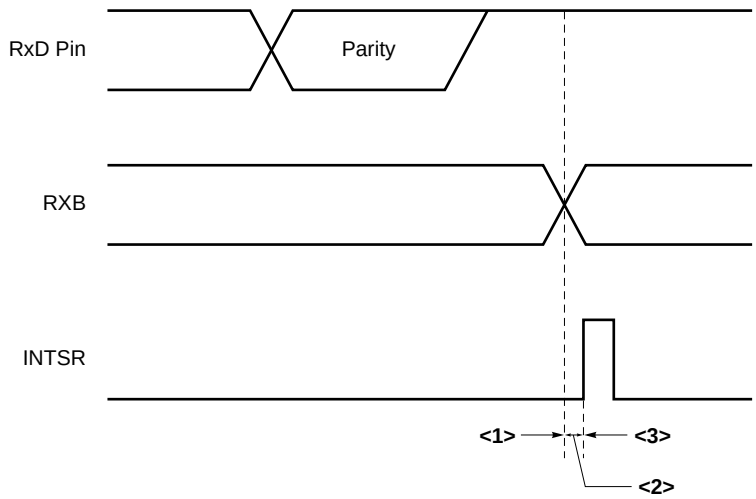
2. The receive buffer register (RXB) must be read even if a receive error is generated. If RXB is not read, an overrun error will be generated when the next data is received, and the receive error state will continue indefinitely.

(3) UART mode cautions

(a) When bit 7 (TXE) of the asynchronous serial interface mode register (ASIM) is cleared during transmission, be sure to set the transmit shift register (TXS) to FFH, then set the TXE to 1 before executing the next transmission.

(b) If the reception operation is stopped by clearing bit 6 (RXE) of the asynchronous serial interface mode register (ASIM) during reception, the status of the receive buffer register (RXB) and whether a reception completion interrupt request (INTSR) occurs differ depending on the timing. Figure 14-11 shows the timing.

Figure 14-11. Status of Receive Buffer Register (RXB) at Reception Stopped and Generation of Interrupt Request (INTSR)



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.

14.4.3 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connection of peripheral I/Os and display controllers, etc., which incorporate a conventional synchronous clocked serial interface, such as the 75X/XL series, 78K series, 17K series, etc.

Communication is performed using three lines: the serial clock ($\overline{SCK2}$), serial output (SO2), and serial input (SI2).

(1) Register setting

3-wire serial I/O mode settings are performed using serial operating mode register 2 (CSIM2), the asynchronous serial interface mode register (ASIM), and the baud rate generator control register (BRGC).

(a) Serial operating mode register 2 (CSIM2)

CSIM2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input sets CSIM2 to 00H.

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
CSIM2	CSIE2	0	0	0	0	CSIM22	CSCK	0	FF72H	00H	R/W

CSCK	Clock Selection in 3-wire Serial I/O Mode
0	Input clock from off-chip to $\overline{SCK2}$ 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 Be sure to set bit 0 and bits 3 to 6 to 0.

(b) Asynchronous serial interface mode register (ASIM)

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

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

When the 3-wire serial I/O mode is selected, 00H should be set in ASIM.

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 sets BRGC to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
BRGC	TPS3	TPS2	TPS1	TPS0	MDL3	MDL2	MDL1	MDL0	FF73H	00H	R/W

MDL3	MDL2	MDL1	MDL0	Baud Rate Generator Input Clock Selection	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	f_{SCK}	—

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 : Oscillation mode selection register (OSMS) bit 0

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-TPS3. Be sure then to set MDL0 to MDL3 to 1,1,1,1.

★ The serial clock frequency becomes half of the source clock frequency for the 5-bit counter.

(ii) When the baud rate generator is used:

Select a serial clock frequency with MDL0-MDL3 and TPS0-TPS3. Be sure then to set MDL0 to MDL3 to a value other than 1,1,1,1.

The serial clock frequency is calculated by the following formula:

$$\text{Serial clock frequency} = \frac{f_{XX}}{2^n \times (k + 16)} \text{ [Hz]}$$

f_X : Main system clock oscillation frequency

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

n : Value set in TPS0 to TPS3 ($1 \leq n \leq 11$)

k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

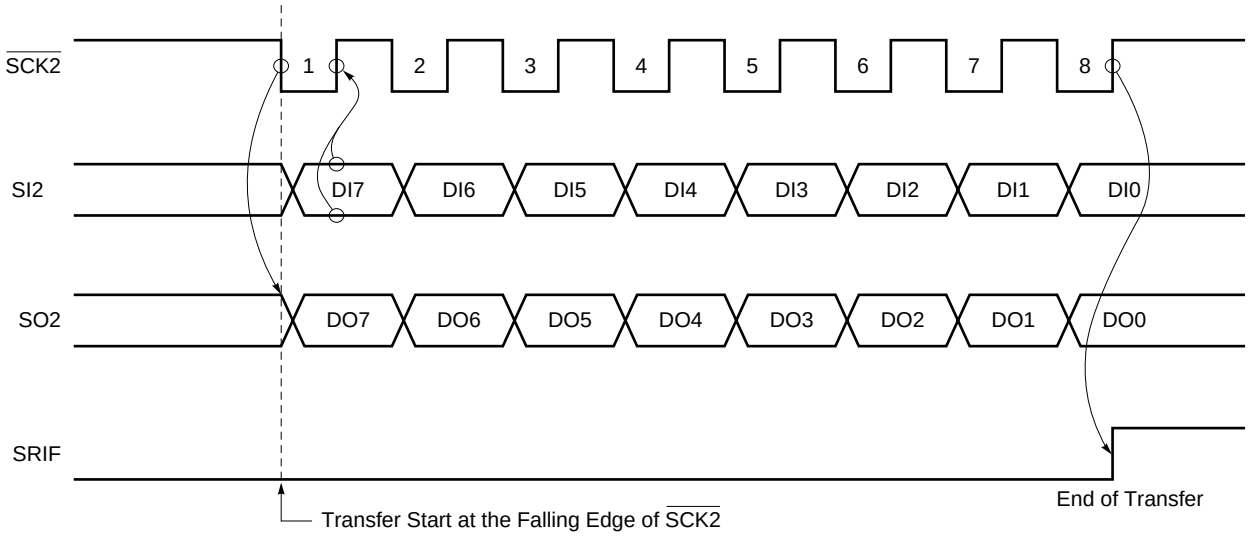
(2) Communication operation

In the 3-wire serial I/O mode, data transmission/reception is performed in 8-bit units. Data is transmitted/received bit by bit in synchronization with the serial clock.

Transmit shift register (TXS/SIO2) and receive shift register (RXS) shift operations are performed in synchronization with the fall of the serial clock ($\overline{SCK2}$). 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 $\overline{SCK2}$.

At the end of an 8-bit transfer, the operation of the transmit shift register (TXS/SIO2) or receive shift register (RXS) stops automatically, and the interrupt request flag (SRIF) is set.

Figure 14-12. 3-Wire Serial I/O Mode Timing



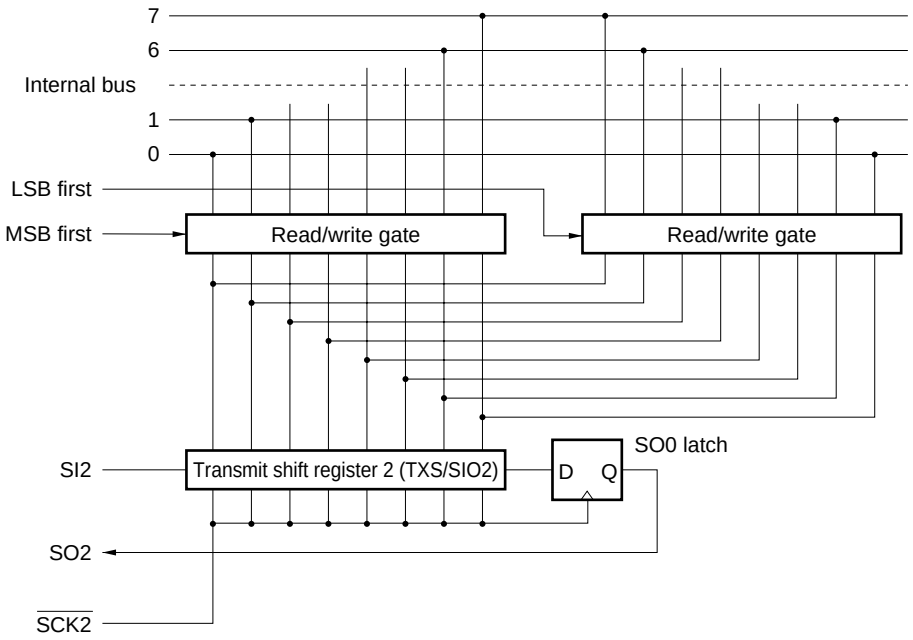
(3) Selecting MSB/LSB first

In the 3-wire serial I/O mode, a function to select whether data is transferred with its MSB or LSB first can be used.

Figure 14-13 shows the configuration of the transmit shift register (TXS/SIO3) and internal bus. As shown in the figure, data can be read or written by inverting the MSB or LSB.

Whether data is transferred with the MSB or LSB first can be specified by using bit 6 (CSIM02) of the serial operation mode register 2 (CSIM2).

Figure 14-13. Transfer Bit Sequence Select Circuit



The first bit to be transferred is selected by changing the bit sequency in which data is written to SIO2. The shift sequence of SIO2 is unchanged.

Therefore, select the first bit to be transferred (MSB or LSB) 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 the serial operation mode register 2 (CSIM2)

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

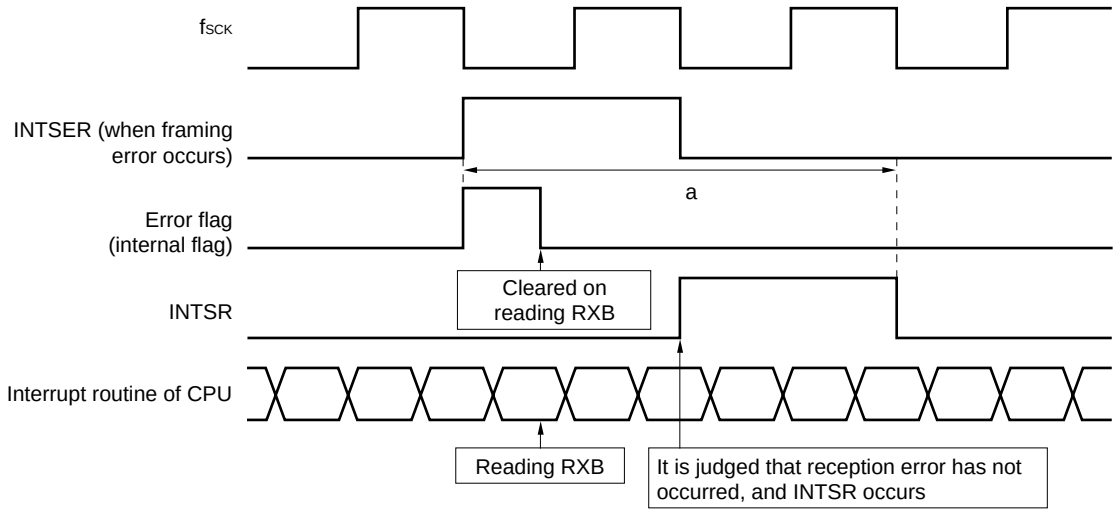
★ **14.4.4 Limitations when UART mode is used**

In the UART mode, the reception completion interrupt (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 (INTSR) does not occur on occurrence of a reception error. If the receive buffer register (RXB) is read at certain timing (a in Figure 14-14) 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 14-14 illustrates this operation.

Figure 14-14. Reception Completion Interrupt Generation Timing (when ISRM = 1)



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:

- **Measures**

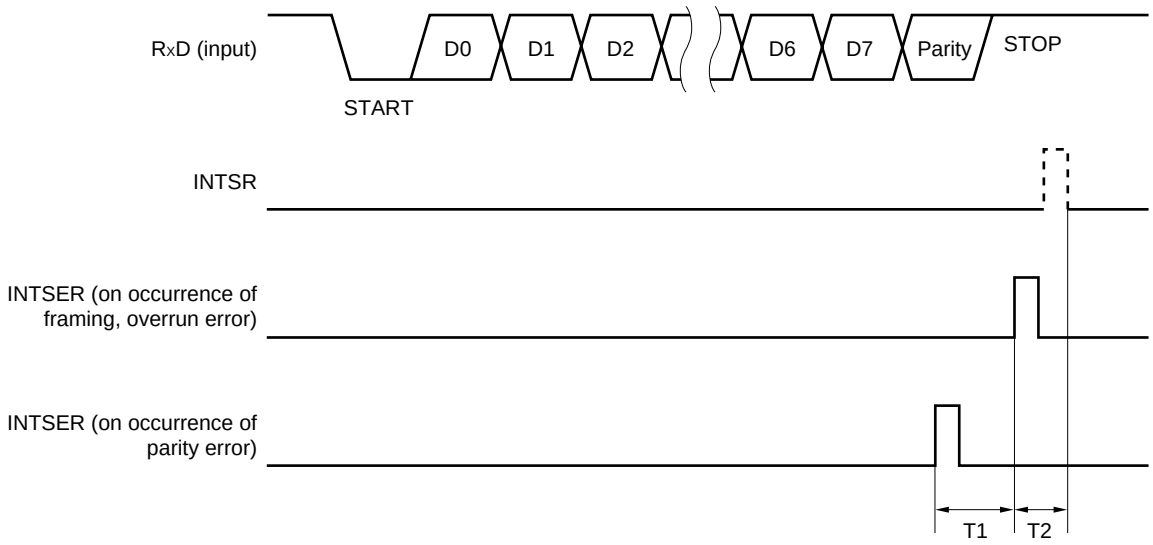
  - **In case of framing error or overrun error**

    Disable the receive buffer register (RXB) from being read for a certain time (T2 in Figure 14-15) after the reception error interrupt (INTSER) has occurred.

- **In case of parity error**

  Disable the receive buffer register (RXB) from being read for a certain time (T1 + T2 in Figure 14-15) after the reception error interrupt (INTSER) has occurred.

Figure 14-15. Receive Buffer Register Read Disable Period



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 (2400 bps selected as baud rate)

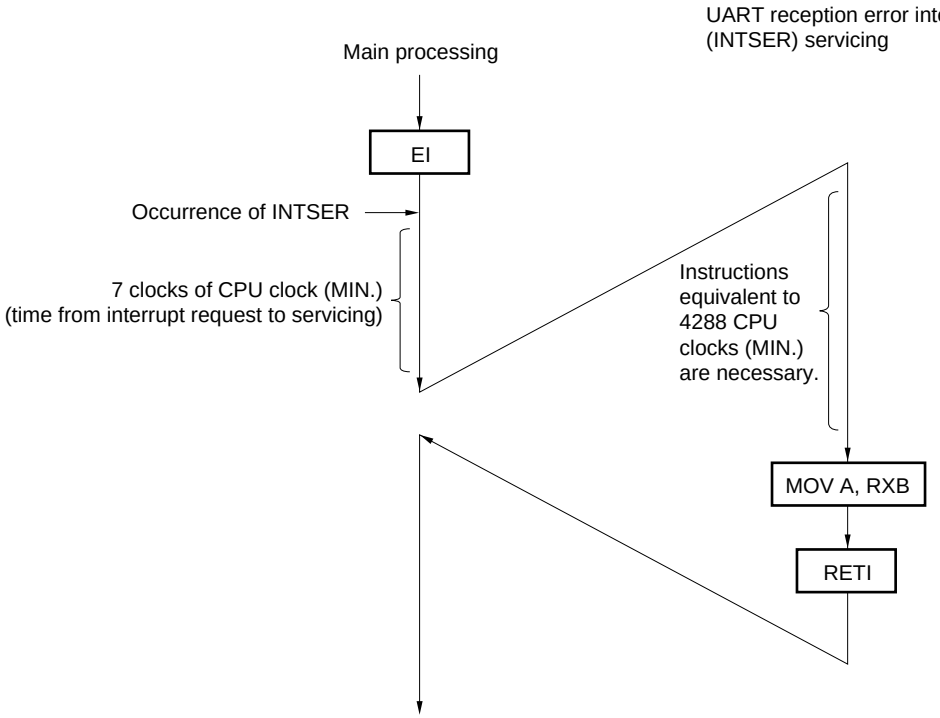
T_{CY} = 0.4 μs (t_{CY} = 0.2 μs)

$$T1 = \frac{1}{2400} = 833.4\ \mu s$$

$$T2 = 12.8 \times 2 = 25.6\ \mu s$$

$$\frac{T1 + T2}{t_{CY}} = 4295 \text{ (clocks)}$$

[Example]


UART reception error interrupt
(INTSER) servicing
Main processing
EI
Occurrence of INTSER
7 clocks of CPU clock (MIN.)
(time from interrupt request to servicing)
Instructions
equivalent to
4288 CPU
clocks (MIN.)
are necessary.
MOV A, RXB
RETI

[MEMO]

CHAPTER 15 LCD CONTROLLER/DRIVER

15.1 LCD Controller/Driver Functions

The functions of the LCD controller/driver incorporated in the μPD78064B subseries are shown below.

(1) Automatic output of segment signals and common signals is possible by automatic reading of the display data memory.
(2) Any of five display modes can be selected.
- Static
- 1/2 duty (1/2 bias)
- 1/3 duty (1/2 bias)
- 1/3 duty (1/3 bias)
- 1/4 duty (1/3 bias)

(3) Any of four frame frequencies can be selected in each display mode.
(4) Maximum of 40 segment signal outputs (S0 to S39); 4 common signal outputs (COM0 to COM3).
Sixteen of the segment signal outputs can be switched to input/output ports in units of 2 (P80/S39 to P87/S32, P90/S31 to P97/S24).
(5) In mask ROM versions, split resistors for LCD drive voltage generation can be incorporated by mask option.
(6) Operation on the subsystem clock is also possible.

The maximum number of displayable pixels in each display mode is shown in Table 15-1.

Table 15-1. Maximum Number of Display Pixels

Bias Method	Time Division	Common Signals Used	Maximum Number of Pixels
–	Static	COM0 (COM1, 2, 3)	40 (40 segments × 1 common)**Note 1**
1/2	2	COM0, COM1	80 (40 segments × 2 commons)**Note 2**
	3	COM0-COM2	120 (40 segments × 3 commons)**Note 3**
1/3	3		
	4	COM0-COM3	160 (40 segments × 4 commons)**Note 4**

Notes 1. 5 digits on 8. type LCD panel with 8 segments/digit.
2. 10 digits on 8. type LCD panel with 4 segments/digit.
3. 13 digits on 8. type LCD panel with 3 segments/digit.
4. 20 digits on 8. type LCD panel with 2 segments/digit.

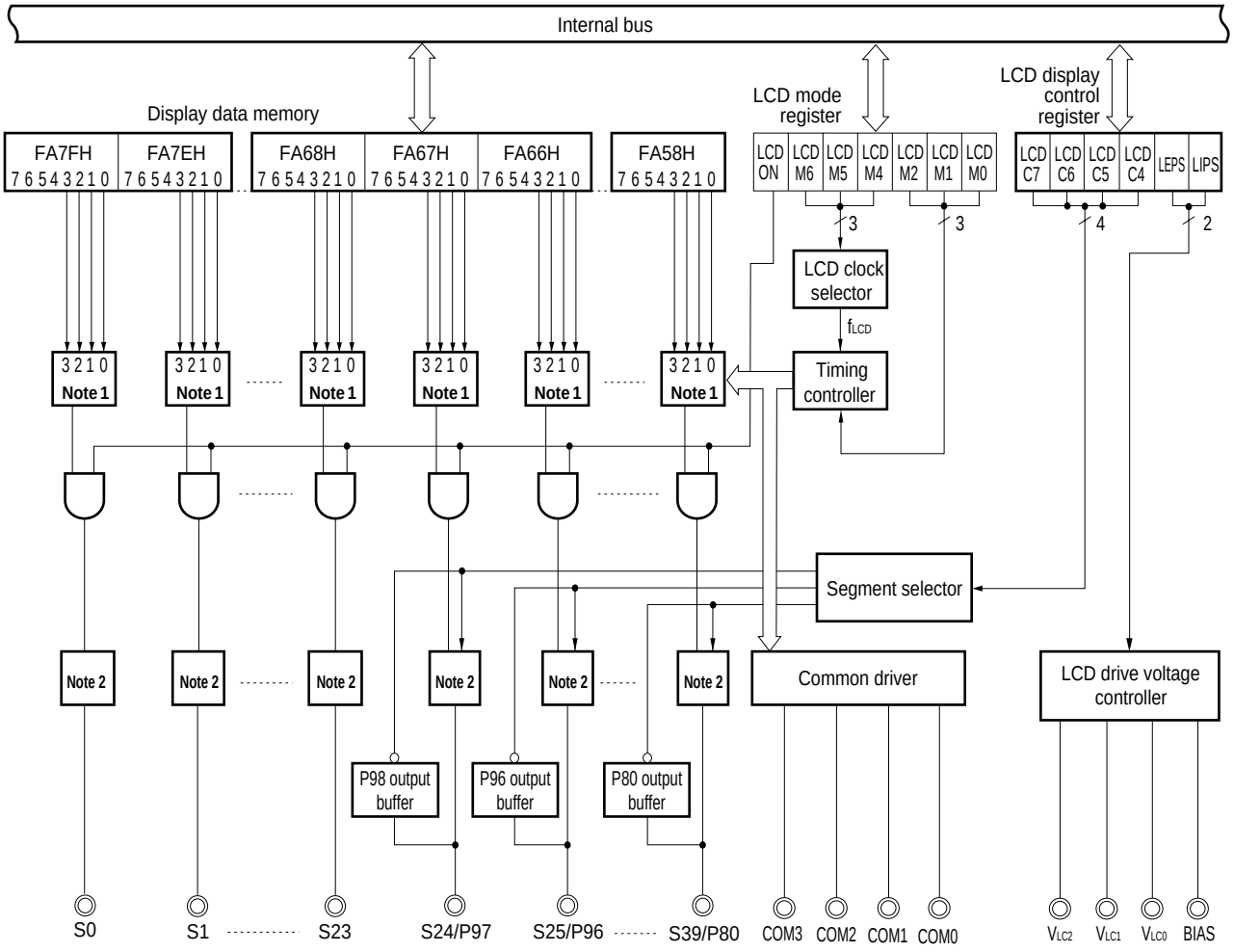
15.2 LCD Controller/Driver Configuration

The LCD controller/driver is composed of the following hardware.

Table 15-2. LCD Controller/Driver Configuration

Item	Configuration
Display outputs	Segment signals : 40 Dedicated segment signals: 24 Segment signal/input/output port dual function: 16 Common signals : 4 (COM0 to COM3)
Control registers	LCD display mode register (LCDM) LCD display control register (LCDC)

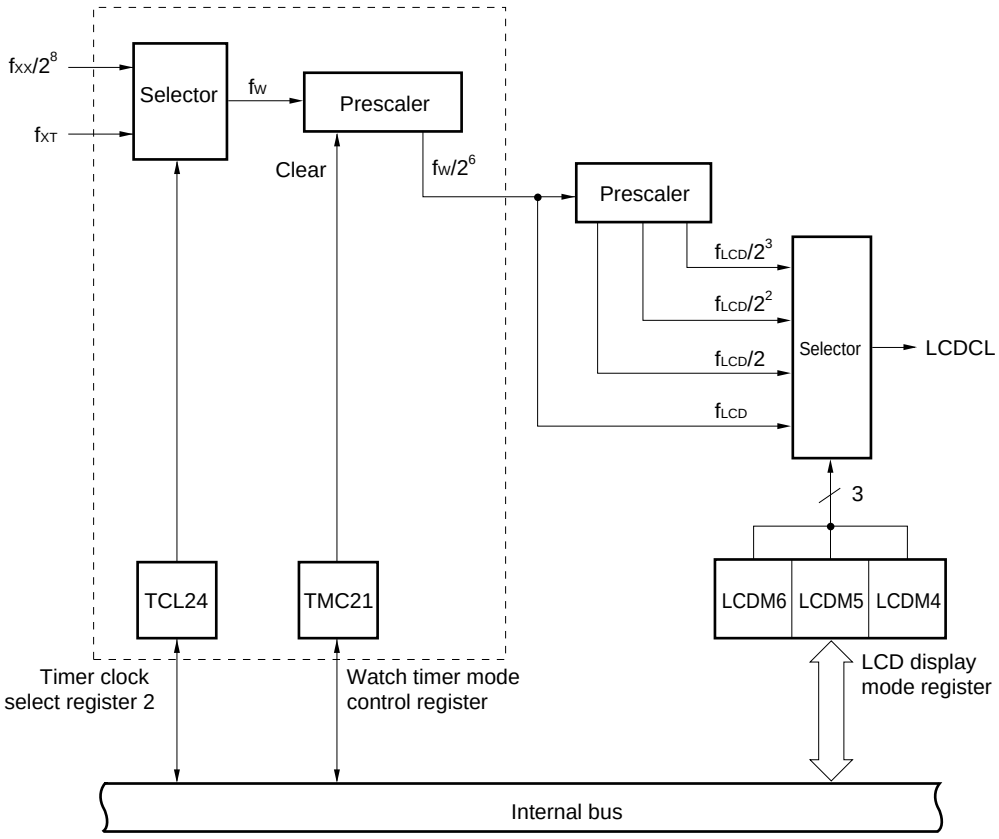
Figure 15-1. LCD Controller/Driver Block Diagram



Notes 1. Selector

2. Segment driver

Figure 15-2. LCD Clock Select Circuit Block Diagram



Remarks 1. The watch timer includes the circuit enclosed with the dotted line.

2. LCDCL : LCD clock

3. f_{LDC} : LCD clock frequency

15.3 LCD Controller/Driver Control Registers

The LCD controller/driver is controlled by the following two registers.

- LCD display mode register (LCDM)
- LCD display control register (LCDC)

(1) LCD display mode register (LCDM)

This register sets display operation enabling/ disabling, the LCD clock, frame frequency, and display mode selection.

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

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

Figure 15-3. LCD Display Mode Register Format

Symbol	⑦	6	5	4	3	2	1	0	Address	State after reset	R/W
LCDM	LCDON	LCDM6	LCDM5	LCDM4	0	LCDM2	LCDM1	LCDM0	FFB0H	00H	R/W

LCDM2	LCDM1	LCDM0	Number of Time Divisions	Bias Method
0	0	0	4	1/3
0	0	1	3	1/3
0	1	0	2	1/2
0	1	1	3	1/2
1	0	0	Static	
Other than above			Setting prohibited	

LCDM5	LCDM5	LCDM4	LCD Clock Selection (See **Note**)		
			f_{XX} = 5.0 MHz	f_{XX} = 4.19 MHz	f_{XT} = 32.768 kHz
0	0	0	$f_W/2^9$ (76 Hz)	$f_W/2^9$ (64 Hz)	$f_W/2^9$ (64 Hz)
0	0	1	$f_W/2^8$ (153 Hz)	$f_W/2^8$ (128 Hz)	$f_W/2^8$ (128 Hz)
0	1	0	$f_W/2^7$ (305 Hz)	$f_W/2^7$ (256 Hz)	$f_W/2^7$ (256 Hz)
0	1	1	$f_W/2^6$ (610 Hz)	$f_W/2^6$ (512 Hz)	$f_W/2^6$ (512 Hz)

LCDON	LCD Display
0	Display on (All segment outputs signal non-selection.)
1	Display off

Note The LCD clock is supplied from the watch timer. When LCD display is performed, 1 should be set in bit 1 (TMC21) of the watch timer mode control register (TMC2). If TMC21 is reset to 0 during LCD display, the LCD clock supply will be stopped and the display will be disrupted.

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

Table 15-3. Frame Frequencies (Hz)

LCDCL / Duty	$f_W/2^9$ (64 Hz)	$f_W/2^8$ (128 Hz)	$f_W/2^7$ (256 Hz)	$f_W/2^6$ (512 Hz)
Static	64	128	256	512
1/2	32	64	128	256
1/3	21	43	85	171
1/4	16	32	64	128

Remarks 1. Figures in parentheses apply to operation with f_X = 4.19 MHz or f_{XT} = 32.768 kHz.

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

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

4. f_X : Main system clock oscillation frequency

5. f_{XT} : Subsystem clock oscillation frequency

(2) LCD display control register (LCDC)

This register sets cut-off of the current flowing to split resistors for LCD drive voltage generation and switchover between segment output and input/output port functions.

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

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

Figure 15-4. LCD Display Control Register Format

Symbol	7	6	5	4	3	2	1	0	Address	State after reset	R/W
LCDC	LCDC7	LCDC6	LCDC5	LCDC4	0	0	LEPS	LIPS	FFB2H	00H	R/W

LEPS	LIPS	LCD driving power supply selection
0	0	Does not supply power to LCD.
0	1	Supplies power to LCD from V_{DD} pin.
1	0	Supplies power to LCD from BIAS pin. (Shorts BIAS and V_{LC0} pins internally.)
1	1	Setting prohibited

LCDC7	LCDC6	LCDC5	LCDC4	P80/S39-P97/S24 pin functions	
				Port pins	Segment pins
0	0	0	0	P80-P97	None
0	0	0	1	P80-P95	S24, S25
0	0	1	0	P80-P93	S24-S27
0	0	1	1	P80-P91	S24-S29
0	1	0	0	P80-P87	S24-S31
0	1	0	1	P80-P85	S24-S33
0	1	1	0	P80-P83	S24-S35
0	1	1	1	P80-P81	S24-S37
1	0	0	0	None	S24-S39
Other than above				Setting prohibited	

Cautions 1. Pins which perform segment output cannot be used as output port pins even if 0 is set in the port mode register (PM).

2. If a pin which performs segment output is read as a port, its value will be 0.

3. Pins set as segment outputs by LCDC cannot have an internal pull-up resistor used regardless of the value of bits 0 and 1 (PUO8 and PUO9) of pull-up resistor option register H (PUOH).

15.4 LCD Controller/Driver Settings

LCD controller/driver settings should be performed as shown below. When the LCD controller/driver is used, the watch timer should be set to the operational state beforehand.

<1> Set “watch operation enabled” in timer clock selection register 2 (TCL2) and the watch timer mode control register (TMC2).

<2> Set the initial value in the display data memory (FA58H to FA7FH).

<3> Set the pins to be used as segment outputs in the LCD display control register (LCDC).

<4> Set the display mode and LCD clock in the LCD display mode register (LCDM).

Next, set data in the display data memory according to the display contents.

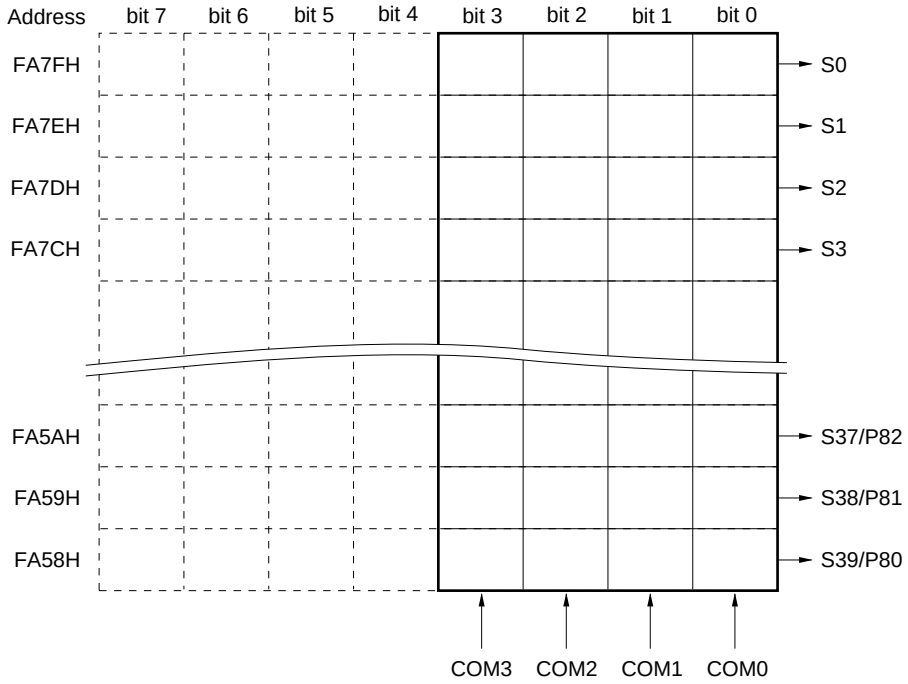
15.5 LCD Display Data Memory

The LCD display data memory is mapped onto addresses FA58H to FA7FH. The data stored in the LCD display data memory can be displayed on an LCD panel by the LCD controller/driver.

Figure 15-5 shows the relationship between the LCD display data memory contents and the segment outputs/common outputs.

Any area not used for display can be used as normal RAM.

Figure 15-5. Relationship between LCD Display Data Memory Contents and Segment/Common Outputs



Caution The high-order 4 bits of the LCD display data memory do not incorporate memory. Be sure to set them to 0.

15.6 Common Signals and Segment Signals

An individual pixel on an LCD panel lights when the potential difference of the corresponding common signal and segment signal reaches or exceeds a given voltage (the LCD drive voltage V_{LCD}).

As an LCD panel deteriorates if a DC voltage is applied in the common signals and segment signals, it is driven by AC voltage.

(1) Common signals

For common signals, the selection timing order is as shown in Table 15-4 according to the number of time divisions set, and operations are repeated with these as the cycle. In the static mode, the same signal is output to COM0 through COM3.

With 2-time-division operation, pins COM2 and COM3 are left open, and with 3-time-division operation, the COM3 pin is left open.

Table 15-4. COM Signals

COM signal / Time division	COM0	COM1	COM2	COM3
Static				
2-time division			Open	Open
3-time division				Open
4-time division				

(2) Segment signals

Segment signals correspond to a 40-byte LCD display data memory (FA58H to FA7FH). Each display data memory bit 0, bit 1, bit 2, and bit 3 is read in synchronization with the COM0, COM1, COM2 and COM3 timings respectively, and if the value of the bit is 1, it is converted to the selection voltage. If the value of the bit is 0, it is converted to the non-selection voltage and output to a segment pin (S0 to S39) (S24 to S39 have a dual function as input/output port pins).

Consequently, it is necessary to check what combination of front surface electrodes (corresponding to the segment signals) and rear surface electrodes (corresponding to the common signals) of the LCD display to be used form the display pattern, and then write bit data corresponding on a one-to-one basis with the pattern to be displayed.

In addition, because LCD display data memory bits 1 and 2 are not used with the static method, bits 2 and 3 are not used with the 2-time-division method, and bit 3 is not used with the 3-time-division method, these can be used for other than display purposes.

Bits 4 to 7 are fixed at 0.

(3) Common signal and segment signal output waveforms

The voltages shown in Table 15-5 are output in the common signals and segment signals.

The $\pm V_{LCD}$ ON voltage is only produced when the common signal and segment signal are both at the selection voltage; other combinations produce the OFF voltage.

Table 15-5. LCD Drive Voltages

(a) Static display mode

Segment / Common	Select	Non-select
	V_{SS} , V_{LC0}	V_{LC0} , V_{SS}
V_{LC0} , V_{SS}	$-V_{LCD}$, $+V_{LCD}$	0 V , 0 V

(b) 1/2 bias method

Segment / Common		Select	Non-select
		V_{SS} , V_{LC0}	V_{LC0} , V_{SS}
Select level	V_{LC0} , V_{SS}	$-V_{LCD}$, $+V_{LCD}$	0 V , 0 V
Non-select level	$V_{LC1}=V_{LC2}$	$-1/2V_{LCD}$, $+1/2V_{LCD}$	$+1/2V_{LCD}$, $-1/2V_{LCD}$

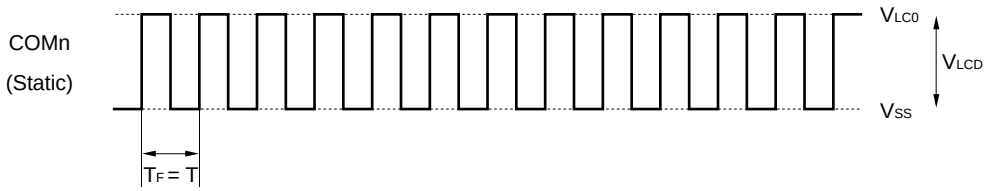
(c) 1/3 bias method

Segment / Common		Select	Non-select
		V_{SS} , V_{LC0}	V_{LC1} , V_{LC2}
Select level	V_{LC0} , V_{SS}	$-V_{LCD}$, $+V_{LCD}$	$-1/3V_{LCD}$, $+1/3V_{LCD}$
Non-select level	V_{LC2} , V_{LC1}	$-1/3V_{LCD}$, $+1/3V_{LCD}$	$-1/3V_{LCD}$, $+1/3V_{LCD}$

Figure 15-6 shows the common signal waveform, and Figure 15-7 shows the common signal and segment signal voltages and phases.

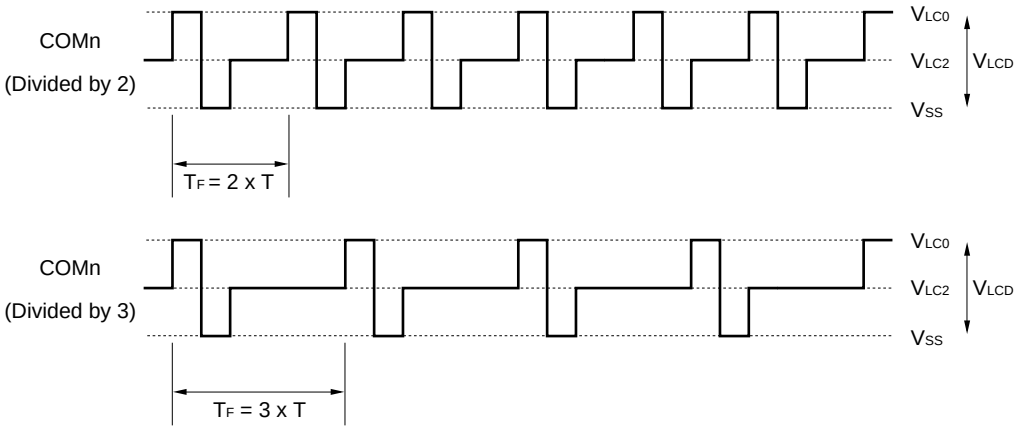
Figure 15-6. Common Signal Waveform

(a) Static display mode



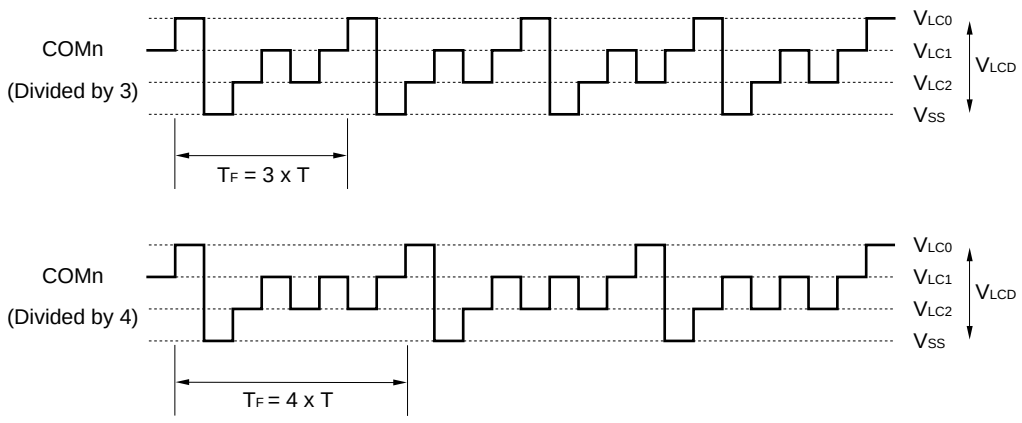
T : One LCDCL cycle
T_F : Frame frequency

(b) 1/2 bias method



T : One LCDCL cycle
T_F : Frame frequency

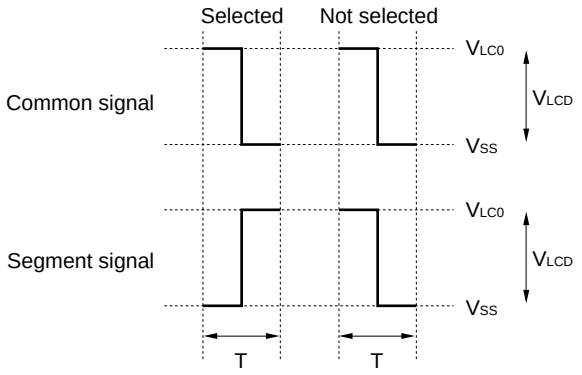
(c) 1/3 bias method



T : One LCDCL cycle
T_F : Frame frequency

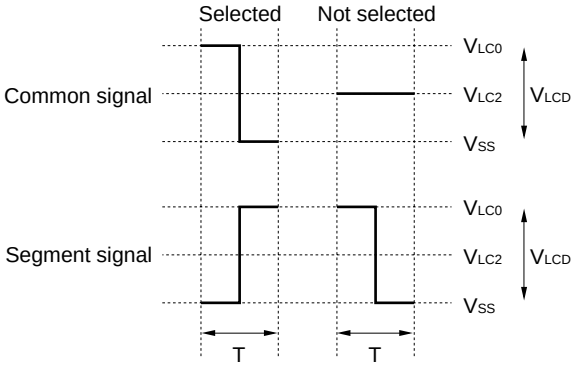
Figure 15-7. Common Signal and Static Signal Voltages and Phases

(a) Static display mode



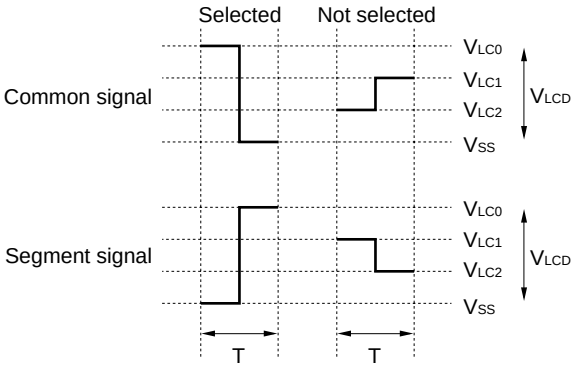
Remark T : One LCDCL cycle

(b) 1/2 bias method



Remark T : One LCDCL cycle

(c) 1/3 bias method



Remark T : One LCDCL cycle

15.7 Supply of LCD Drive Voltages V_{LC0}, V_{LC1}, V_{LC2}

Dividing resistors for producing the LCD drive voltages can be incorporated in the μPD78064B by mask option (the μPD78P064B does not incorporate dividing resistors). Incorporating the dividing resistors makes it possible to produce LCD drive voltages appropriate to the various bias methods shown in Table 15-6 without using external dividing resistors.

Also, an LCD drive voltage can be externally supplied from the BIAS pin to produce other LCD drive voltages.

Table 15-6. LCD Drive Voltages (with On-Chip Dividing Resistor)

Bias Method / LCD Drive Voltage Pin	No Bias (Static Mode)	1/2 Bias	1/3 Bias
V_{LC0}	V_{LCD}	V_{LCD}	V_{LCD}
V_{LC1}	2/3 V_{LCD}	1/2 V_{LCD} **Note**	2/3 V_{LCD}
V_{LC2}	1/3 V_{LCD}		1/3 V_{LCD}

Note With the 1/2 bias method, the V_{LC1} pin and V_{LC2} pin must be connected externally.

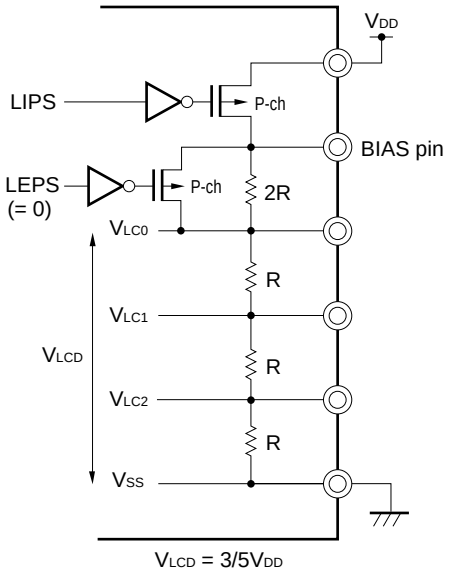
Remarks 1. When the BIAS pin and V_{LC0} pin are open, V_{LCD} = 3/5 V_{DD} (with on-chip dividing resistor).

2. When the BIAS pin and V_{LC0} pin are connected, V_{LCD} = V_{DD}.

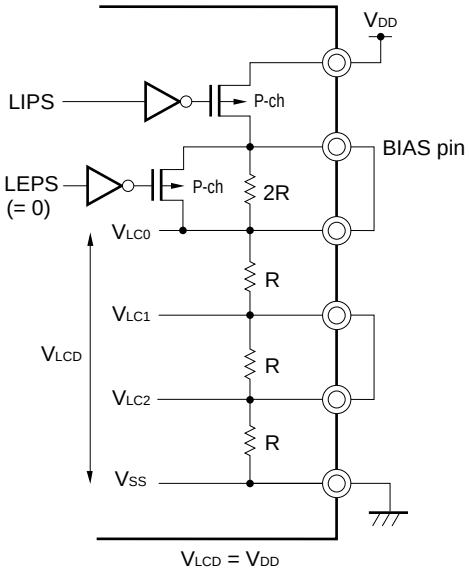
Examples of internal supply of the LCD drive voltage in accordance with Table 15-6 are shown in Figures 15-8 and 15-9. An example of supply of the LCD drive voltage from off-chip is shown in Figure 15-10. Stepless LCD drive voltages can be supplied by means of variable resistor r.

Figure 15-8. LCD Drive Power Supply Connection Examples (with On-Chip Dividing Resistor)

(a) 1/3 bias method and static display mode (Example with V_{DD} = 5 V, V_{LCD} = 3 V)



(b) 1/2 bias method mode (Example with V_{DD} = 5 V, V_{LCD} = 5 V)



(c) 1/3 bias method and static display mode (Example with V_{DD} = 5 V, V_{LCD} = 5 V)

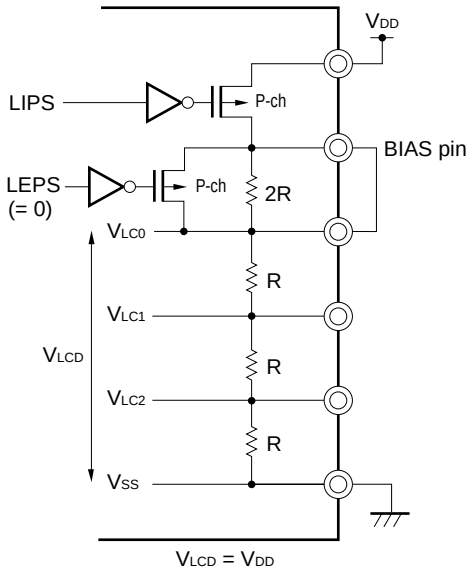
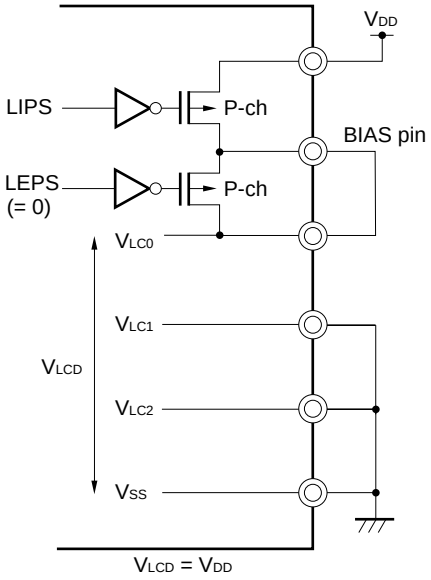
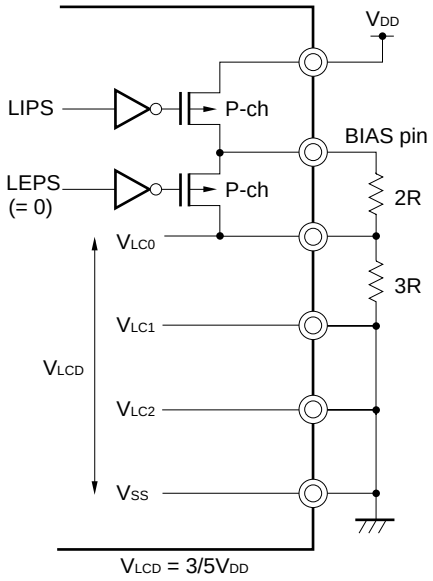


Figure 15-9. LCD Drive Power Supply Connection Examples (with External Dividing Resistor)

(a) Static display mode [Note] (Example with V_{DD} = 5 V, V_{LCD} = 5 V)

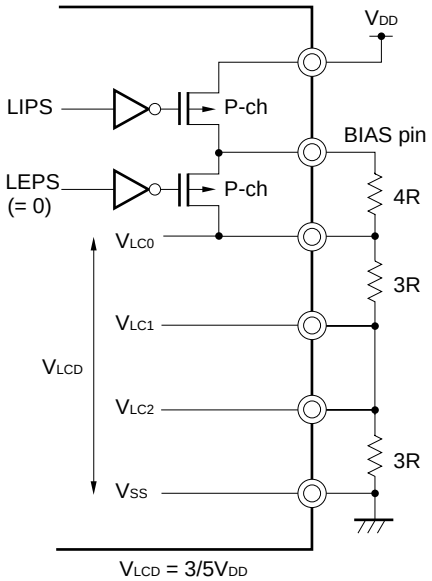


(b) Static display mode (Example with V_{DD} = 5 V, V_{LCD} = 3 V)



Note LIPS should always be set to 1 (including in standby mode).

(c) 1/2 bias method (Example with V_{DD} = 5 V, V_{LCD} = 3 V)



(d) 1/3 bias method (Example with V_{DD} = 5 V, V_{LCD} = 3 V)

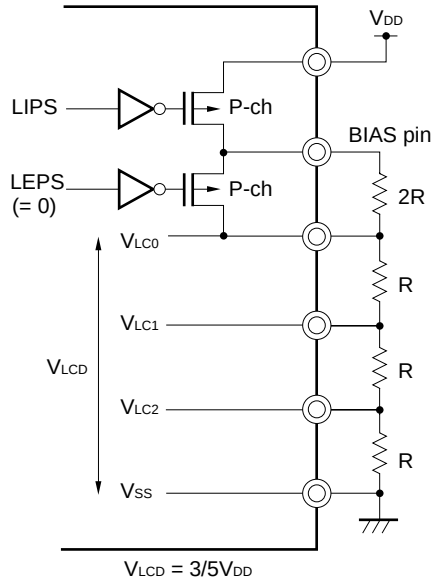
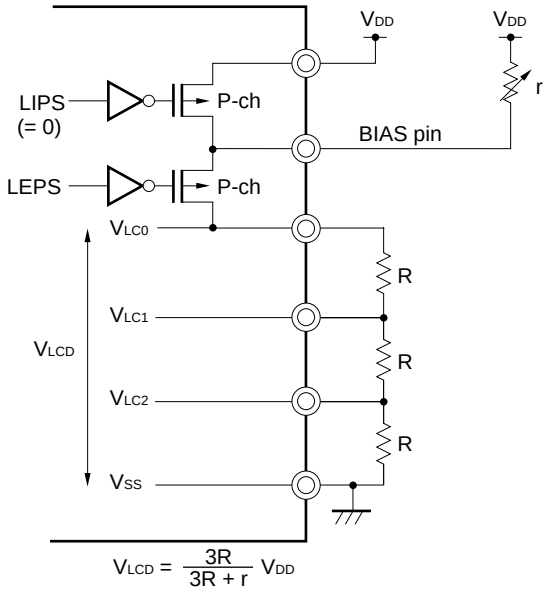


Figure 15-10. Example of LCD Drive Voltage Supply from Off-Chip



15.8 Display Modes

15.8.1 Static Display Example

Figure 15-12 shows the connection of a static type 5-digit LCD panel with the display pattern shown in Figure 15-11 with the μPD78064B subseries segment (S0 to S39) and common (COM0) signals. The display example is "123.45," and the display data memory contents (addresses FA58H to FA7FH) correspond to this.

An explanation is given here taking the example of the third digit "3." (3.). In accordance with the display pattern in Figure 15-11, selection and non-selection voltages must be output to pins S16 through S23 as shown in Table 15-7 at the COM0 common signal timing.

Table 15-7. Selection and Non-Selection Voltages (COM0)

Segment / Common	S16	S17	S18	S19	S20	S21	S22	S23
COM0	S	S	S	S	NS	S	NS	S

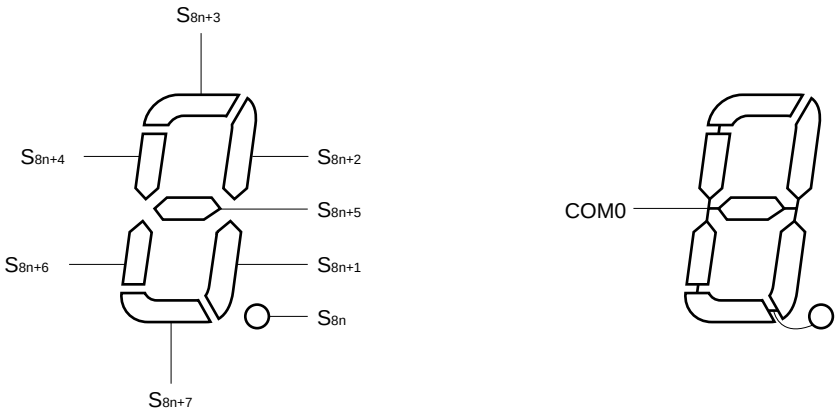
S: Selection, NS: Non-selection

From this, it can be seen that 10101111 must be prepared in the BIT0 bits of the display data memory (addresses FA68H to FA6FH) corresponding to S16 to S23.

The LCD drive waveforms for S19, S20, and COM0 are shown in Figure 15-13. When S19 is at the selection voltage at the timing for selection with COM0, it can be seen that the $+V_{LCD}/-V_{LCD}$ AC square wave, which is the LCD illumination (ON) level, is generated.

Shorting the COM0 through COM3 lines increases the current drive capability because the same waveform as COM0 is output to COM1 through COM3.

Figure 15-11. Static LCD Display Pattern and Electrode Connections



Remark n = 0 to 4

Figure 15-12. Static LCD Panel Connection Example

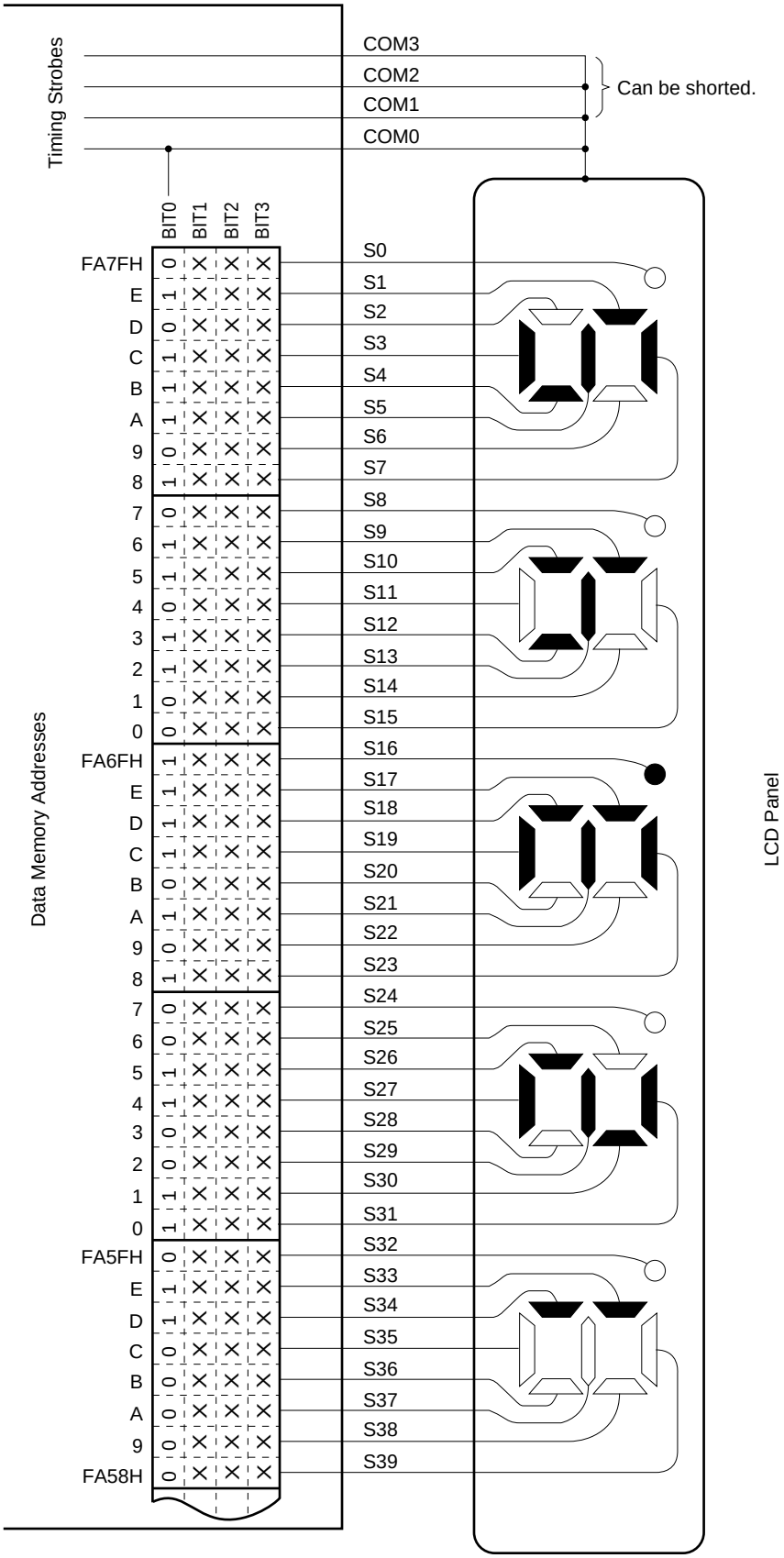
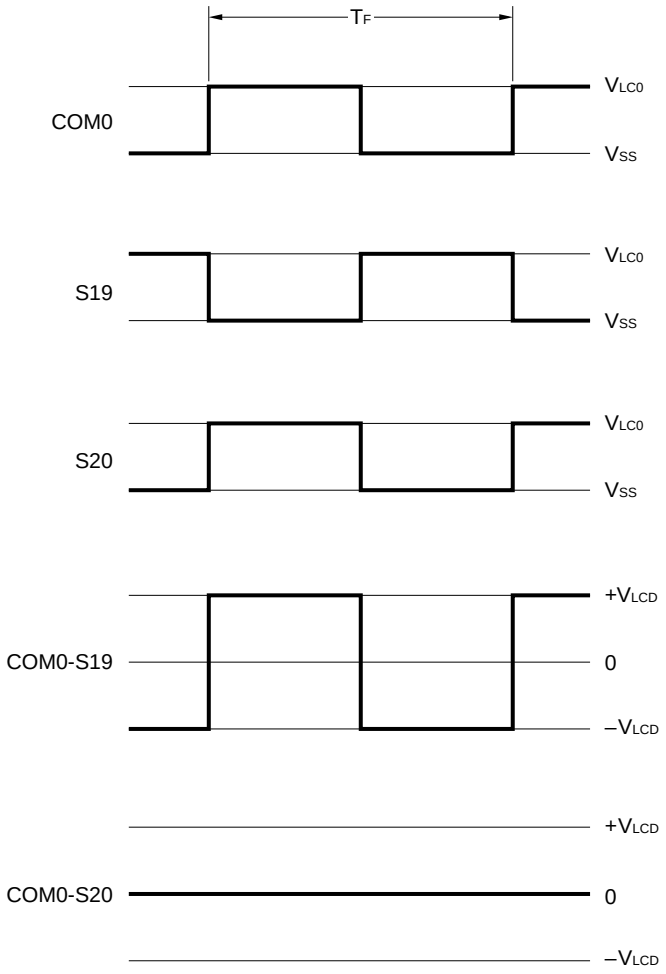


Figure 15-13. Static LCD Drive Waveform Examples



15.8.2 2-Time-Division Display Example

Figure 15-15 shows the connection of a 2-time-division type 10-digit LCD panel with the display pattern shown in Figure 15-14 with the μPD78064B subseries segment signals (S0 to S39) and common signals (COM0, COM1). The display example is "123456.7890," and the display data memory contents (addresses FA58H to FA7FH) correspond to this.

An explanation is given here taking the example of the eighth digit "3" (3). In accordance with the display pattern in Figure 15-14, selection and non-selection voltages must be output to pins S28 through S31 as shown in Table 15-8 at the COM0 and COM1 common signal timings.

Table 15-8. Selection and Non-Selection Voltages (COM0, COM1)

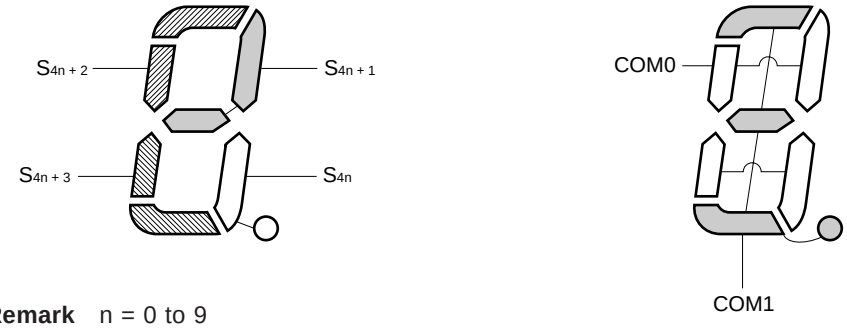
Segment / Common	S28	S29	S30	S31
COM0	S	S	NS	NS
COM1	NS	S	S	S

S: Selection, NS: Non-selection

From this, it can be seen that, for example, ××10 must be prepared in the display data memory (address FA60H) corresponding to S31.

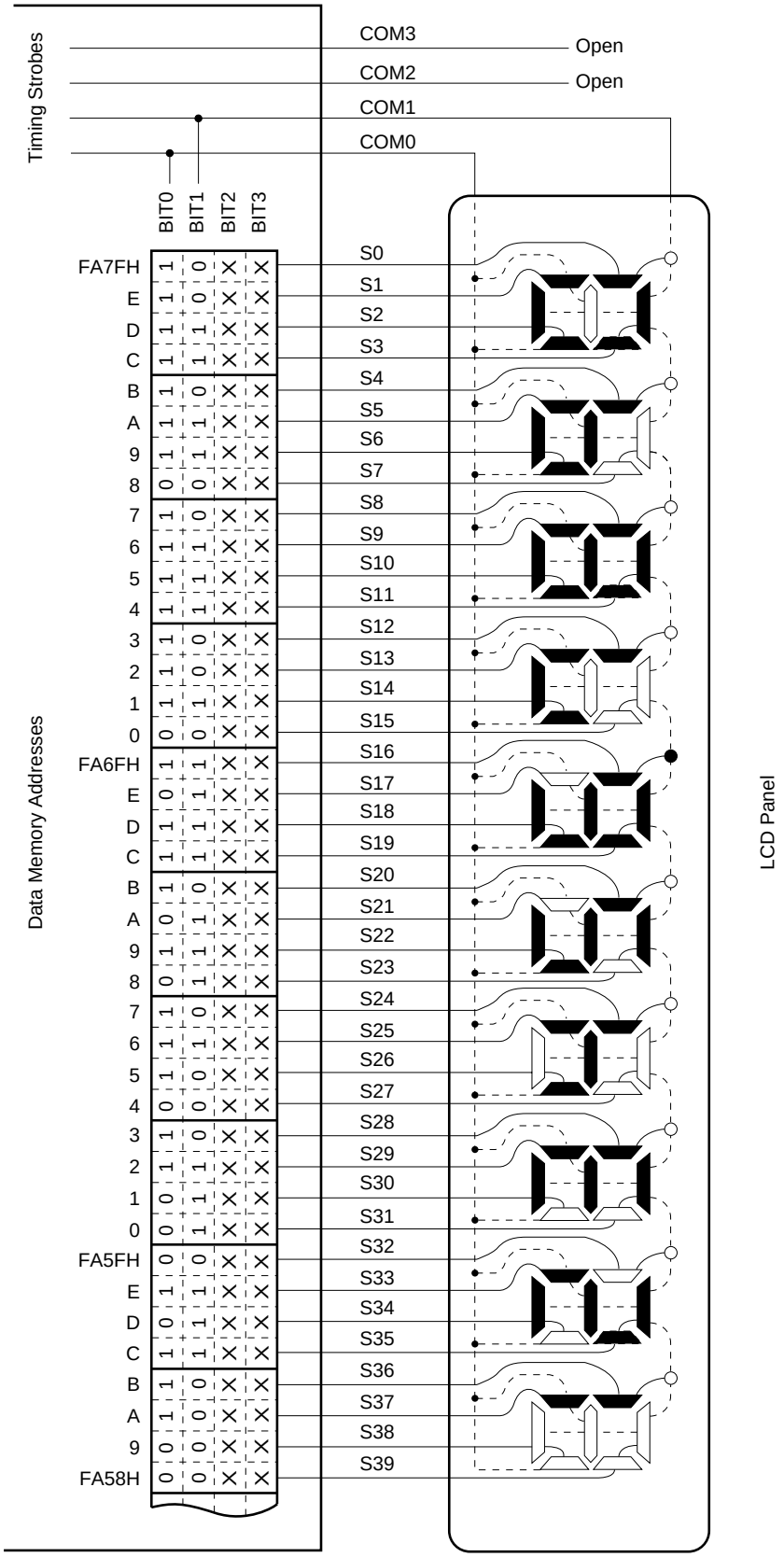
Examples of the LCD drive waveforms between S31 and the common signals are shown in Figure 15-16. When S31 is at the selection voltage at the COM1 selection timing, it can be seen that the $+V_{LCD}/-V_{LCD}$ AC square wave, which is the LCD illumination (ON) level, is generated.

Figure 15-14. 2-Time-Division LCD Display Pattern and Electrode Connections



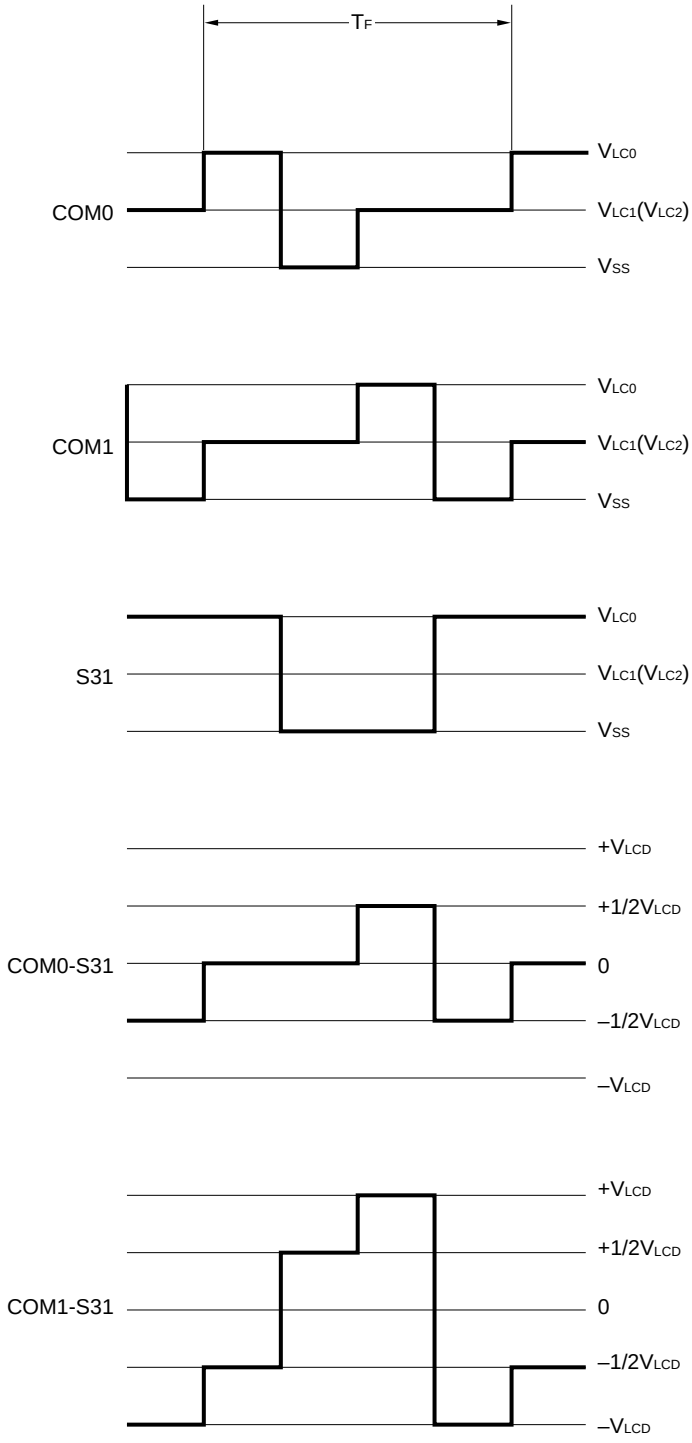
Remark n = 0 to 9

Figure 15-15. 2-Time-Division LCD Panel Connection Example



Remark In bits marked ×, 0 or 1 may be stored because this is a 2-time-division display.

Figure 15-16. 2-Time-Division LCD Drive Waveform Examples (1/2 Bias Method)



15.8.3 3-Time-Division Display Example

Figure 15-18 shows the connection of a 3-time-division type 13-digit LCD panel with the display pattern shown in Figure 15-17 with the μPD78064B subseries segment signals (S0 to S38) and common signals (COM0 to COM2). The display example is "123456.7890123," and the display data memory contents (addresses FA59H to FA7FH) correspond to this.

An explanation is given here taking the example of the eighth digit "6." (6.). In accordance with the display pattern in Figure 15-17, selection and non-selection voltages must be output to pins S21 through S23 as shown in Table 15-9 at the COM0 to COM2 common signal timings.

Table 15-9. Selection and Non-Selection Voltages (COM0 to COM2)

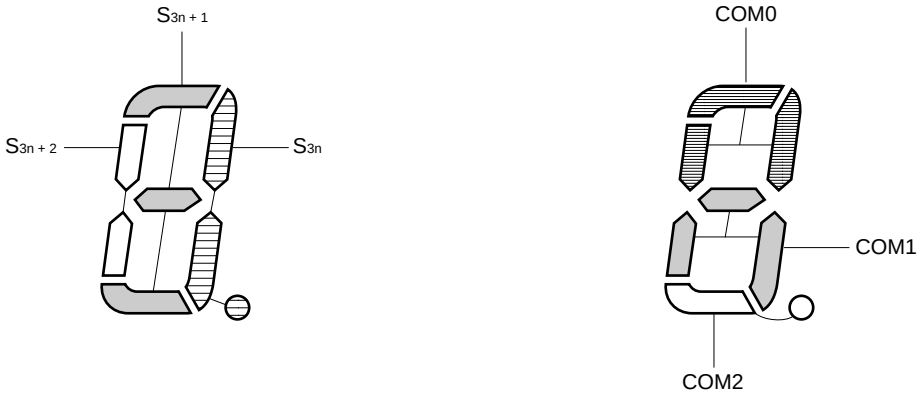
Segment / Common	S21	S22	S23
COM0	NS	S	S
COM1	S	S	S
COM2	S	S	—

S: Selection, NS: Non-selection

From this, it can be seen that ×110 must be prepared in the display data memory (address FA6AH) corresponding to S21.

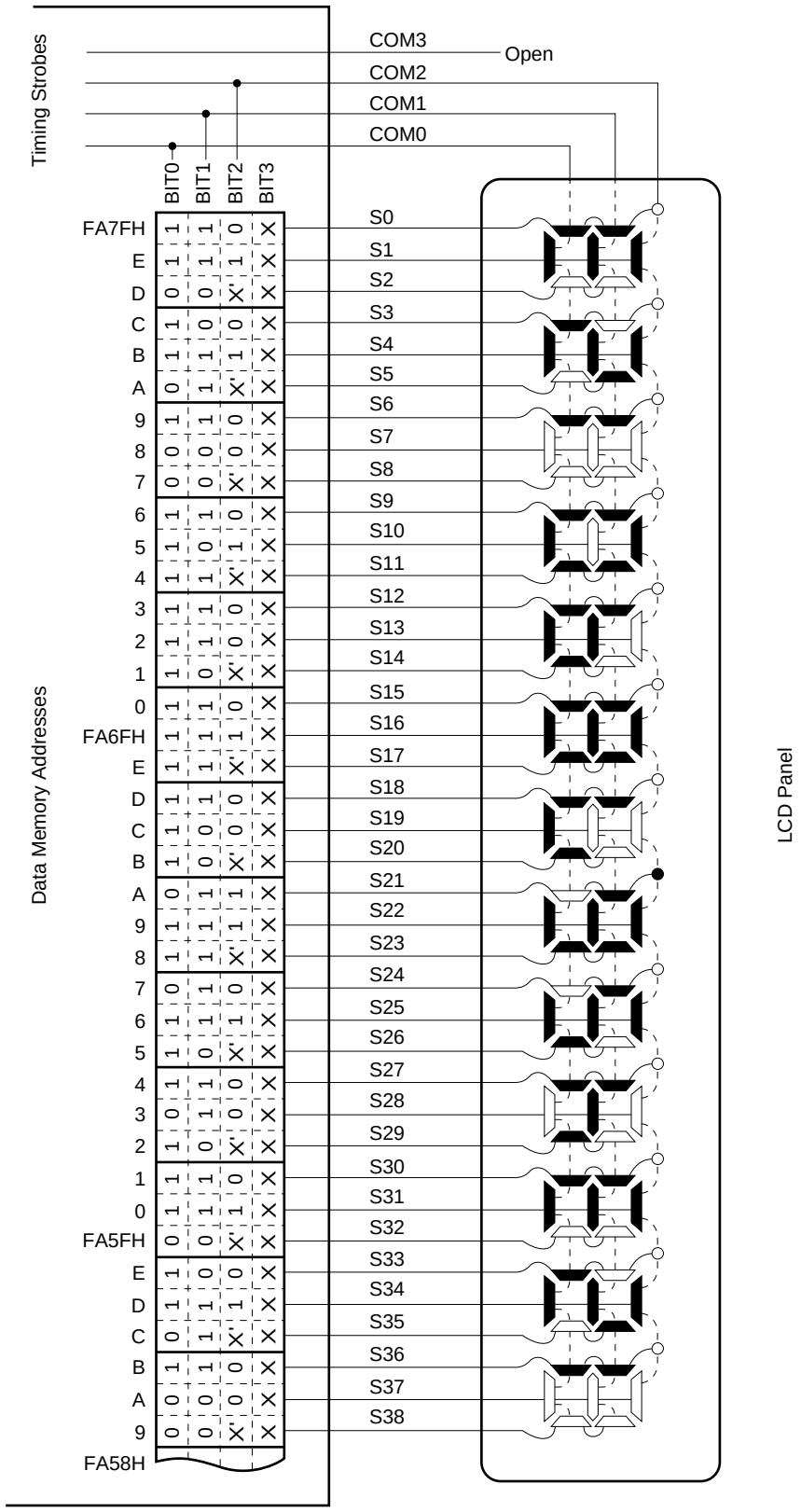
Examples of the LCD drive waveforms between S21 and the common signals are shown in Figure 15-19 (1/2 bias method) and Figure 15-20 (1/3 bias method). When S21 is at the selection voltage at the COM1 selection timing, and S21 is at the selection voltage at the COM2 selection timing, it can be seen that the $+V_{LCD}/-V_{LCD}$ AC square wave, which is the LCD illumination (ON) level, is generated.

Figure 15-17. 3-Time-Division LCD Display Pattern and Electrode Connections



Remark n = 0 to 12

Figure 15-18. 3-Time-Division LCD Panel Connection Example



Remarks 1. ×' : Irrelevant bits because they have no corresponding segment in the LCD panel

2. × : Irrelevant bits because this is a 3-time-division display

Figure 15-19. 3-Time-Division LCD Drive Waveform Examples (1/2 Bias Method)

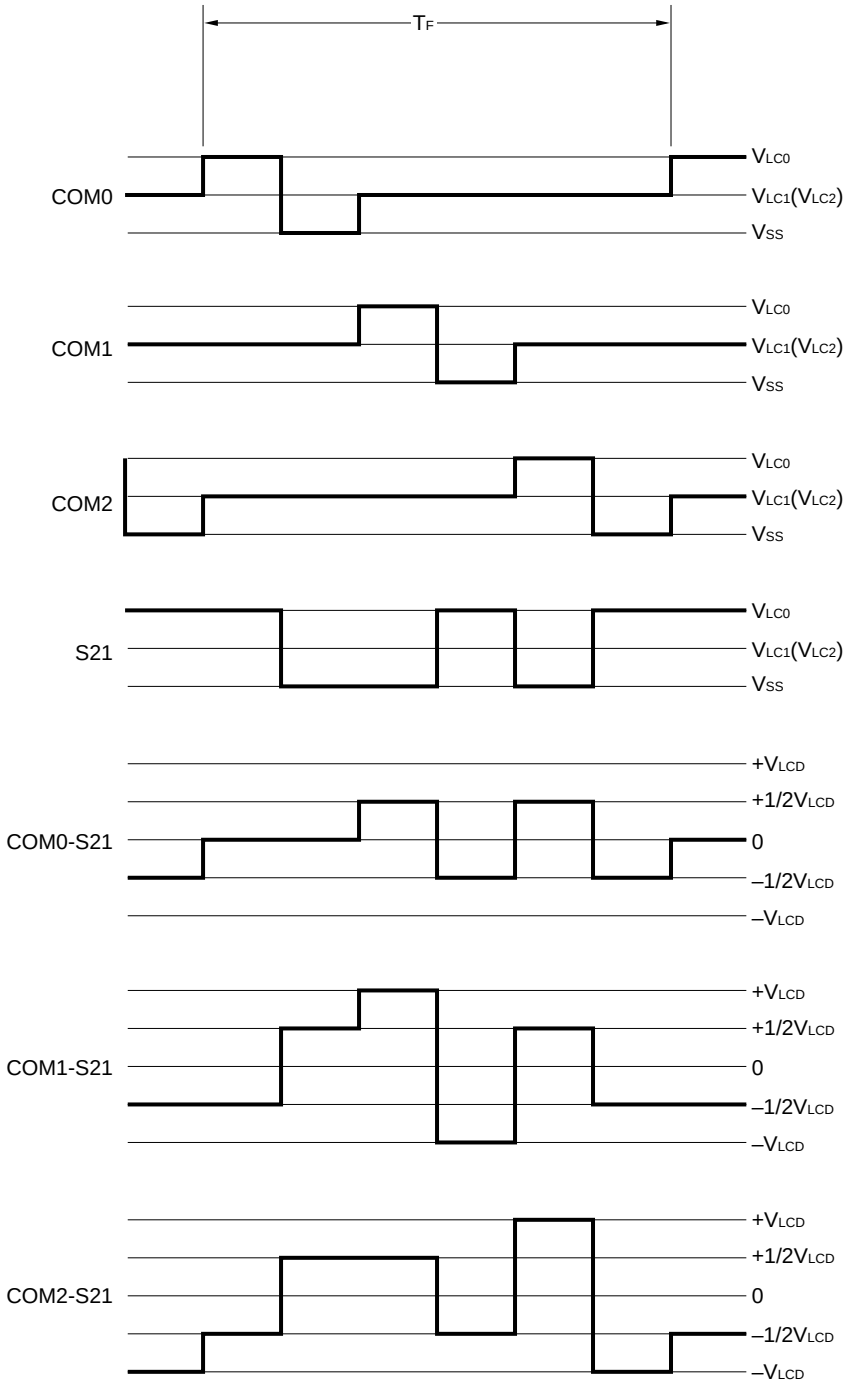
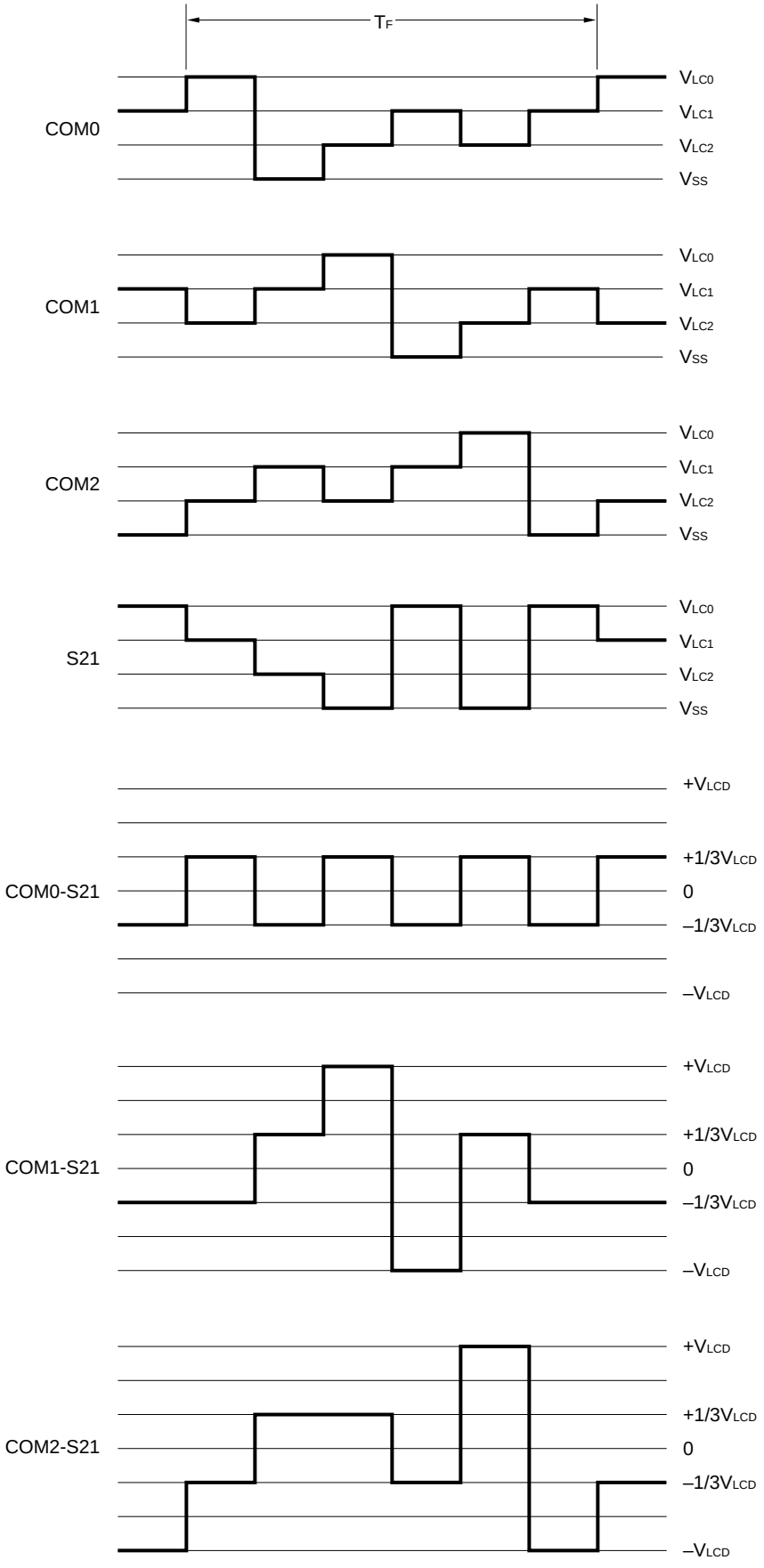


Figure 15-20. 3-Time-Division LCD Drive Waveform Examples (1/3 Bias Method)



15.8.4 4-Time-Division Display Example

Figure 15-22 shows the connection of a 4-time-division type 20-digit LCD panel with the display pattern shown in Figure 15-21 with the μPD78064B subseries segment signals (S0 to S39) and common signals (COM0 to COM3). The display example is "123456.78901234567890," and the display data memory contents (addresses FA58H to FA7FH) correspond to this.

An explanation is given here taking the example of the 15th digit "6." (ξ.). In accordance with the display pattern in Figure 15-21, selection and non-selection voltages must be output to pins S28 and S29 as shown in Table 15-10 at the COM0 to COM3 common signal timings.

Table 15-10. Selection and Non-Selection Voltages (COM0 to COM3)

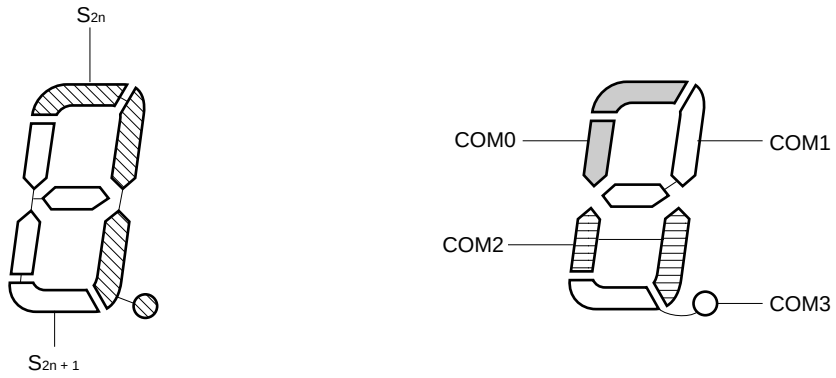
Segment / Common	S28	S29
COM0	S	S
COM1	NS	S
COM2	S	S
COM3	S	S

S: Selection, NS: Non-selection

From this, it can be seen that 1101 must be prepared in the display data memory (address FA63H) corresponding to S28.

Examples of the LCD drive waveforms between S28 and the COM0 and COM1 signals are shown in Figure 15-23 (for the sake of simplicity, waveforms for COM2 and COM3 have been omitted). When S28 is at the selection voltage at the COM0 selection timing, it can be seen that the $+V_{LCD}/-V_{LCD}$ AC square wave, which is the LCD illumination (ON) level, is generated.

Figure 15-21. 4-Time-Division LCD Display Pattern and Electrode Connections



Remark n = 0 to 18

Figure 15-22. 4-Time-Division LCD Panel Connection Example

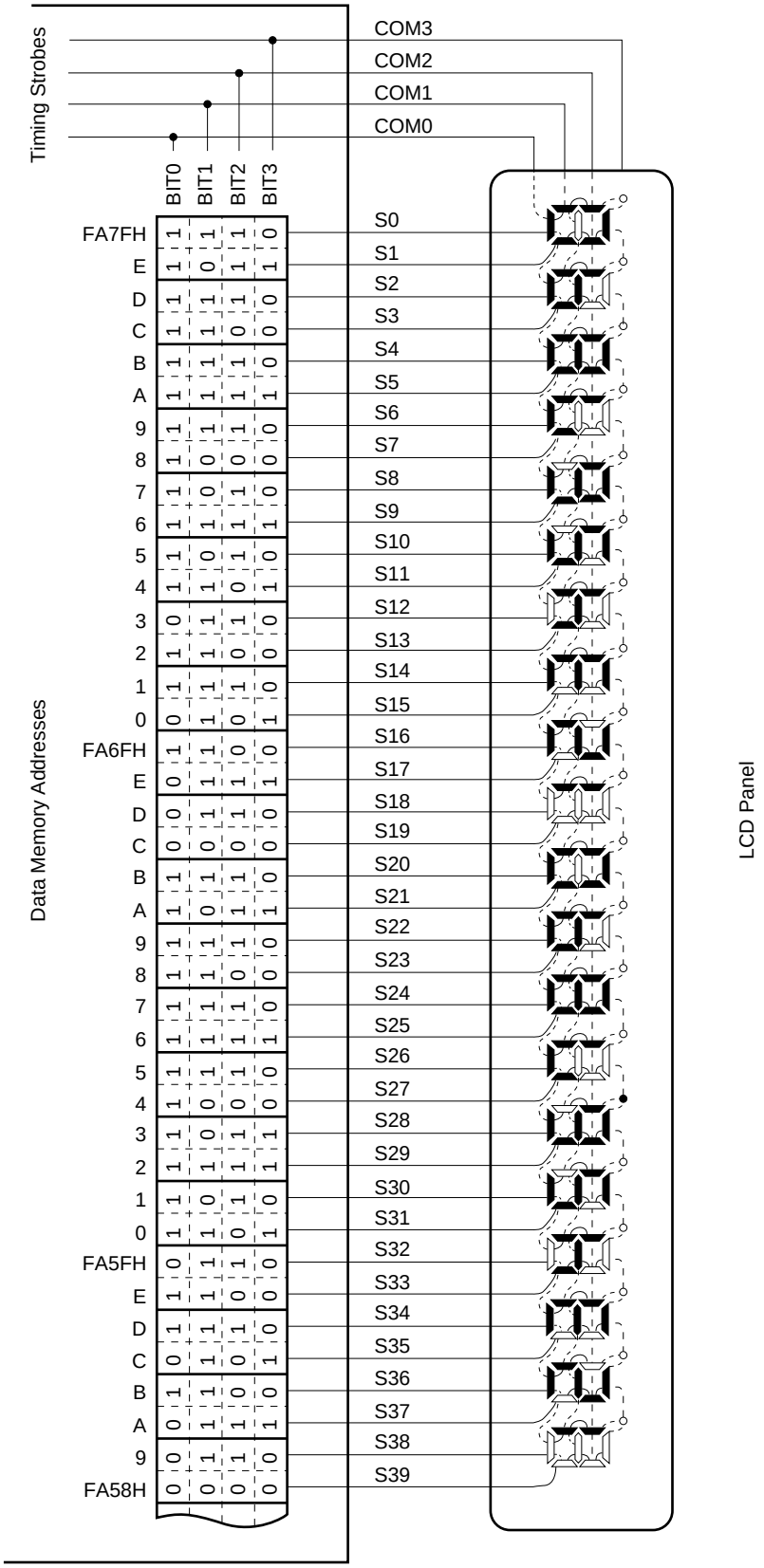
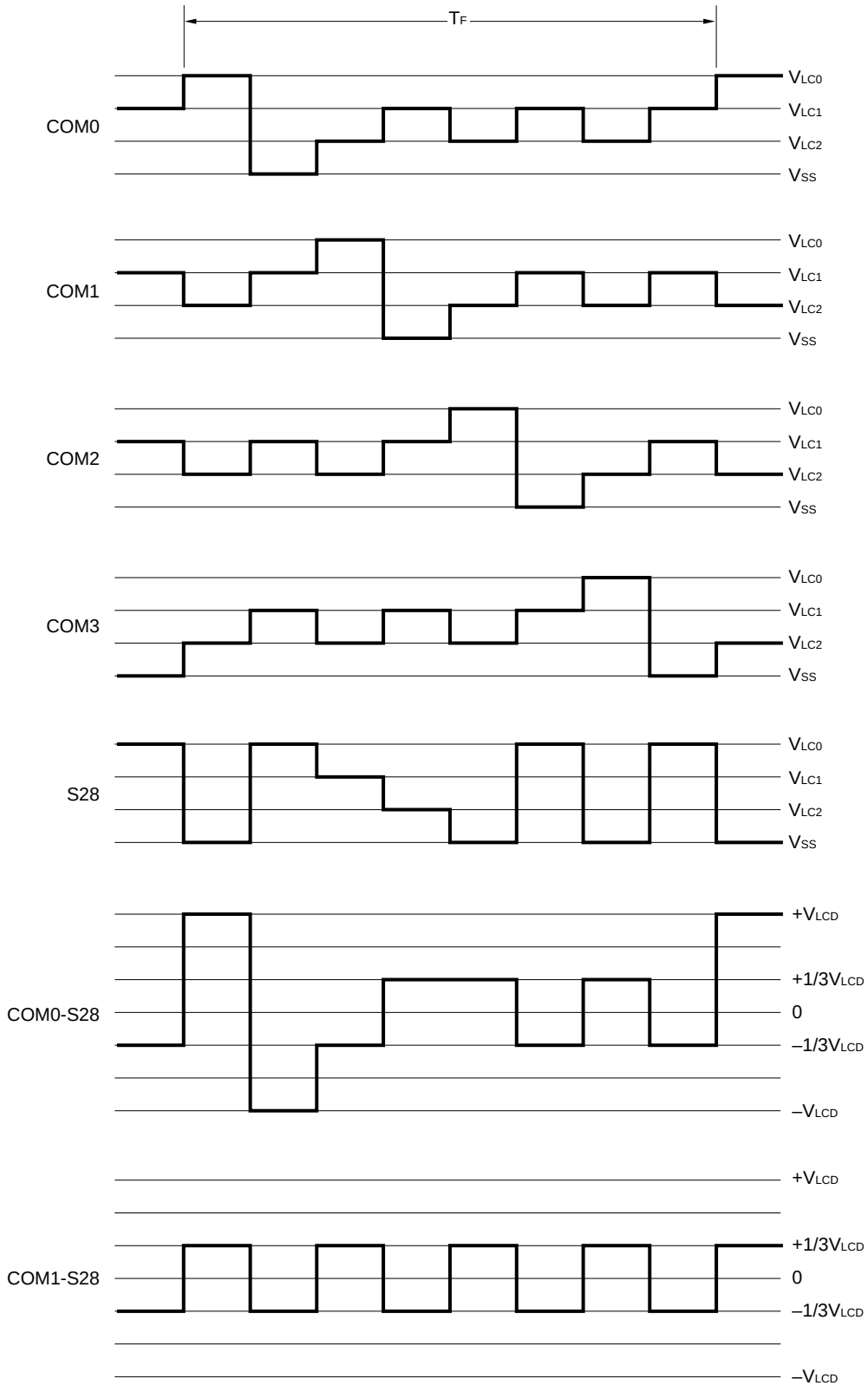


Figure 15-23. 4-Time-Division LCD Drive Waveform Examples (1/3 Bias Method)



[MEMO]

CHAPTER 16 INTERRUPT AND TEST FUNCTIONS

16.1 Interrupt Function Types

The following three types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally (that is, even in interrupt disabled state). It does not undergo interrupt priority control and is given top priority over all other interrupt requests.
It generates a standby release signal.
One interrupt source from the watchdog timer is incorporated as a non-maskable interrupt request.

(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 (refer to **Table 16-1**).
A standby release signal is generated.
Six external interrupt sources and 15 internal interrupt sources are incorporated as maskable interrupt requests.

(3) Software interrupt

This is a vectored interrupt to be generated by executing the BRK instruction. It is acknowledged even in interrupt disabled state. The software interrupt does not undergo interrupt priority control.

16.2 Interrupt Sources and Configuration

Twenty non-maskable, maskable, and software interrupts are provided as interrupt sources (refer to **Table 16-1**).

Table 16-1. Interrupt Source List

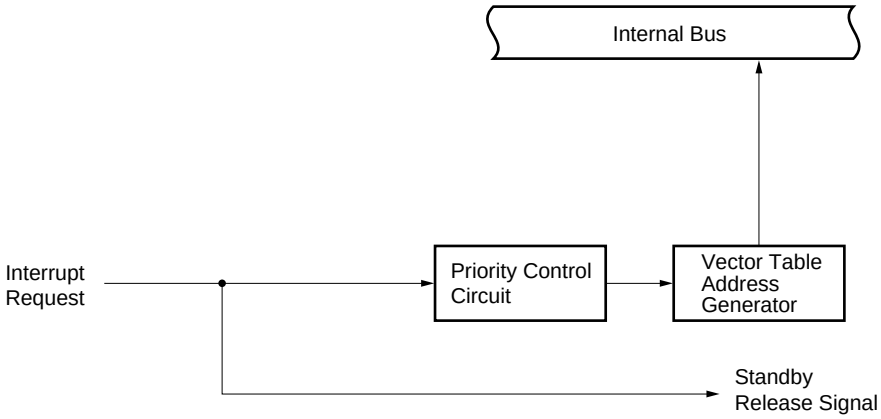
Maskability	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Non-Maskable	—	INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (with interval timer 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	INTSER	Serial interface channel 2 UART reception error generation		0018H	
	9	INTSR	End of serial interface channel 2 UART reception		001AH	
		INTCSI2	End of serial interface channel 2 3-wire transfer			
	10	INTST	End of serial interface channel 2 UART transfer		001CH	
	11	INTTM3	Reference time interval signal from watch timer		001EH	
	12	INTTM00	Generation of 16-bit timer register, capture/compare register (CR00) match signal		0020H	
	13	INTTM01	Generation of 16-bit timer register, capture/compare register (CR01) match signal		0022H	
	14	INTTM1	Generation of 8-bit timer/event counter 1 match signal		0024H	
	15	INTTM2	Generation of 8 bit timer/event counter 2 match signal		0026H	
	16	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 16 is the lowest priority.

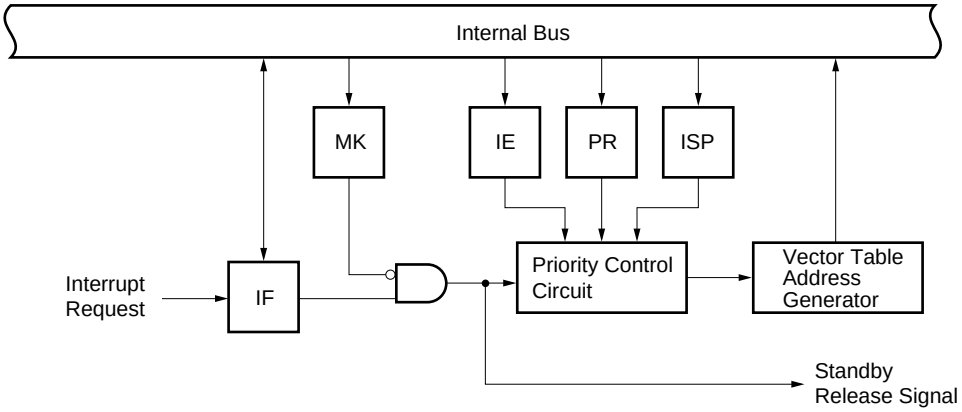
2. Basic configuration types (A) to (E) correspond to (A) to (E) of Figure 16-1.

Figure 16-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt (INTP0)

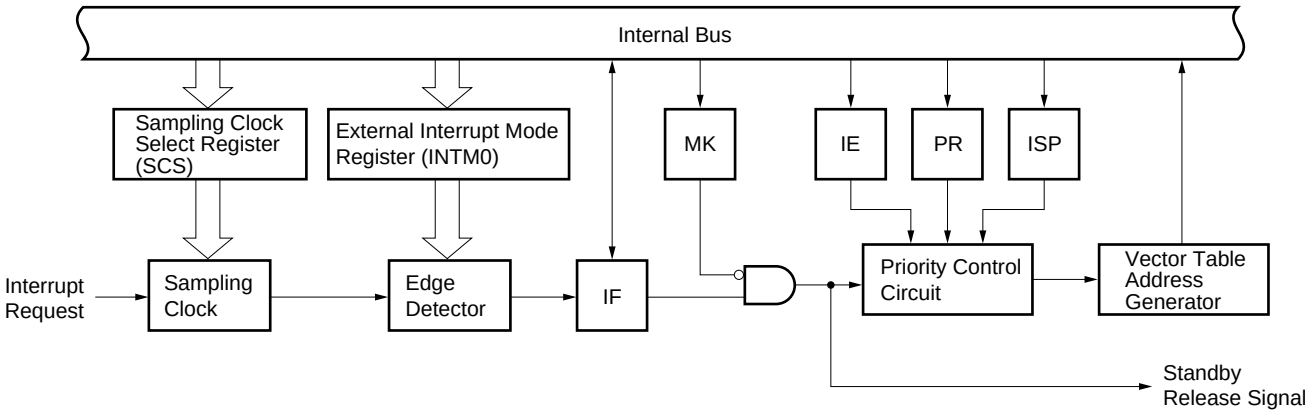
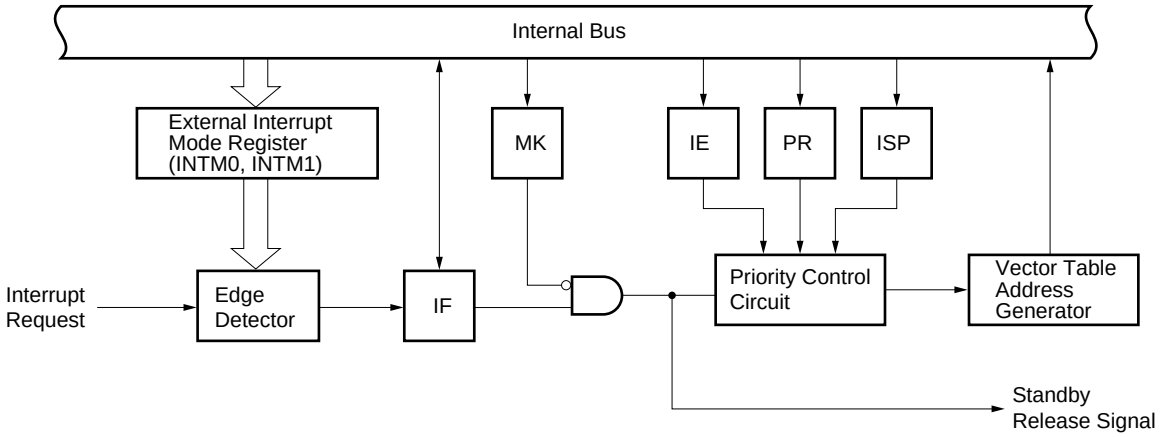
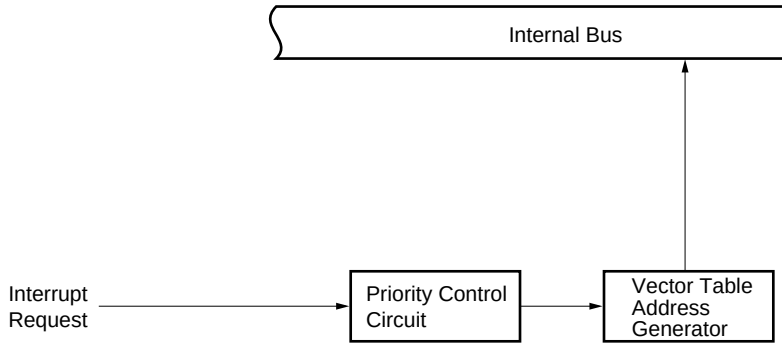


Figure 16-1. Basic Configuration of Interrupt Function (2/2)

(D) External maskable interrupt (except INTP0)



(E) Software interrupt



IF : Interrupt request flag
IE : Interrupt enable flag
ISP : Inservice priority flag
MK : Interrupt mask flag
PR : Priority specify flag

16.3 Interrupt Function Control Registers

The following six types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H, 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 16-2 gives a listing of interrupt request flags, interrupt mask flags, and priority specify flags corresponding to interrupt request sources.

Table 16-2. Various Flags Corresponding to Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specify Flag	
		Register		Register		Register
INTWDT	TMIF4	IF0L	TMMK4	MK0L	TMPR4	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTP4	PIF4		PMK4		PPR4	
INTP5	PIF5		PMK5		PPR5	
INTCSI0	CSIIF0	IF0H	CSIMK0	MK0H	CSIPR0	PR0H
INTSER	SERIF		SERMK		SERPR	
INTSR/INTCSI2	SRIF		SRMK		SRPR	
INTST	STIF		STMK		STPR	
INTTM3	TMIF3		TMMK3		TMPR3	
INTTM00	TMIF00		TMMK00		TMPR00	
INTTM01	TMIF01		TMMK01		TMPR01	
INTTM1	TMIF1	IF1L	TMMK1	MK1L	TMPR1	PR1L
INTTM2	TMIF2		TMMK2		TMPR2	
INTAD	ADIF		ADMK		ADPR	

(1) Interrupt request flag registers (IF0L, IF0H, IF1L)

The interrupt request flag is set to 1 when the corresponding interrupt request is generated or an instruction is executed. It is cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon application of $\overline{\text{RESET}}$ input.

IF0L, IF0H, and IF1L are set with a 1-bit or 8-bit memory manipulation instruction. If IF0L and IF0H are used as a 16-bit register IF0 use a 16-bit memory manipulation instruction for the setting.

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

Figure 16-2. 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	0	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 in IF1L bits 3 through 6, IF0L bit 7, and IF0H bit 1.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L)

The interrupt mask flag is used to enable/disable the corresponding maskable interrupt service and to set standby clear enable/disable.

MK0L, MK0H, and MK1L are set with a 1-bit or 8-bit memory manipulation instruction. If 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 16-3. 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	1	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

xxMKx	Interrupt Service Control
0	Interrupt service enabled
1	Interrupt service disabled

Note WTMK controls standby mode release enable/disable. It does not control interrupt functions.

Cautions 1. If TMMK4 flag is read when a watchdog timer is used in watchdog timer mode 1, MK0 value becomes undefined.

2. Because port 0 has a dual function as the external interrupt request input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, 1 should be set in the interrupt mask flag before using the output mode.

3. Set always 1 in MK1L bits 3 through 6, MK0L bit 7, and MK0H bit 1.

(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 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 16-4. 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	1	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

xxPRx	Priority Level Selection
0	High priority level
1	Low priority level

Cautions 1. When a watchdog timer is used in watchdog timer mode 1, set 1 in TMPR4 flag.

2. Set always 1 in PR1L bits 3 through 7, PR0L bit 7, and PR0H bit 1.

(4) External interrupt mode register (INTM0, INTM1)

These registers set the valid edge for INTP0 to INTP5.

INTM0 and INTM1 are set by 8-bit memory manipulation instructions.

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

Figure 16-5. 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 Set the valid edge of the INTP0/TI0/P00 pin after setting bits 1 through 3 (TMC01 through TMC03) of the 16-bit timer mode control register to 0, 0, 0, and stopping the timer operation.**

Figure 16-6. 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 removed with sampling clocks.

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

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

Figure 16-7. 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 : Oscillation mode selection register (OSMS) bit 0

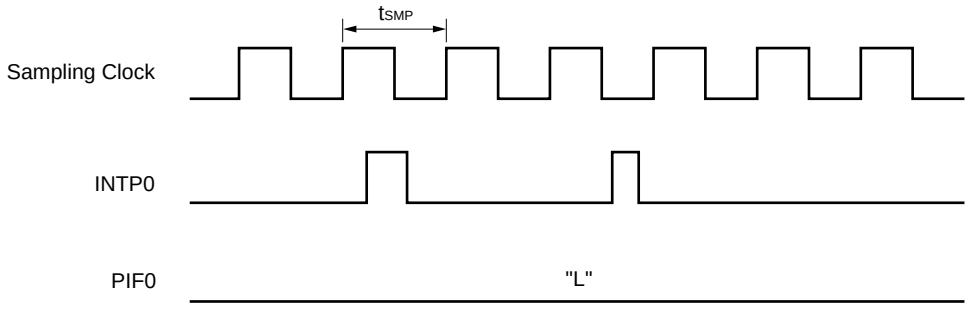
5. Values in parentheses when operated with f_X = 5.0 MHz.

When the sampled INTP0 input level is active twice in succession, the noise remover sets interrupt request flag (PIF0) flag to 1.

Figure 16-8 shows the noise remover input/output timing.

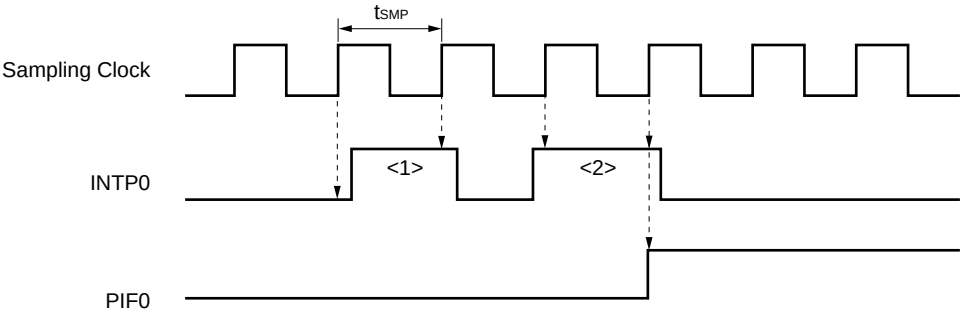
Figure 16-8. Noise Remover Input/Output Timing (during rising edge detection)

(a) When input is less than the sampling cycle (t_{SMP})



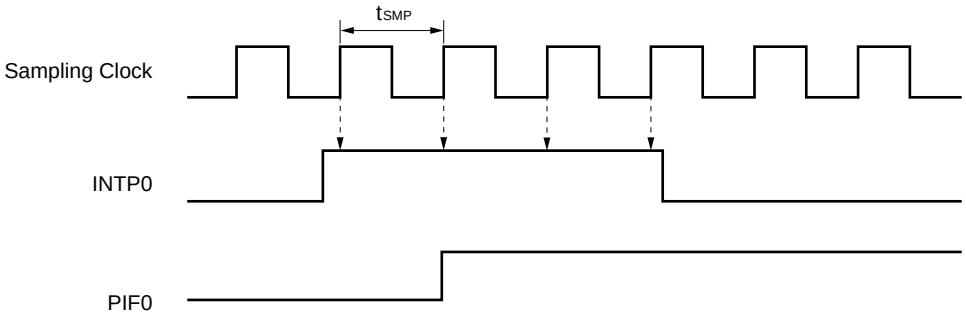
Because INTP0 level is not high during sampling,
PIF0 output remains at low level.

(b) When input is equal to or twice the sampling cycle (t_{SMP})



Because the sampled INTP0 level is high twice in succession in <2>,
PIF0 flag is set to 1.

(c) When input is twice or more than the cycle frequency (t_{SMP})



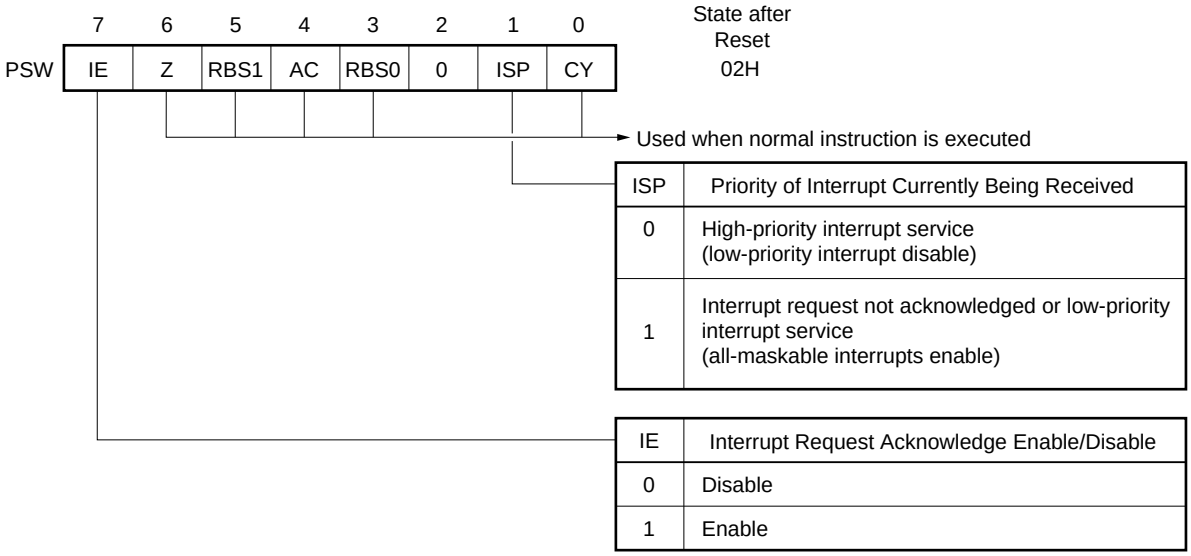
At the point when INTP0 level is high twice in succession,
PIF0 flag is set to 1.

(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 nesting 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 the BRK instruction is executed, PSW is automatically saved into a stack and the IE flag is reset to 0. If a maskable interrupt request is acknowledged contents of the priority specify flag of the acknowledged interrupt are transferred to the ISP flag. The acknowledged interrupt is also saved into the stack with the PUSH PSW instruction. It is restored from the stack with the RETI, RETB, and POP PSW instructions.

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

Figure 16-9. Program Status Word Format



ISP	Priority of Interrupt Currently Being Received
0	High-priority interrupt service (low-priority interrupt disable)
1	Interrupt request not acknowledged or low-priority interrupt service (all-maskable interrupts enable)

IE	Interrupt Request Acknowledge Enable/Disable
0	Disable
1	Enable

16.4 Interrupt Service Operations

16.4.1 Non-maskable interrupt request acknowledge operation

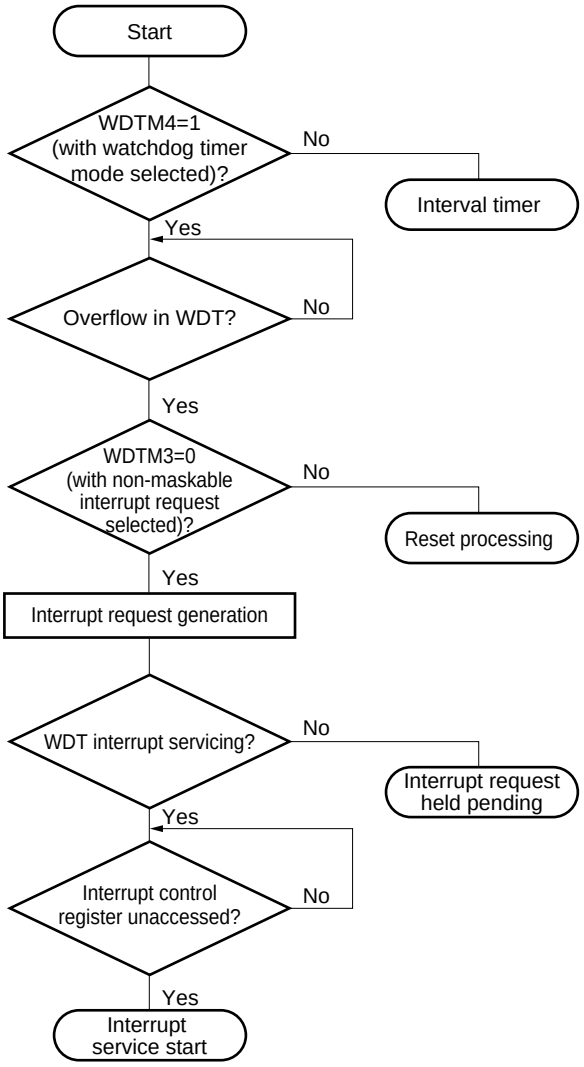
A non-maskable interrupt request is unconditionally acknowledged even if in an interrupt request acknowledge disable state. It does not undergo interrupt priority control and has highest priority over all other interrupts.

If a non-maskable interrupt request is acknowledged, the acknowledged interrupt is saved in the stacks, program status word (PSW) and program counter (PC), in that order, the IE and ISP flags are reset to 0, and the vector table contents are loaded into PC and branched.

A new non-maskable interrupt request generated during execution of a non-maskable interrupt service program is acknowledged after the current execution of the non-maskable interrupt service program is terminated (following RETI instruction execution) and one main routine instruction is executed. If a new non-maskable interrupt request is generated twice or more during non-maskable interrupt service program execution, only one non-maskable interrupt request is acknowledged after termination of the non-maskable interrupt service program execution.

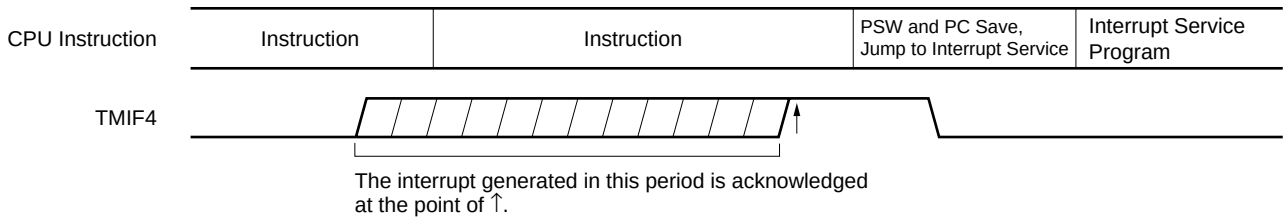
Figure 16-10 shows the flowchart from generation of the non-maskable interrupt to accepting it. Figure 16-12 shows the timing of accepting the non-maskable interrupt request, and Figure 16-12 shows the operation performed if the non-maskable interrupt request occurs in duplicate.

Figure 16-10. Flowchart of Non-Maskable Interrupt Request from Generation to Acknowledge



WDTM : Watchdog timer mode register
WDT : Watchdog timer

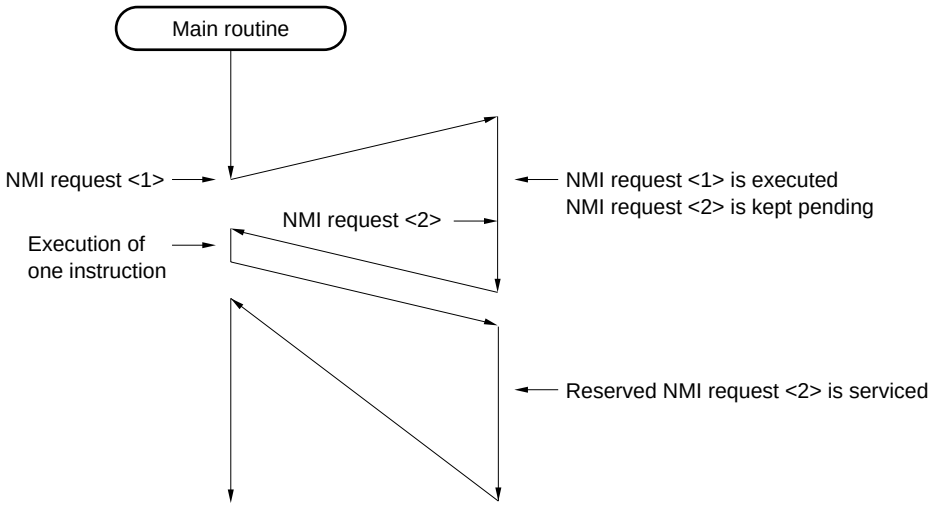
Figure 16-11. Non-Maskable Interrupt Request Acknowledge Timing



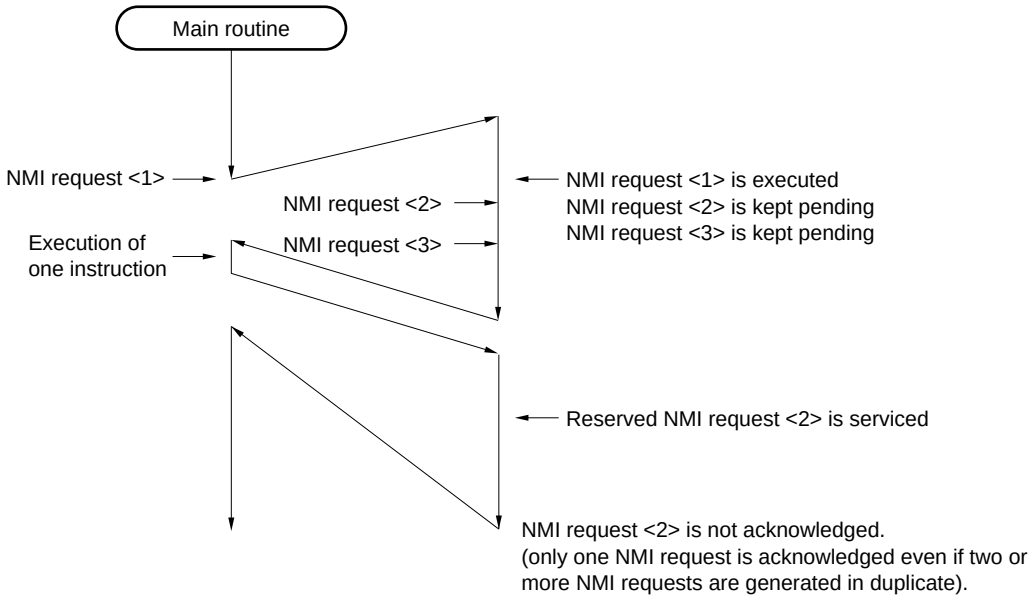
TMIF4 : Watchdog timer interrupt request flag

Figure 16-12. 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



16.4.2 Maskable interrupt request acknowledge operation

A maskable interrupt request becomes acknowledgeable when an interrupt request flag is set to 1 and the interrupt request mask (MK) flag flag is cleared to 0. A vectored interrupt request is acknowledged in an interrupt enable state (with IE flag set to 1). However, a low-priority interrupt request is not acknowledged during high-priority interrupt request service (with ISP flag reset to 0).

Wait times maskable interrupt request generation to interrupt service are as follows.

For the acknowledge timing of interrupt request, refer to **Figures 16-13** and **16-14**.

Table 16-3. Times from Maskable Interrupt Request Generation to Interrupt Service

	Minimum Time	Maximum Time **Note**
When ××PR× = 0	7 clock cycles	32 clock cycles
When ××PR× = 1	8 clock cycles	33 clock cycles

Note If an interrupt request is generated just before a divide instruction, the wait time is maximized.

Remark 1 clock cycle = 1/CPU clock frequency (f_{CPU})

If two or more maskable interrupt requests are generated simultaneously, the request specified for higher priority with the priority specify flag is acknowledged first. Two or more requests specified for the same priority with the priority specify flag, the default priorities apply.

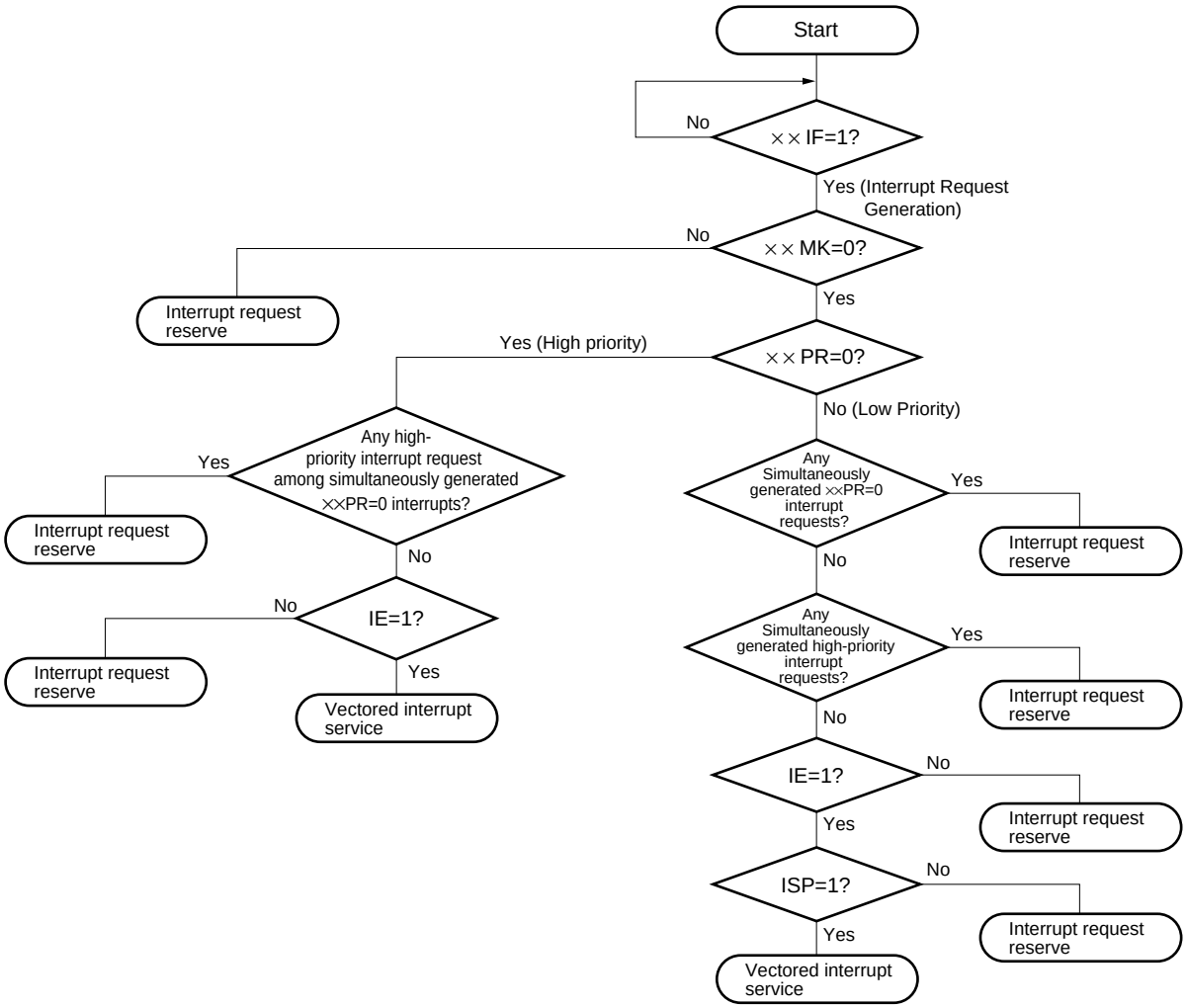
Any interrupt requests that kept pending are acknowledged when they become acknowledgeable.

Figure 16-13 shows interrupt request acknowledge algorithms.

If a maskable interrupt request is acknowledged, the acknowledged interrupt is 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 request 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.

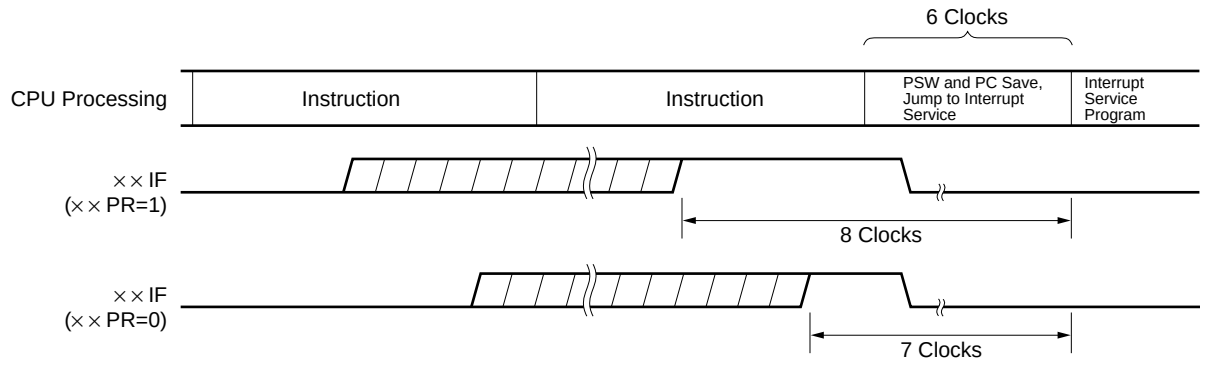
Restore from the interrupt is possible with the RETI instruction.

Figure 16-13. Interrupt Request Acknowledge Processing Algorithm



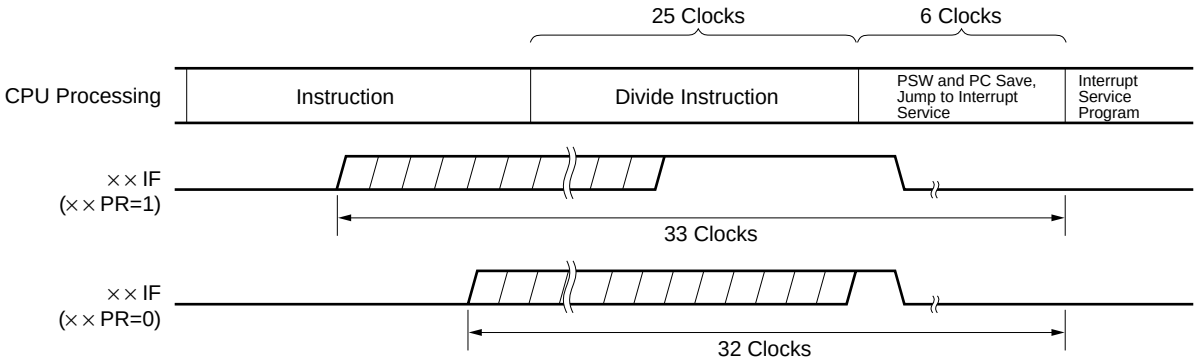
××IF : Interrupt request flag
××MK : Interrupt mask flag
××PR : Priority specification flag
IE : Flag controlling accepting maskable interrupt request (1 = enable, 0 = disable)
ISP : Flag indicating priority of interrupt currently serviced (0 = interrupt with high priority serviced, 1 = interrupt request is not accepted, or interrupt with low priority is serviced)

Figure 16-14. Interrupt Request Acknowledge Timing (Minimum Time)



Remark 1 clock cycle = 1/CPU clock frequency (f_{CPU})

Figure 16-15. Interrupt Request Acknowledge Timing (Maximum Time)



Remark 1 clock cycle = 1/f_{CPU} clock frequency (f_{CPU}: CPU clock)

16.4.3 Software interrupt request acknowledge operation

A software interrupt request is acknowledged by BRK instruction execution. Software interrupt cannot be disabled.

If a software interrupt request is acknowledged, it is saved in the stacks, 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.

Restore from the software interrupt is possible with the RETB instruction.

Caution Do not use the RETI instruction for returning from the software interrupt.

16.4.4 Nesting interrupt service

Acknowledgment of another interrupt request while an interrupt is being serveced is called nesting.

Nesting does not take place unless the interrupts (except the non-maskable interrupt) are enabled to be acknowledged (IE = 1). Acknowledgment of another interrupt request is disabled (IE = 0) when one interrupt has been acknowledged. Therefore, to enable nesting, the EI flag must be set to 1 during interrupt servicing, to enable another interrupt.

Nesting may not occur even when the interrupts are enabled. This is controlled but the priorities of the interrupts. Although two types of priorities, default priority and programmable priority, may be assigned to an interrupt, nesting is controlled by using the programmable priority.

If an interrupt with the same level of priority as or higher priority than the interrupt currently serviced occurs, that interrupt can be acknowledged and nested. If an interrupt with a priority lower than that of the currently serviced interrupt occurs, that interrupt cannot be acknowledged nor nested.

An interrupt that is not acknowledged and nested because of it is disabled or it has a low priority is kept reserved. This interrupt is acknowledged after servicing og the current interrupt has been completed and one instruction of the main routine has been executed.

Nesting is not enabled while the non-maskable interrupt is being serviced.

Table 16-4 shows the interrupts that can be nested, and Figure 16-16 shows an example of nesting.

Table 16-4. Interrupt Request Enabled for Nesting Interrupt during Interrupt Service

Nesting Interrupt Request / Interrupt being serviced		Non-maskable Interrupt Request	Maskable Interrupt Request			
			PR=0		PR=1	
			IE=1	IE=0	IE=1	IE=0
Non-maskable interrupt		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 : Nesting interrupt enable

2. D : Nesting interrupt disable

3. ISP and IE are the flags contained in PSW

ISP=0 : An interrupt with higher priority is being serviced

ISP=1 : An interrupt is not accepted or an interrupt with lower priority is being serviced

IE=0 : Interrupt acknowledge is disabled

IE=1 : Interrupt acknowledge is enabled

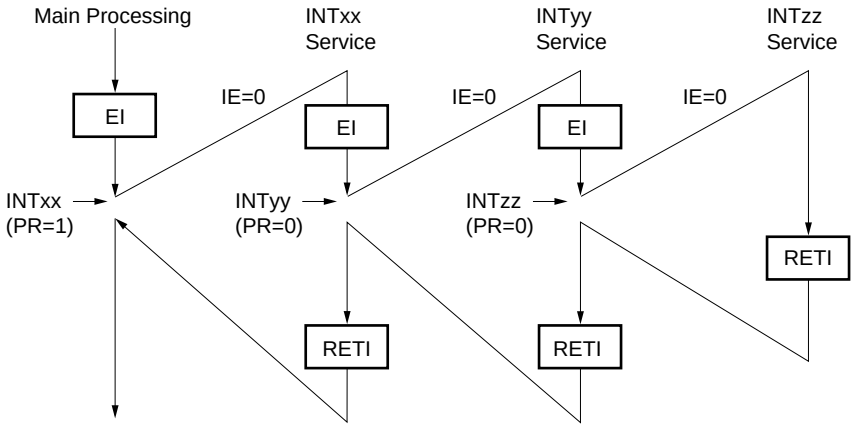
4. PR is a flag contained in PR0L, PR0H, PR1L

PR=0 : Higher priority level

PR=1 : Lower priority level

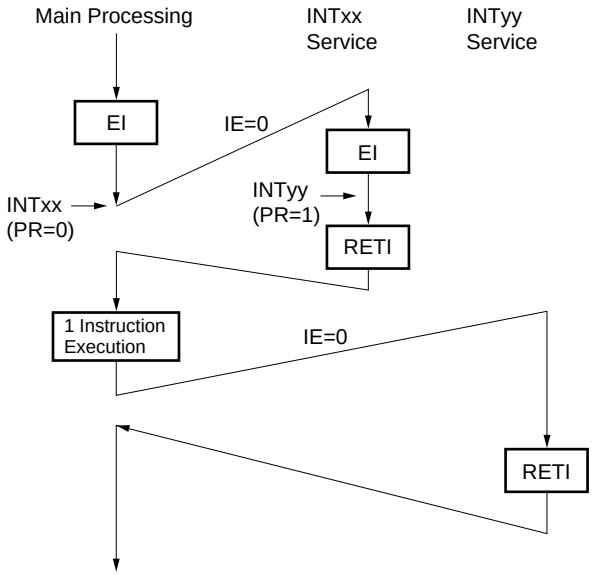
Figure 16-16. Nesting Interrupt Example (1/2)

Example 1. Example where nesting interrupt takes place two times



Two interrupt requests, INTyy and INTzz, are acknowledged while interrupt INTxx is serviced, and nesting takes place. Before each interrupt requests is acknowledged, the IE instruction is always executed, and the interrupt is enabled.

Example 2. Example where nesting interrupt does not take place because of priority control

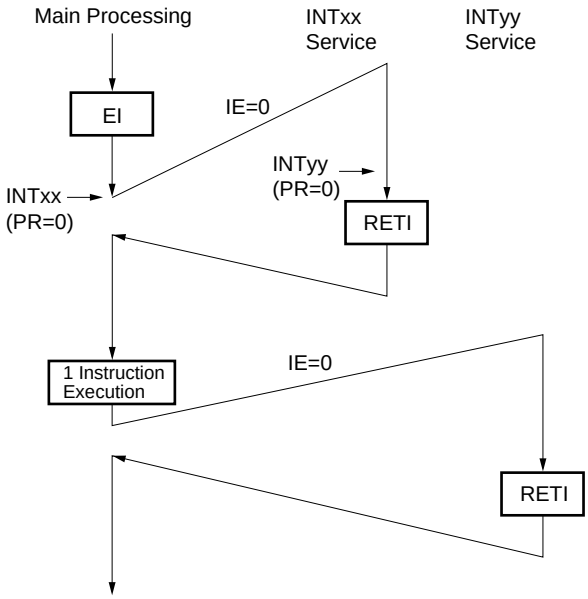


Interrupt request INTyy that is generated while interrupt INTxx is being serviced is not acknowledged because its priority is lower than that of INTxx, and therefore, nesting does not take place. INTyy request is reserved, and is acknowledged after one instruction of the main routine has been executed.

PR = 0: High-priority level
PR = 1: Low-priority level
IE = 0: Acknowledge of interrupt request is disabled.

Figure 16-16. Nesting Interrupt Example (2/2)

Example 3. Eample where nesting interrupt does not take place because interrupts are not enabled



Because interrupts are not enabled (EI instruction is not issued) in interrupt processing INTxx, interrupt request INTyy is not acknowledged, and nesting does not take place. INTyy request is reserved, and is acknowledged after one instruction of the main routine has been executed.

PR = 0: High-priority level
IE = 0: Acknowledge of interrupt request is disabled.

16.4.5 Interrupt request pending

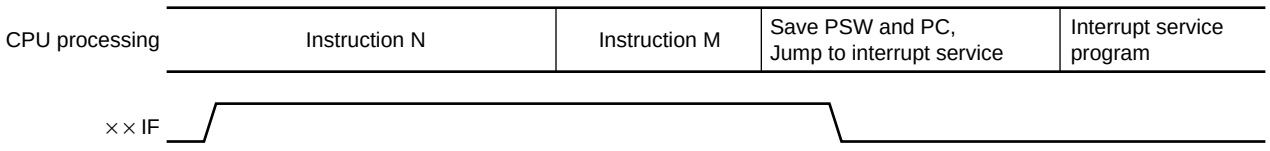
The followings are the instructions which keep interrupt acknowledge pending.

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW.bit, CY
- MOV1 CY, PSW.bit
- AND1 CY, PSW.bit
- OR1 CY, PSW.bit
- XOR1 CY, PSW.bit
- SET1 PSW.bit
- CLR1 PSW.bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW.bit, $addr16
- BF PSW.bit, $addr16
- BTCLR PSW.bit, $addr16
- EI
- DI
- Manipulate instructions for IF0L, IF0H, IF1L, MK0L, MK0H, MK1L, PR0L, PR0H, PR1L, INTM0, INTM1 registers

Caution The BRK instruction is not one of the above instructions that keep an interrupt request pending. However, the software interrupt that is started by execution of the BRK instruction clears the IE flag to 0. Therefore, even if a maskable interrupt request is generated while the BRK instruction is being executed, it is not accepted. However, the non-maskable interrupt is accepted.

Figure 16-17 shows the timing at which an interrupt request is kept pending.

Figure 16-17. Interrupt Request Pending



Remarks 1. Instruction N: Instruction that keeps interrupts requests pending

2. Instruction M: Instructions other than interrupt request pending instruction

3. The ××PR (priority level) values do not affect the operation of ××IF (interrupt request).

16.5 Test Functions

The test function sets the corresponding test input flag and generates a standby release signal when an overflow occurs in the watch timer and when a falling edge at port 4 is detected.

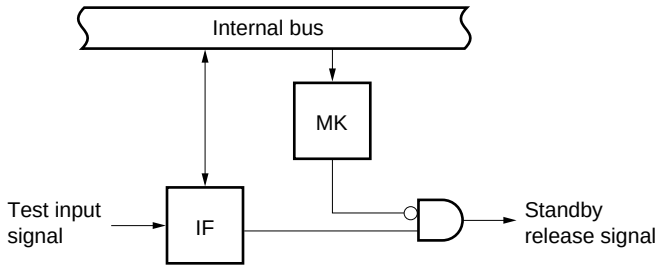
Unlike the interrupt function, this function does not perform vector processing.

There are two test input factors as shown in Table 16-5. The basic configuration is shown in Figure 16-18.

Table 16-5. Test Input Factors

Test Input Factors		Internal/
Name	Trigger	External
INTWT	Watch timer overflow	Internal
INTPT11	Falling edge detection at port 11	External

Figure 16-18. Basic Configuration of Test Function



IF: test input flag

MK: test mask flag

16.5.1 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 16-6.

Table 16-6. Flags Corresponding to Test Input Signals

Test input signal name	Test input flag	Test mask flag
INTWT	WTIF	WTMK
INTPT11	KRIF	KRMK

(1) Interrupt request flag register 1L (IF1L)

It indicates whether a watch timer overflow is detected or not.

It is set by a 1-bit memory manipulation instruction and 8-bit memory manipulation instruction.

It is set to 00H by the $\overline{\text{RESET}}$ signal input.

Figure 16-19. Format of Interrupt Request Flag Register 1L

Symbol	⑦	6	5	4	3	②	①	⓪	Address	When 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 through 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.

It is set by a 1-bit memory manipulation instruction and 8-bit memory manipulation instruction.

It is set to FFH by the $\overline{\text{RESET}}$ signal input.

Figure 16-20. Format of Interrupt Mask Flag Register 1L

Symbol	⑦	6	5	4	3	②	①	⓪	Address	When 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 through 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 11 falling edge detection), and selects port 11 falling edge input.

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

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

Figure 16-21. Key Return Mode Register Format

Symbol	7	6	5	4	3	2	①	⓪	Address	When Reset	R/W
KRM	0	0	0	0	KRM3	KRM2	KRMK	KRIF	FFB8H	02H	R/W

KRIF	Key Return Signal
0	Not detected
1	Detected (port 11 falling edge detection)

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

KRM3	KRM2	Selection of Port 11 Falling Edge Input
0	0	P117
0	1	P114-P117
1	0	P112-P117
1	1	P110-P117

Caution When port 11 falling edge detection is used, be sure to clear KRIF to 0 (not cleared to 0 automatically).

16.5.2 Test input signal acknowledge operation

(1) Internal test signal

An internal test input signal (INTWT) is generated when the watch timer overflows and the WTIF flag is set by it. At this time, the standby release signal is generated if it is not masked by the interrupt mask flag (WTMK). The watch function is available by checking the WTIF flag at a shorter cycle than the watch timer overflow cycle.

(2) External test signal

When a falling edge is input to the port 11 (P110 to P117) pins and an external test input signal (INTP11) is generated, KRIF is set. At this time, the standby release signal is generated if it is not masked by the interrupt mask flag (KRMK). If port 11 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 17 STANDBY FUNCTION

17.1 Standby Function and Configuration

17.1.1 Standby function

The standby function is designed to decrease power consumption of the system. The following two modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. The HALT mode is intended to stop the CPU operation clock. System clock oscillator continues oscillation. In this mode, current consumption cannot be decreased as in the STOP mode. The HALT mode is valid to restart immediately upon interrupt request and to carry out intermittent operations such as in watch applications.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the main system clock oscillator stops and the whole system stops. CPU current consumption can be considerably decreased.

Data memory low-voltage hold (down to V_{DD} = 1.8 V) is possible. Thus, the STOP mode is effective to hold data memory contents with ultra-low current consumption. Because this mode can be cleared upon interrupt request, it enables intermittent operations to be carried out.

However, because a wait time is necessary to secure an oscillation stabilization time after the STOP mode is cleared, select the HALT mode if it is necessary to start processing immediately upon interrupt request.

In any mode, all the contents of the register, flag and data memory just before standby mode setting are held. The input/output port output latch and output buffer statuses are also held.

Cautions 1. The STOP mode can be used only when the system operates with the main system clock (subsystem clock oscillation cannot be stopped). The HALT mode can be used with either the main system clock or the subsystem clock.

2. When proceeding to the STOP mode, be sure to stop the peripheral hardware operation and execute the STOP instruction.

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

17.1.2 Standby function control register

A wait time after the STOP mode is cleared upon interrupt request till the oscillation stabilizes is controlled with the oscillation stabilization time select register (OSTS).

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

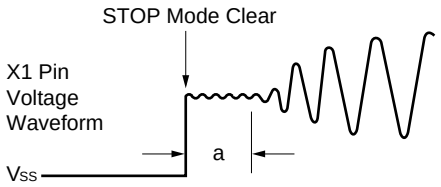
$\overline{\text{RESET}}$ input sets OSTS to 04H. However, it takes $2^{17}/f_X$, not $2^{18}/f_X$, until the STOP mode is cleared by $\overline{\text{RESET}}$ input.

Figure 17-1. Oscillation Stabilization Time Select Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFAH	04H	R/W

OSTS2	OSTS1	OSTS0	Selection of Oscillation Stabilization Time when STOP Mode is Released		
				MCS = 1	MCS = 0
0	0	0	$2^{12}/f_{XX}$	$2^{12}/f_X$ (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 when the STOP mode is released does not include the time required for the clock oscillation to start after the STOP mode has been released (see "a" in the figure below), regardless of whether the mode has been released by the $\overline{\text{RESET}}$ signal or an interrupt request.



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

2. f_X : Main system clock oscillation frequency

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

4. Values in parentheses apply to operating at f_X = 5.0 MHz

17.2 Standby Function Operations

17.2.1 HALT mode

(1) HALT mode set and operating status

The HALT mode is set by executing the HALT instruction. It can be set with the main system clock or the subsystem clock.

The operating status in the HALT mode is described below.

Table 17-1. HALT Mode Operating Status

HALT Mode Setting / Item		HALT execution during main system clock operation		HALT execution during subsystem clock operation	
		w/ subsystem clock[Note 1]	w/o. subsystem clock[Note 2]	Main system clock oscillates	Main system clock stops
Clock generator		Both main system and subsystem clocks can be oscillated. Clock supply to the CPU stops.			
CPU		Operation stop.			
Port (output latch)		Status before HALT mode setting is held.			
16-bit timer/event counter		Operable.			Operable when watch timer output with f_{XT} selected as count clock (f_{XT} is selected as count clock for watch timer).
8-bit timer/event counter 1 and 2		Operable.			Operablewhen TI1 or TI2 is selected as count clock.
Watch timer		Operable if $f_{XX}/2^7$ is selected as count clock.	Operable.		Operable if f_{XT} is selected as count clock.
Watchdog timer		Operable.		Operable.	
A/D converter		Operable.			Operation stops.
Serial Interface		Operable			Operable at external SCK.
LCD controller/driver		Operable if $f_{XX}/2^7$ is selected as count clock.	Operable.		Operable if f_{XT} is selected as count clock.
External interrupt	INTP0	Operable when a clock ($f_{XX}/2^5$, $f_{XX}/2^6$, $f_{XX}/2^7$) for the peripheral hardware is selected as sampling clock.			Operation stops.
	INTP1-INTP5	Operable.			

Notes 1. Including case when external clock is supplied.

2. Including case when external clock is not supplied.

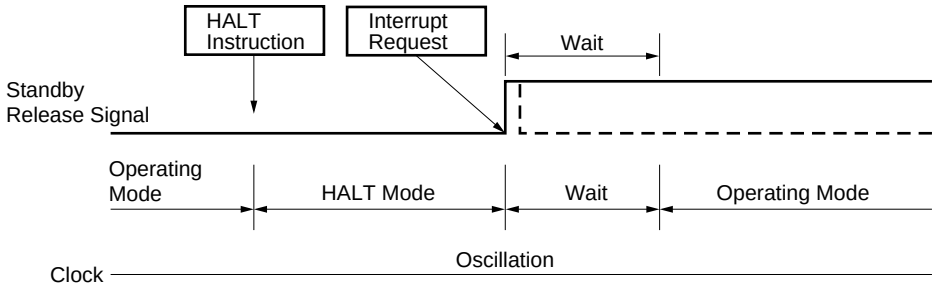
(2) HALT mode clear

The HALT mode can be cleared with the following four types of sources.

(a) Clear upon unmasked interrupt request

If an unmasked interrupt request is generated, the HALT mode is cleared. If interrupt acknowledge is enabled, vectored interrupt service is carried out. If disabled, the next address instruction is executed.

Figure 17-2. HALT Mode Clear upon Interrupt Request Generation



Remarks 1. The broken line indicates the case when the interrupt request which has cleared the standby status is acknowledged.

2. Wait time will be as follows:

- When vectored interrupt service is carried out: 8 to 9 clocks
- When vectored interrupt service is not carried out: 2 to 3 clocks

(b) Clear upon non-maskable interrupt request

If a non-maskable interrupt request is generated, the HALT mode is cleared and vectored interrupt service is carried out whether interrupt acknowledge is enabled or disabled.

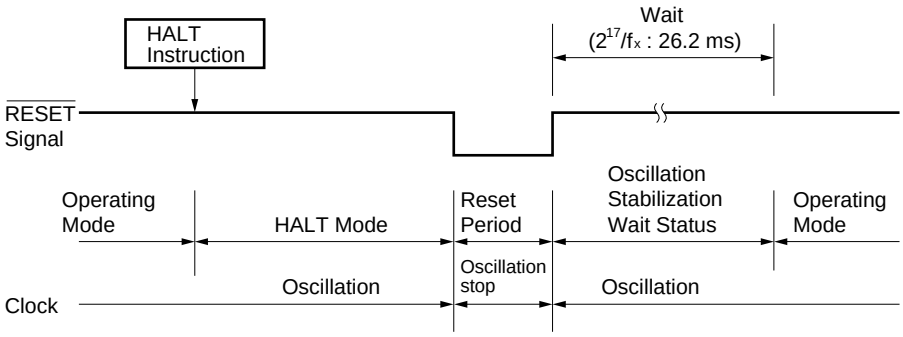
(c) Clear upon unmasked test input

The HALT mode is cleared by unmasked test signal input and the next address instruction of the HALT instruction is executed.

(d) Clear upon $\overline{\text{RESET}}$ input

When a $\overline{\text{RESET}}$ signal is input, the HALT mode is cleared. As is the case with normal reset operation, a program is executed after branch to the reset vector address.

Figure 17-3. HALT Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fx : Main system clock oscillation frequency

2. Time value in parentheses is when fx = 5.0 MHz.

Table 17-2. Operation after HALT Mode Release

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt 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

×: Don't care

17.2.2 STOP mode

(1) STOP mode set and operating status

The STOP mode is set by executing the STOP instruction. It can be set only with the main system clock.

Cautions 1. When the STOP mode is set, the X2 pin is internally connected to V_{DD} via a pull-up resistor to minimize 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 17-3. STOP Mode Operating Status

STOP Mode Setting / Item		With Subsystem Clock	Without Subsystem Clock
Clock generator		Only main system clock stops oscillation.	
CPU		Operation stop.	
Port (output latch)		Status before STOP mode setting is held.	
16-bit timer/event counter		Operable when watch timer output with f_{XT} selected is selected as count clock (f_{XT} is selected as count clock for watch timer).	Operation stops.
8-bit timer/event counter 1 and 2		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		Operation stops.	
Serial Interface	Other than UART	Operable when externally supplied clock is specified as the serial clock.	
	UART	Operation stops.	
LCD controller/driver		Operable when f_{XT} is selected for the count clock.	Operation stops.
External interrupt	INTP0	Operation is impossible.	
	INTP1-INTP5	Operable.	

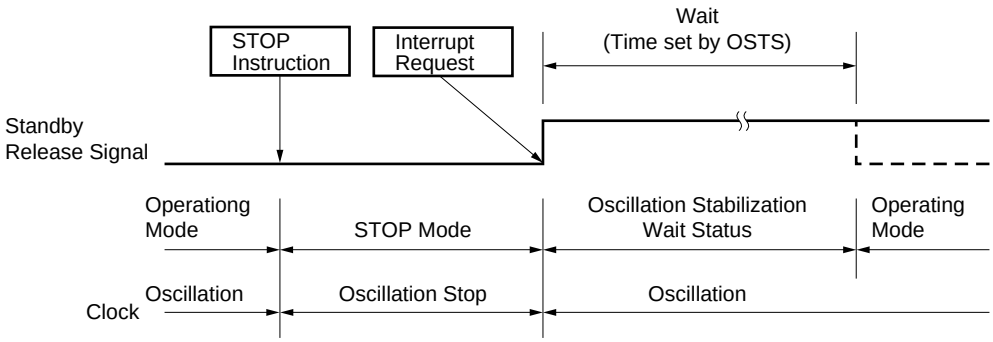
(2) STOP mode release

The STOP mode can be cleared with the following three types of sources.

(a) Release by unmasked interrupt request

If an unmasked interrupt request is generated, the STOP mode is cleared. If interrupt acknowledge is enabled after the lapse of oscillation stabilization time, vectored interrupt service is carried out. If interrupt acknowledge is disabled, the next address instruction is executed.

Figure 17-4. STOP Mode Release by Interrupt Request Generation



Remark The broken line indicates the case when the interrupt request which has cleared the standby status is acknowledged.

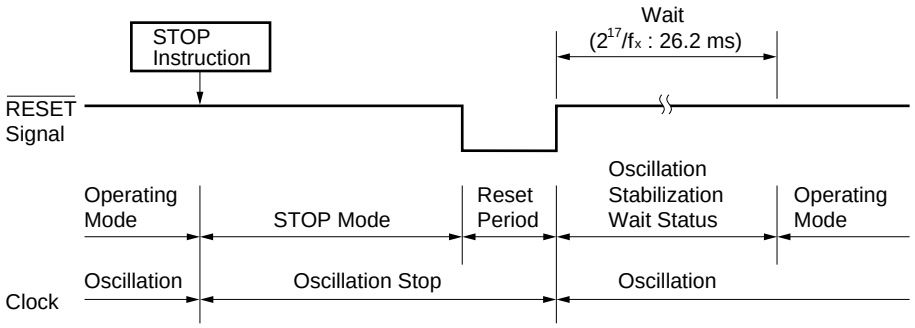
(b) Release by unmasked test input

The STOP mode is cleared by unmasked test signal input. 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

When a $\overline{\text{RESET}}$ signal is input, the STOP mode is cleared and after the lapse of oscillation stabilization time, reset operation is carried out.

Figure 17-5. Release by STOP Mode $\overline{\text{RESET}}$ Input



Remarks 1. fx : Main system clock oscillation frequency

2. Time value in parentheses is when fx = 5.0 MHz.

Table 17-4. Operation after STOP Mode Release

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt 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

×: Don't care

CHAPTER 18 RESET FUNCTION

18.1 Reset Function

The following two operations are available to generate the reset signal.

(1) External reset input with $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer inadvertent program loop 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 18-1. 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 ($2^{17}/f_X$) (refer to **Figure 18-2** to **18-4**).

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 18-1. Block Diagram of Reset Function

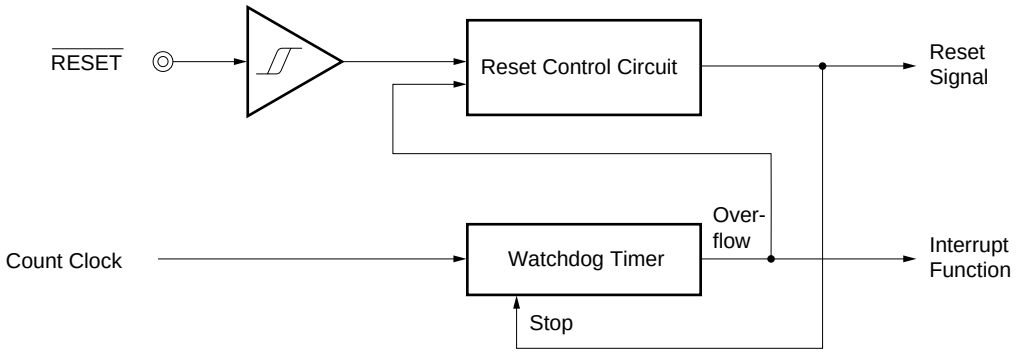


Figure 18-2. Timing of Reset Input by $\overline{\text{RESET}}$ Input

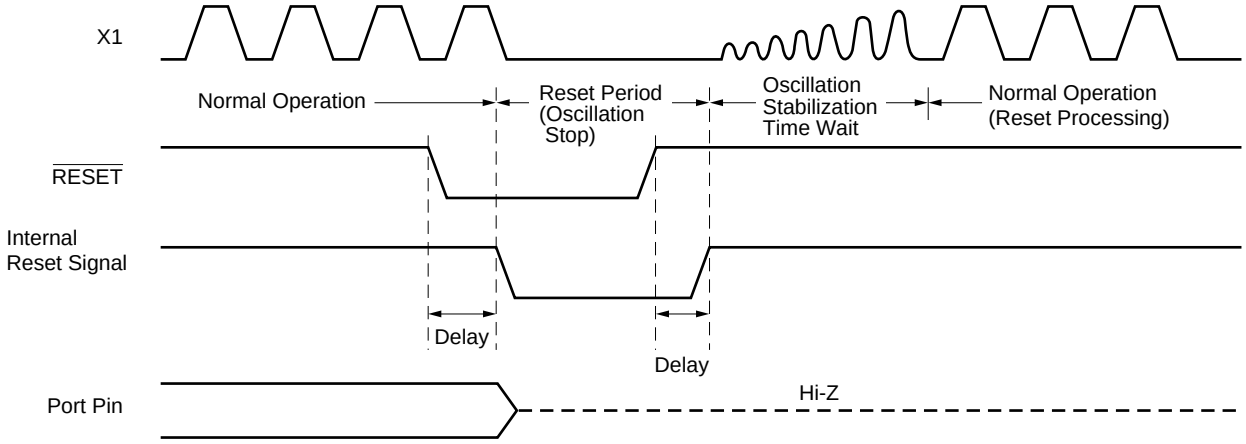


Figure 18-3. Timing of Reset due to Watchdog Timer Overflow

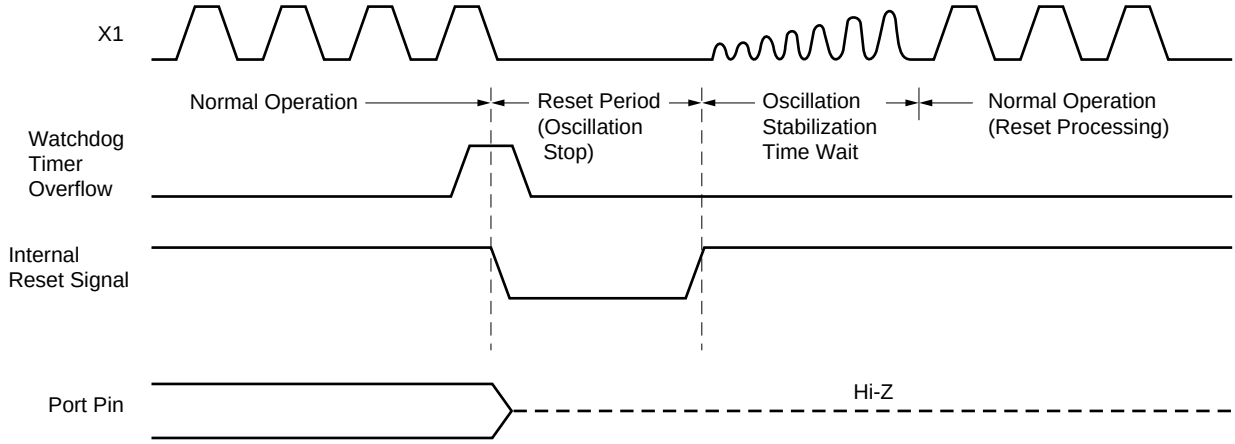


Figure 18-4. Timing of Reset Input in STOP Mode by $\overline{\text{RESET}}$ Input

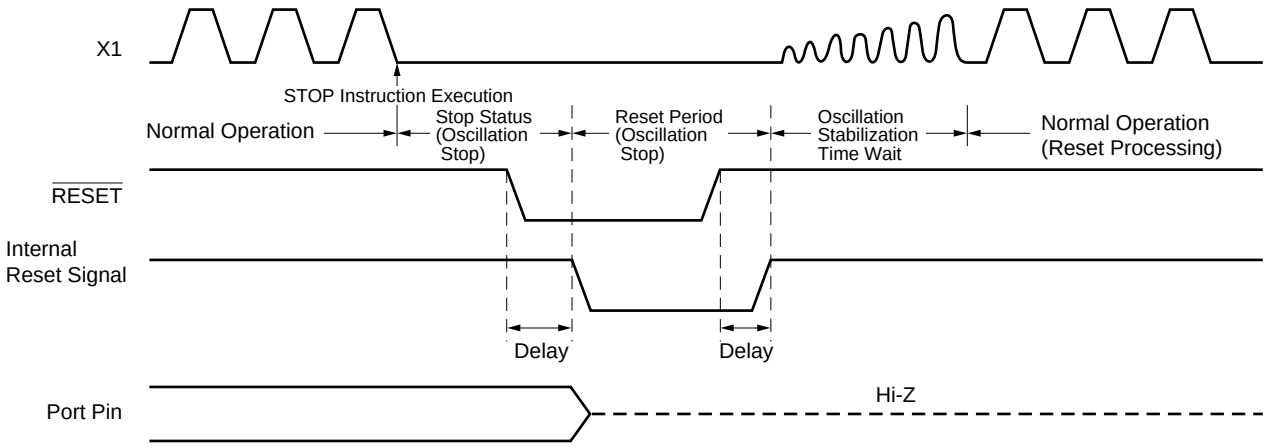


Table 18-1. Hardware Status after Reset (1/2)

Hardware		Status after Reset
Program counter (PC)Note 1		The contents of reset vector tables (0000H and 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	UndefinedNote 2
	General register	UndefinedNote 2
Port (Output latch)	Ports 0-3, Port 7-11 (P0-P3, P7-P11)	00H
Port mode register (PM0-PM3, PM5-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)		C8H
Oscillation stabilization time select register (OSTS)		04H
16-bit timer/event counter	Timer register (TM0)	0000H
	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, 2	Timer register (TM1, TM2)	00H
	Compare registers (CR10, CR20)	Undefined
	Clock select register (TCL1)	00H
	Mode control registers (TMC1)	00H
	Output control register (TOC1)	00H

Notes 1. During reset input or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remains unchanged after reset.

2. When this is reset in the standby mode, the status before reset is held even after reset.

Table 18-1. Hardware Status after Reset (2/2)

Hardware		Status after Reset
Watch timer	Mode control register (TMC2)	00H
	Clock select register (TCL2)	00H
Watchdog timer	Mode register (WDTM)	00H
Serial interface	Clock select register (TCL3)	88H
	Shift registers (SIO0)	Undefined
	Mode registers (CSIM0, CSIM2)	00H
	Serial bus interface control register (SBIC)	00H
	Slave address register (SVA)	Undefined
	Asynchronous serial interface mode register (ASIM)	00H
	Asynchronous serial interface status register (ASIS)	00H
	Baud rate generator control register (BRGC)	00H
	Transmit shift register (TXS)	FFH
	Receive buffer register (RXB)	
	Interrupt timing specify register (SINT)	00H
A/D converter	Mode register (ADM)	01H
	Conversion result register (ADCR)	Undefined
	Input select register (ADIS)	00H
LCD controller/driver	Display mode register (LCDM)	00H
	Display control register (LCDC)	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

CHAPTER 19 μPD78P064B

The μPD78P064B replace the internal mask ROM of the μPD78064B with one-time PROM. Table 19-1 lists the differences among the μPD78P064B and the mask ROM versions.

Table 19-1. Differences among μPD78P064B and Mask ROM Versions

Item	μPD78P064B	Mask ROM versions
Internal ROM structure	One-time PROM	Mask ROM
IC pin	None	Available
V_{PP} pin	Available	None
On-chip mask option dividing resistors for LCD driving power supply	None	Available
Electrical characteristics	Refer to individual data sheet.	

★ **Caution The noise immunity and radiation differ between the PROM model and mask ROM model. To replace a PROM model with a mask ROM model in the course from experimental production to mass production, evaluate your system with the CS model (not ES model) of the mask ROM model.**

19.1 Memory Size Switching Register

The μPD78P064B allows users to define its internal ROM and high-speed RAM sizes 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 and high-speed RAM is possible. IMS is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IMS to C8H.

Figure 19-1. 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	C8H	R/W

ROM3	ROM2	ROM1	ROM0	Internal ROM Capacity selection
1	0	0	0	32 Kbytes
Other than above				Setting prohibited

RAM2	RAM1	RAM0	Internal High-Speed RAM Capacity Selection
1	1	0	1024 bytes
Other than above			Setting prohibited

The IMS settings to give the same memory map as mask ROM versions are shown in Table 19-2.

Table 19-2. Examples of Memory Size Switching Register Settings

Relevant Mask ROM Version	IMS Setting
μPD78064B	C8H

19.2 PROM Programming

The μPD78P064B each incorporate a 32-Kbyte PROM as program memory. To write a program into the μPD78P064B PROM, make the device enter the PROM programming mode by setting the levels of the V_{PP} and $\overline{RESET}$ pins as specified. For the connection of unused pins, refer to paragraph **(2) PROM Programming Mode** in **1.5**.

Caution Write the program in the range of addresses 0000H to 7FFFH (specify the last address as 7FFFH.)
The program cannot be correctly written by a PROM programmer which does not have a write address specification function.

19.2.1 Operating modes

When +5 V or +12.5 V is applied to the V_{PP} pin and a low-level signal is applied to the $\overline{RESET}$ pin, the μPD78064B is set to the PROM programming mode. This is one of the operating modes shown in Table 19-3 below according to the setting of the $\overline{CE}$, $\overline{OE}$, and $\overline{PGM}$ pins.

The PROM contents can be read by setting the read mode.

Table 19-3. PROM Programming Operating Modes

Operating Mode \ Pin	$\overline{RESET}$	V_{PP}	V_{DD}	$\overline{CE}$	$\overline{OE}$	$\overline{PGM}$	D0-D7
Page data latch	L	+12.5 V	+6.5 V	H	L	H	Data input
Page write				H	H	L	High impedance
Byte write				L	H	L	Data input
Program verify				L	L	H	Data output
Program inhibit				×	H	H	High impedance
				×	L	L	
Read		+5 V	+5V	L	L	H	Data output
Output disabled				L	H	×	High impedance
Standby				H	×	×	High impedance

×: L or H

(1) Read mode

Read mode is set by setting $\overline{CE}$ to L and $\overline{OE}$ to L.

(2) Output disable mode

If $\overline{OE}$ is set to H, data output becomes high impedance and the output disable mode is set.

Therefore, if multiple μPD78P064Bs are connected to the data bus, data can be read from any one device by controlling the $\overline{OE}$ pin.

(3) Standby mode

Setting $\overline{CE}$ to H sets the standby mode.

In this mode, data output becomes high impedance irrespective of the status of $\overline{OE}$.

(4) Page data latch mode

Setting $\overline{CE}$ to H, $\overline{PGM}$ to H, and $\overline{OE}$ to L at the start of the page write mode sets the page data latch mode.

In this mode, 1-page 4-byte data is latched in the internal address/data latch circuit.

(5) Page write mode

After a 1-page 4-byte address and data are latched by the page data latch mode, a page write is executed by applying a 0.1-ms program pulse (active-low) to the $\overline{PGM}$ pin while $\overline{CE}$=H and $\overline{OE}$=H. After this, program verification can be performed by setting $\overline{CE}$ to L and $\overline{OE}$ to L.

If programming is not performed by one program pulse, repeated write and verify operations are executed X times (X ≤ 10).

(6) Byte write mode

A byte write is executed by applying a 0.1-ms program pulse (active-low) to the $\overline{PGM}$ pin while $\overline{CE}$=L and $\overline{OE}$=H. After this, program verification can be performed by setting $\overline{OE}$ to L.

If programming is not performed by one program pulse, repeated write and verify operations are executed X times (X ≤ 10).

(7) Program verify mode

Setting $\overline{CE}$ to L, $\overline{PGM}$ to H, and $\overline{OE}$ to L sets the program verify mode.

After writing is performed, this mode should be used to check whether the data was written correctly.

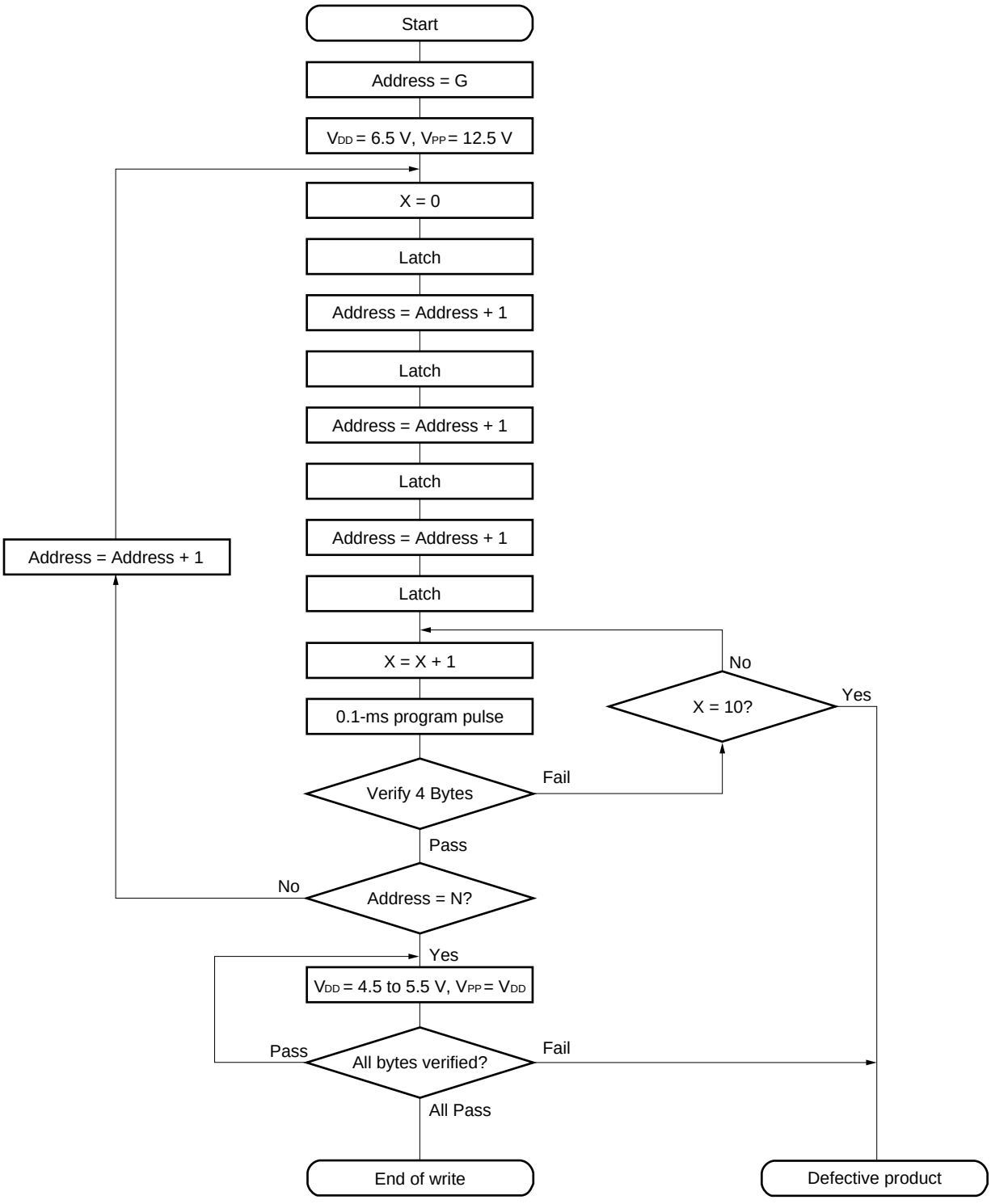
(8) Program inhibit mode

The program inhibit mode is used when the $\overline{OE}$ pins, V_{PP} pins and pins D0 to D7 of multiple μPD78P064Bs are connected in parallel and any one of these devices must be written to.

The page write mode or byte write mode described above is used to perform a write. At this time, the write is not performed on the device which has the $\overline{PGM}$ pin driven high.

19.2.2 PROM write procedure

Figure 19-2. Page Program Mode Flowchart



Remark G = Start address
N = Last address of program

Figure 19-3. Page Program Mode Timing

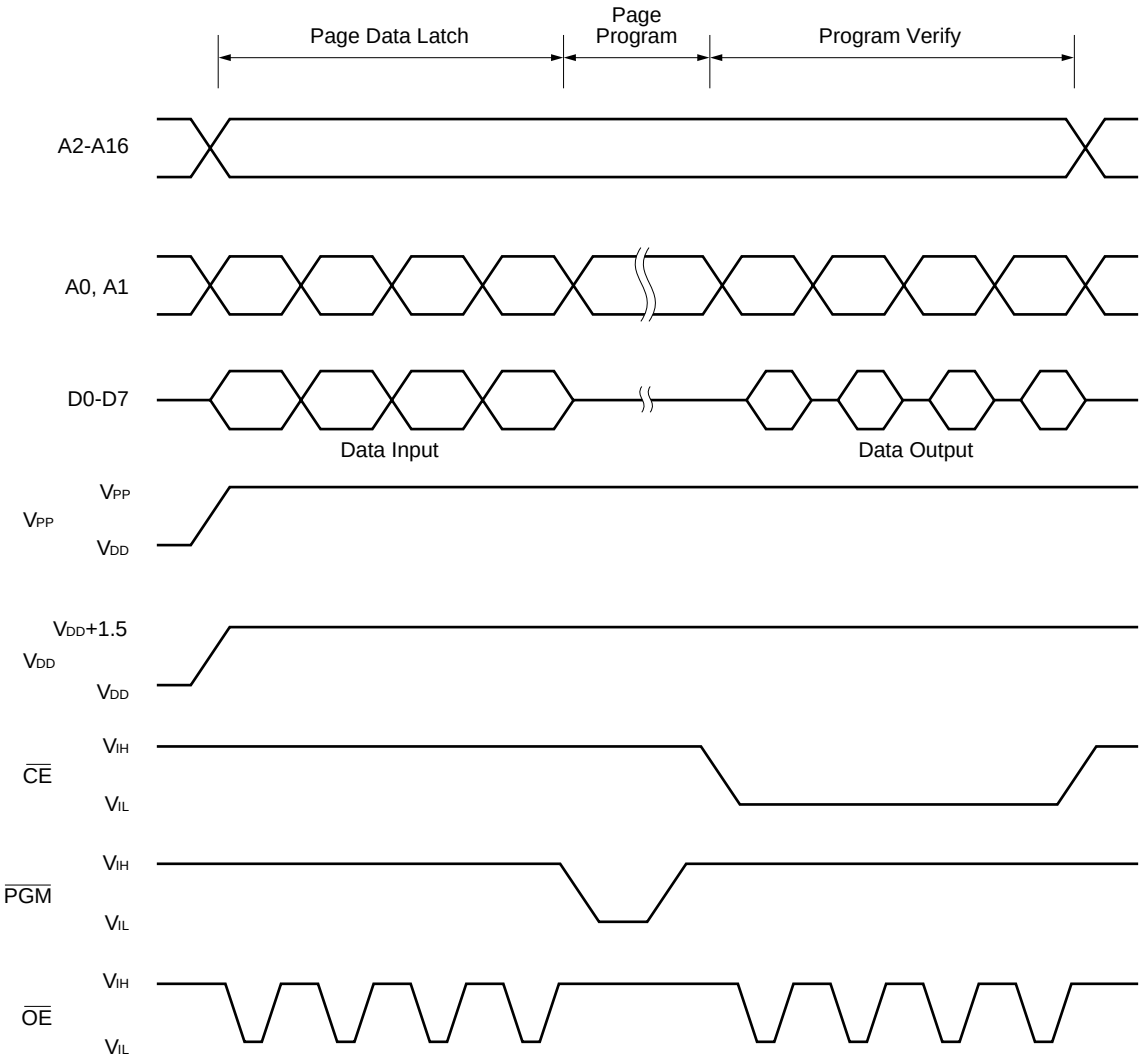
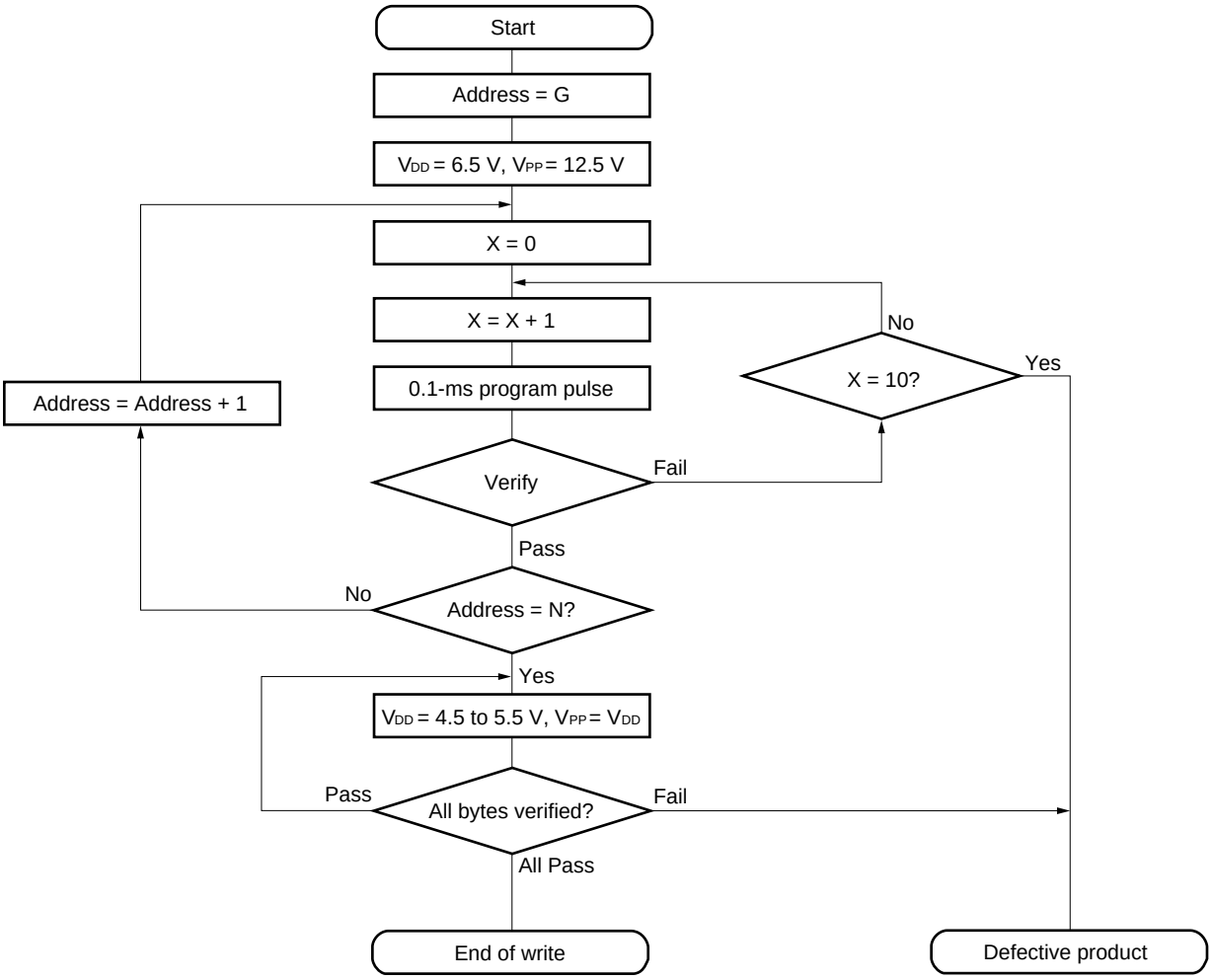
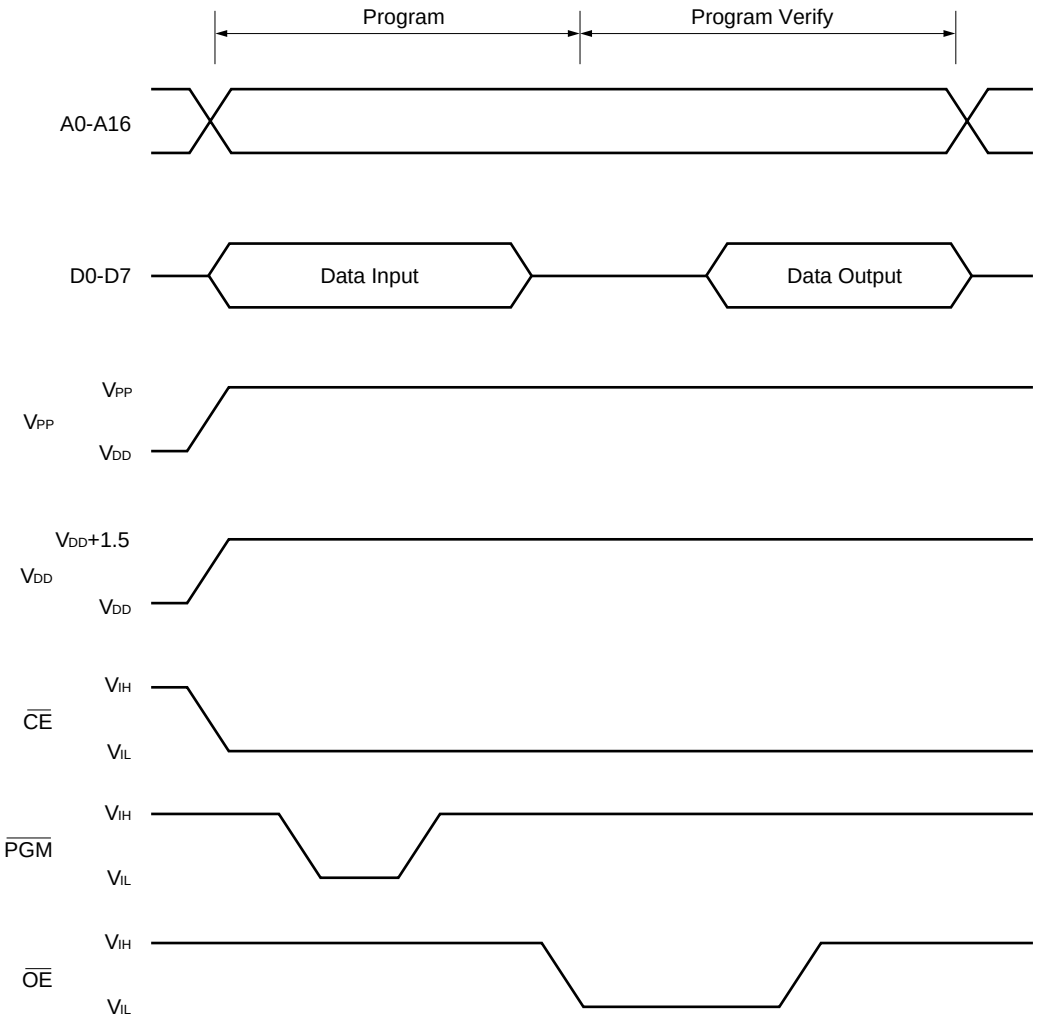


Figure 19-4. Byte Program Mode Flowchart



Remark G = Start address
N = Last address of program

Figure 19-5. Byte Program Mode Timing



Cautions 1. Be sure to apply V_{DD} before applying V_{PP}, and remove it after removing V_{PP}.

2. V_{PP} must not exceed +13.5 V including overshoot voltage.

3. Disconnecting/inserting the device from/to the on-board socket while +12.5 V is being applied to the V_{PP} pin may have an adverse affect on device reliability.

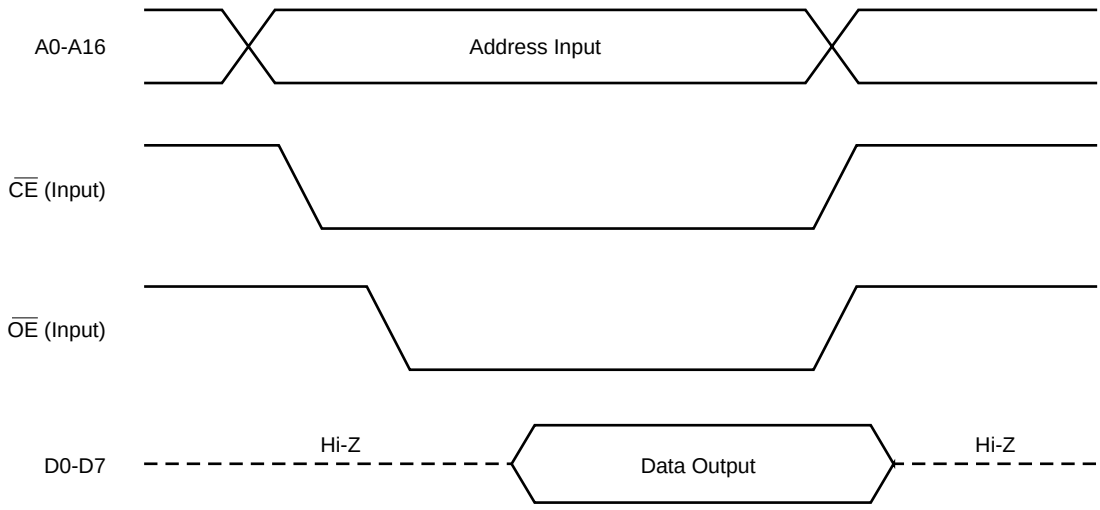
19.2.3 PROM reading procedure

PROM contents can be read onto the external data bus (D0 to D7) using the following procedure.

(1) Fix the $\overline{\text{RESET}}$ pin low, and supply +5 V to the V_{PP} pin. Unused pins are handled as shown in paragraph, **1.5 (2) PROM Programming mode**.
(2) Supply +5 V to the V_{DD} and V_{PP} pins.
(3) Input the address of data to be read to pins A0 through A16.
(4) Read mode is entered.
(5) Data is output to pins D0 through D7.

The timing for steps (2) through (5) above is shown in Figure 19-6.

Figure 19-6. PROM Read Timing



19.3 Screening of One-Time PROM Versions

One-time PROM versions cannot be fully tested by NEC before shipment due to the structure of one-time PROM. Therefore, after users have written data into the PROM, screening should be implemented by user: that is, store devices at high temperature for one day as specified below, and verify their contents after the devices have returned to room temperature.

Storage Temperature	Storage Time
125°C	24 hours

For users who do not wish to implement screening by themselves, NEC provides such users with a charged service in which NEC performs a series of processes from writing one-time PROMs and screening them to verifying their contents for users by request. The PROM version devices which provide this service are called QTOP™ microcontrollers. For details, please consult an NEC sales representative.

CHAPTER 20 INSTRUCTION SET

This chapter describes each instruction set of the μPD78064B subseries as list table. For details of its operation and operation code, refer to the separate document **78K/0 Series User's Manual — Instruction (U12326E)**.

20.1 Legends Used in Operation List

20.1.1 Operand identifiers and methods

Operands are written in “Operand” column of each instruction in accordance with the method of the instruction operand identifier (refer to the assembler specifications for detail). When there are two or more 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, write an appropriate numeric value or a label. When using a label, be sure to write the #, !, $, and [ ] symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for description.

Table 20-1. Operand Identifiers and Methods

Identifier	Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7),
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special-function register symbol**Note**
sfrp	Special-function register symbol (16-bit manipulatable register even addresses only)**Note**
saddr	FE20H-FF1FH Immediate data or labels
saddrp	FE20H-FF1FH Immediate data or labels (even address only)
addr16	0000H-FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions)
addr11	0800H-0FFFH Immediate data or labels
addr5	0040H-007FH Immediate data or labels (even address only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special-function register symbols, refer to **Table 3-4 Special-Function Register List**.

20.1.2 Description of “operation” column

A	:	A register; 8-bit accumulator
X	:	X register
B	:	B register
C	:	C register
D	:	D register
E	:	E register
H	:	H register
L	:	L register
AX	:	AX register pair; 16-bit accumulator
BC	:	BC register pair
DE	:	DE register pair
HL	:	HL register pair
PC	:	Program counter
SP	:	Stack pointer
PSW	:	Program status word
CY	:	Carry flag
AC	:	Auxiliary carry flag
Z	:	Zero flag
RBS	:	Register bank select flag
IE	:	Interrupt request enable flag
NMIS	:	Non-maskable interrupt servicing flag
( )	:	Memory contents indicated by address or register contents in parentheses
$\times_H$, $\times_L$	:	High-order 8 bits and low-order 8 bits of 16-bit register
$\wedge$	:	Logical product (AND)
$\vee$	:	Logical sum (OR)
$\veebar$	:	Exclusive logical sum (exclusive OR)
$\overline{\quad}$	:	Inverted data
addr16	:	16-bit immediate data or label
jdisp8	:	Signed 8-bit data (displacement value)

20.1.3 Description of “flag operation” column

(Blank)	:	Nt affected
0	:	Cleared to 0
1	:	Set to 1
×	:	Set/cleared according to the result
R	:	Previously saved value is restored

20.2 Operation List

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	MOV	r, #byte	2	4	–	r ← byte	
		saddr, #byte	3	6	7	(saddr) ← byte	
		sfr, #byte	3	–	7	sfr ← byte	
		A, r Note 3	1	2	–	A ← r	
		r, A Note 3	1	2	–	r ← A	
		A, saddr	2	4	5	A ← (saddr)	
		saddr, A	2	4	5	(saddr) ← A	
		A, sfr	2	–	5	A ← sfr	
		sfr, A	2	–	5	sfr ← A	
		A, !addr16	3	8	9	A ← (addr16)	
		!addr16, A	3	8	9	(addr16) ← A	
		PSW, #byte	3	–	7	PSW ← byte	× × ×
		A, PSW	2	–	5	A ← PSW	
		PSW, A	2	–	5	PSW ← A	× × ×
		A, [DE]	1	4	5	A ← (DE)	
		[DE], A	1	4	5	(DE) ← A	
		A, [HL]	1	4	5	A ← (HL)	
		[HL], A	1	4	5	(HL) ← A	
		A, [HL + byte]	2	8	9	A ← (HL + byte)	
		[HL + byte], A	2	8	9	(HL + byte) ← A	
		A, [HL + B]	1	6	7	A ← (HL + B)	
		[HL + B], A	1	6	7	(HL + B) ← A	
		A, [HL + C]	1	6	7	A ← (HL + C)	
		[HL + C], A	1	6	7	(HL + C) ← A	
	XCH	A, r Note 3	1	2	–	A ↔ r	
		A, saddr	2	4	6	A ↔ (saddr)	
		A, sfr	2	–	6	A ↔ sfr	
		A, !addr16	3	8	10	A ↔ (addr16)	
		A, [DE]	1	4	6	A ↔ (DE)	
		A, [HL]	1	4	6	A ↔ (HL)	
		A, [HL + byte]	2	8	10	A ↔ (HL + byte)	
		A, [HL + B]	2	8	10	A ↔ (HL + B)	
		A, [HL + C]	2	8	10	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"

Remark One instruction clock cycle is one cycle of the CPU clock (fCPU) selected by the processor clock control register (PCC) register.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
16-bit data transfer	**MOVW**	rp, #word	3	6	–	rp ← word	
		saddrp, #word	4	8	10	(saddrp) ← word	
		sfrp, #word	4	–	10	sfrp ← word	
		AX, saddrp	2	6	8	AX ← (saddrp)	
		saddrp, AX	2	6	8	(saddrp) ← AX	
		AX, sfrp	2	–	8	AX ← sfrp	
		sfrp, AX	2	–	8	sfrp ← AX	
		AX, rp **Note 3**	1	4	–	AX ← rp	
		rp, AX **Note 3**	1	4	–	rp ← AX	
		AX, !addr16	3	10	12	AX ← (addr16)	
		!addr16, AX	3	10	12	(addr16) ← AX	
	XCHW	AX, rp **Note 3**	1	4	–	AX ↔ rp	
8-bit operation	**ADD**	A, #byte	2	4	–	A, CY ← A + byte	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte	× × ×
		A, r **Note 4**	2	4	–	A, CY ← A + r	× × ×
		r, A	2	4	–	r, CY ← r + A	× × ×
		A, saddr	2	4	5	A, CY ← A + (saddr)	× × ×
		A, !addr16	3	8	9	A, CY ← A + (addr16)	× × ×
		A, [HL]	1	4	5	A, CY ← A + (HL)	× × ×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte)	× × ×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B)	× × ×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C)	× × ×
	ADDC	A, #byte	2	4	–	A, CY ← A + byte + CY	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte + CY	× × ×
		A, r **Note 4**	2	4	–	A, CY ← A + r + CY	× × ×
		r, A	2	4	–	r, CY ← r + A + CY	× × ×
		A, saddr	2	4	5	A, CY ← A + (saddr) + CY	× × ×
		A, !addr16	3	8	9	A, CY ← A + (addr16) + CY	× × ×
		A, [HL]	1	4	5	A, CY ← A + (HL) + CY	× × ×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte) + CY	× × ×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B) + CY	× × ×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C) + CY	× × ×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Only when rp = BC, DE or HL

4. Except "r = A"

Remark One instruction clock cycle is one cycle of the CPU clock (fCPU) selected by the processor clock control register (PCC) register.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	SUB	A, #byte	2	4	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r	×	×	×
		r, A	2	4	–	r, CY ← r – A	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr)	×	×	×
		A, !addr16	3	8	9	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C)	×	×	×
	SUBC	A, #byte	2	4	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r – CY	×	×	×
		r, A	2	4	–	r, CY ← r – A – CY	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr) – CY	×	×	×
		A, !addr16	3	8	9	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C) – CY	×	×	×
	AND	A, #byte	2	4	–	A ← A ∧ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	4	–	A ← A ∧ r	×		
		r, A	2	4	–	r ← r ∧ A	×		
		A, saddr	2	4	5	A ← A ∧ (saddr)	×		
		A, !addr16	3	8	9	A ← A ∧ (addr16)	×		
		A, [HL]	1	4	5	A ← A ∧ (HL)	×		
		A, [HL + byte]	2	8	9	A ← A ∧ (HL + byte)	×		
		A, [HL + B]	2	8	9	A ← A ∧ (HL + B)	×		
		A, [HL + C]	2	8	9	A ← A ∧ (HL + C)	×		

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except "r = A"

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC) register.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit operation	OR	A, #byte	2	4	–	A ← A∨byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr)∨byte	×
		A, r Note 3	2	4	–	A ← A∨r	×
		r, A	2	4	–	r ← r∨A	×
		A, saddr	2	4	5	A ← A∨(saddr)	×
		A, !addr16	3	8	9	A ← A∨(addr16)	×
		A, [HL]	1	4	5	A ← A∨(HL)	×
		A, [HL + byte]	2	8	9	A ← A∨(HL + byte)	×
		A, [HL + B]	2	8	9	A ← A∨(HL + B)	×
		A, [HL + C]	2	8	9	A ← A∨(HL + C)	×
	XOR	A, #byte	2	4	–	A ← A⊻byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr)⊻byte	×
		A, r Note 3	2	4	–	A ← A⊻r	×
		r, A	2	4	–	r ← r⊻A	×
		A, saddr	2	4	5	A ← A⊻(saddr)	×
		A, !addr16	3	8	9	A ← A⊻(addr16)	×
		A, [HL]	1	4	5	A ← A⊻(HL)	×
		A, [HL + byte]	2	8	9	A ← A⊻(HL + byte)	×
		A, [HL + B]	2	8	9	A ← A⊻(HL + B)	×
		A, [HL + C]	2	8	9	A ← A⊻(HL + C)	×
	CMP	A, #byte	2	4	–	A – byte	× × ×
		saddr, #byte	3	6	8	(saddr) – byte	× × ×
		A, r Note 3	2	4	–	A – r	× × ×
		r, A	2	4	–	r – A	× × ×
		A, saddr	2	4	5	A – (saddr)	× × ×
		A, !addr16	3	8	9	A – (addr16)	× × ×
		A, [HL]	1	4	5	A – (HL)	× × ×
		A, [HL + byte]	2	8	9	A – (HL + byte)	× × ×
		A, [HL + B]	2	8	9	A – (HL + B)	× × ×
		A, [HL + C]	2	8	9	A – (HL + C)	× × ×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except "r = A"

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC) register.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	**ADDW**	AX, #word	3	6	–	AX, CY ← AX + word	×	×	×
	SUBW	AX, #word	3	6	–	AX, CY ← AX – word	×	×	×
	CMPW	AX, #word	3	6	–	AX – word	×	×	×
Multiply/ divide	**MULU**	X	2	16	–	AX ← A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C			
Increment/ decrement	**INC**	r	1	2	–	r ← r + 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) + 1	×	×	
	DEC	r	1	2	–	r ← r – 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) – 1	×	×	
	INCW	rp	1	4	–	rp ← rp + 1			
	DECW	rp	1	4	–	rp ← rp – 1			
Rotate	**ROR**	A, 1	1	2	–	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$ time			×
	ROL	A, 1	1	2	–	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$ time			×
	RORC	A, 1	1	2	–	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$ time			×
	ROLC	A, 1	1	2	–	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$ time			×
	ROR4	[HL]	2	10	12	$A_{3-0} \leftarrow (HL)_{3-0}$, $(HL)_{7-4} \leftarrow A_{3-0}$, $(HL)_{3-0} \leftarrow (HL)_{7-4}$			
	ROL4	[HL]	2	10	12	$A_{3-0} \leftarrow (HL)_{7-4}$, $(HL)_{3-0} \leftarrow A_{3-0}$, $(HL)_{7-4} \leftarrow (HL)_{3-0}$			
BCD adjust	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit 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	CY ← (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	8	sfr.bit ← CY			
		A.bit, CY	2	4	–	A.bit ← CY			
		PSW.bit, CY	3	–	8	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	6	8	(HL).bit ← CY			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC) register.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	**AND1**	CY, saddr.bit	3	6	7	CY ← CY ∧ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ∧ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ∧ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ∧ (HL).bit			×
	OR1	CY, saddr.bit	3	6	7	CY ← CY ∨ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ∨ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ∨ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ∨ (HL).bit			×
	XOR1	CY, saddr.bit	3	6	7	CY ← CY ⊻ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ⊻ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ⊻ A.bit			×
		CY, PSW. bit	3	–	7	CY ← CY ⊻ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ⊻ (HL).bit			×
	SET1	saddr.bit	2	4	6	(saddr.bit) ← 1			
		sfr.bit	3	–	8	sfr.bit ← 1			
		A.bit	2	4	–	A.bit ← 1			
		PSW.bit	2	–	6	PSW.bit ← 1	×	×	×
		[HL].bit	2	6	8	(HL).bit ← 1			
	CLR1	saddr.bit	2	4	6	(saddr.bit) ← 0			
		sfr.bit	3	–	8	sfr.bit ← 0			
		A.bit	2	4	–	A.bit ← 0			
		PSW.bit	2	–	6	PSW.bit ← 0	×	×	×
		[HL].bit	2	6	8	(HL).bit ← 0			
	SET1	CY	1	2	–	CY ← 1			1
	CLR1	CY	1	2	–	CY ← 0			0
	NOT1	CY	1	2	–	CY ← $\overline{CY}$			×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC) register.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call/return	**CALL**	!addr16	3	7	–	$(SP-1) \leftarrow (PC+3)_H$, $(SP-2) \leftarrow (PC+3)_L$, $PC \leftarrow$ addr16, $SP \leftarrow SP-2$			
	CALLF	!addr11	2	5	–	$(SP-1) \leftarrow (PC+2)_H$, $(SP-2) \leftarrow (PC+2)_L$, $PC_{15-11} \leftarrow 00001$, $PC_{10-0} \leftarrow$ addr11, $SP \leftarrow SP-2$			
	CALLT	[addr5]	1	6	–	$(SP-1) \leftarrow (PC+1)_H$, $(SP-2) \leftarrow (PC+1)_L$, $PC_H \leftarrow$ (00000000, addr5 + 1), $PC_L \leftarrow$ (00000000, addr5), $SP \leftarrow SP-2$			
	BRK		1	6	–	$(SP-1) \leftarrow PSW$, $(SP-2) \leftarrow (PC+1)_H$, $(SP-3) \leftarrow (PC+1)_L$, $PC_H \leftarrow$ (003FH), $PC_L \leftarrow$ (003EH), $SP \leftarrow SP-3$, $IE \leftarrow 0$			
	RET		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $SP \leftarrow SP+2$			
	RETI		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP+2)$, $SP \leftarrow SP+3$, $NMIS \leftarrow 0$	R	R	R
	RETB		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP+2)$, $SP \leftarrow SP+3$	R	R	R
Stack manipulate	**PUSH**	PSW	1	2	–	$(SP-1) \leftarrow PSW$, $SP \leftarrow SP-1$			
		rp	1	4	–	$(SP-1) \leftarrow rp_H$, $(SP-2) \leftarrow rp_L$, $SP \leftarrow SP-2$			
	POP	PSW	1	2	–	$PSW \leftarrow (SP)$, $SP \leftarrow SP+1$	R	R	R
		rp	1	4	–	$rp_H \leftarrow (SP+1)$, $rp_L \leftarrow (SP)$, $SP \leftarrow SP+2$			
	MOVW	SP, #word	4	–	10	$SP \leftarrow$ word			
		SP, AX	2	–	8	$SP \leftarrow AX$			
		AX, SP	2	–	8	$AX \leftarrow SP$			
Unconditional branch	**BR**	!addr16	3	6	–	$PC \leftarrow$ addr16			
		$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8			
		AX	2	8	–	$PC_H \leftarrow A$, $PC_L \leftarrow X$			
Conditional branch	**BC**	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if CY = 1			
	BNC	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if CY = 0			
	BZ	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if Z = 1			
	BNZ	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if Z = 0			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC) register.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Conditional branch	**BT**	saddr.bit, $addr16	3	8	9	PC ← PC + 3 + jdisp8 if(saddr.bit) = 1	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 1	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1	
		PSW.bit, $addr16	3	–	9	PC ← PC + 3 + jdisp8 if PSW.bit = 1	
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 1	
	BF	saddr.bit, $addr16	4	10	11	PC ← PC + 4 + jdisp8 if(saddr.bit) = 0	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 0	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 0	
		PSW.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if PSW. bit = 0	
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 0	
	BTCLR	saddr.bit, $addr16	4	10	12	PC ← PC + 4 + jdisp8 if(saddr.bit) = 1 then reset(saddr.bit)	
		sfr.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit	
		PSW.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	× × ×
		[HL].bit, $addr16	3	10	12	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit	
	DBNZ	B, $addr16	2	6	–	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0	
		C, $addr16	2	6	–	C ← C –1, then PC ← PC + 2 + jdisp8 if C ≠ 0	
		saddr. $addr16	3	8	10	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if(saddr) ≠ 0	
CPU control	**SEL**	RBn	2	4	–	RBS1, 0 ← n	
	NOP		1	2	–	No Operation	
	EI		2	–	6	IE ← 1(Enable Interrupt)	
	DI		2	–	6	IE ← 0(Disable Interrupt)	
	HALT		2	6	–	Set HALT Mode	
	STOP		2	6	–	Set STOP Mode	

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC) register.

20.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

First Operand \ Second Operand	#byte	A	r[Note]	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

Note Except r = A

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

1st Operand \ Second Operand	#word	AX	rp[Note]	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW[Note]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

First Operand \ Second Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call/instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

First Operand \ Second Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

[MEMO]

APPENDIX A DIFFERENCES BETWEEN μPD78064 AND 78064B SUBSERIES

Table A-1 lists the major differences between the μPD78064 and 78064B subseries.

Table A-1. Major Differences between μPD78064 and 78064B Subseries

Item \ Part Number	μPD78064 Subseries	μPD78064B Subseries
EMI noise measures	None	Provided
Y subseries	Provided	None
PROM model	μPD78P064	μPD78P064B
Internal ROM size	μPD78062: 16K bytes μPD78063: 24K bytes μPD78064: 32K bytes μPD78P064: 32K bytes	μPD78064B: 32K bytes μPD78P064B: 32K bytes
Internal high-speed RAM size	μPD78062: 512 bytes μPD78063: 1024 bytes μPD78064: 1024 bytes μPD78P064: 1024 bytes	μPD78064B: 1024 bytes μPD78P064B: 1024 bytes
V_{DD} pin	Positive power supply (except ports)	Positive power supply (including ports)
V_{SS} pin	Ground potential (except ports)	Ground potential (including ports)
AV_{DD} pin	Analog power supply to A/D converter and power supply to ports	Analog power supply to A/D converter
AV_{SS} pin	Ground of A/D converter and ground of ports	Ground of A/D converter
Package	• 100-pin plastic QFP (fine pitch) (14 × 14 mm) • 100-pin plastic LQFP (fine pitch) (14 × 14 mm) • 100-pin plastic QFP (14 × 20 mm) • 100-pin ceramic WQFN**Note**	• 100-pin plastic QFP (fine pitch) (14 × 14 mm) • 100-pin plastic LQFP (fine pitch) (14 × 14 mm) • 100-pin plastic QFP (14 × 20 mm)
Electrical specifications and recommended soldering conditions	Refer to individual Data Sheet.	

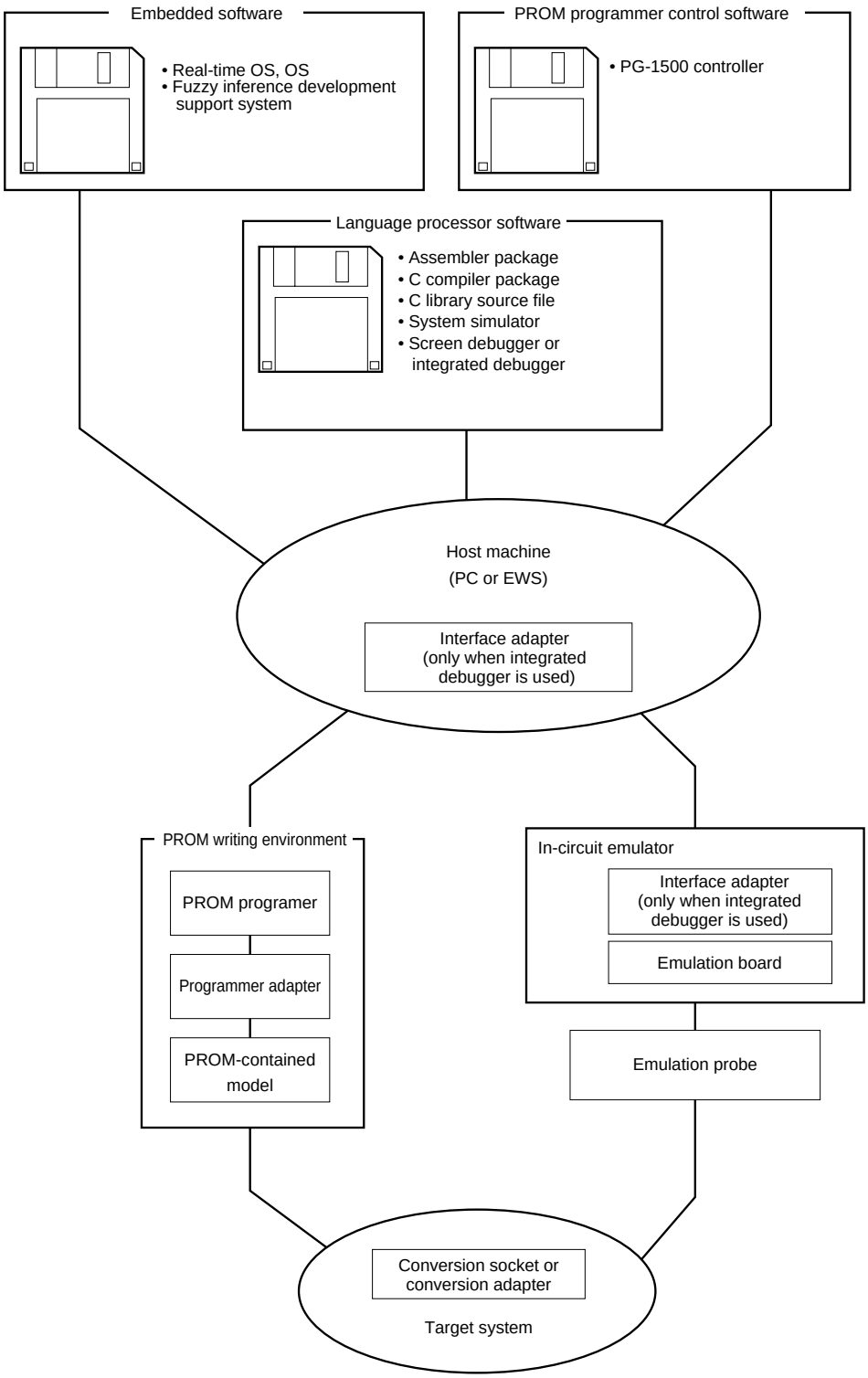
Note PROM model only

[MEMO]

APPENDIX B DEVELOPMENT TOOLS

The following development tools are available for the development of systems which employ the μPD78064B subseries. Figure B-1 shows the configuration example of the tools.

Figure B-1. Development Tool Configuration



B.1 Language Processing Software

RA78K/0 Assembler Package	This assembler converts a program written in mnemonics into an object code executable with a microcomputer. Further, this assembler is provided with functions capable of automatically creating symbol tables and branch instruction optimization. This data file is used together with DF78064 device file (option).
	Part number: μS××××RA78K0
CC78K/0 C Compiler Package	This compiler converts a program written in C language into an object code executable with a microcomputer. This data file is used together with RA78K/0 assembler package and DF78064 device file (option).
	Part number: μS××××CC78K0
DF78064 Device File[Note]	File containing information unique to the devices. This data file is used together with RA78K/0, CC78K/0, SM78K0, ID78K0 and SD78K/0.
	Part number: μS××××DF78064
CC78K/0-L C Compiler Library Source File	Source program of a function configurating object library included in CC78K/0 C compiler. This file is necessary when customers change the object library in CC78K/0 following their specifications.
	Part number: μS××××CC78K0-L

Note This device file can be used for any of RA78K/0, CC78K/0, SM78K0, ID78K0 and SD78K/0.

Remark ×××× of the part number differs depending on the host machine and OS used. Refer to the table below.

μS×××× RA78K0
μS×××× CC78K0
μS×××× DF78064
μS×××× CC78K0-L

××××	Host Machine	OS	Supply Medium
5A13	PC-9800 series	MS-DOS (ver. 3.30 to 6.2)[Note]	3.5-inch 2HD
5A10			5-inch 2HD
7B13	IBM PC/AT or compatible machine	Refer to **B.4**.	3.5-inch 2HC
7B10			5-inch 2HC
3H15	HP9000 series 300™	HP-UX™ (rel.7.05B)	Cartidge tape (QIC-24)
3P16	HP9000 series 700™	HP-UX (rel.9.01)	Digital audio tape (DAT)
3K15	SPARCstation™	SunOS™ (rel.4.1.1)	Cartidge tape (QIC-24)
3M15	EWS4800 series (RISC)	EWS-UX/V (rel.4.0)	

Note The task swap function is not available with this software though the function is provided in MS-DOS version 5.0 or later.

B.2 PROM Programming Tools

B.2.1 Hardware

PG-1500 PROM Programmer	This is a PROM programmer capable of programming the single-chip microcontroller incorporating PROM by manipulating from the stand-alone or host machine through connection of an optional PROM programmer adapter and attached board. It can also program representative PROMs ranging from 256K bits to 4M bits.
PA-78P064GC**Note** PA-78P064GF**Note** ★ PA-78P0308GC ★ PA-78P0308GF PROM Programmer Adapter	PROM programmer adapter common to the μPD78P064. Used connected to the PG-1500. PA-78P064GC : 100-pin plastic QFP (GC-7EA, GC-8EU type) PA-78P064GF : 100-pin plastic QFP (GF-3BA type) PA-78P0308GC : 100-pn plastic QFP (GC-7EA, GC-8EU type) PA-78P0308GF : 100-pin plastic QFP (GF-3BA type)

Note For maintenance use

B.2.2 Software

PG-1500 Controller	The PG-1500 is controlled in the host machine through connection with the host machine and PG-1500 via serial and parallel interfaces.
	Part number : μS××××PG1500

Remark ×××× of the part number differs depending on the host machine and OS used. Refer to the table below.

μS×××× PG1500

××××	Host Machine	OS	Supply Medium
5A13	PC-9800 series	MS-DOS (ver. 3.30 to 6.2)**Note**	3.5-inch 2HD
5A10			5-inch 2HD
7B13	IBM PC/AT or compatible machine	Refer to **B.4**.	3.5-inch 2HD
7B10			5-inch 2HC

Note The task swap function is not available with this software though the function is provided in MS-DOS version 5.0 or later.

B.3 Debugging Tools

B.3.1 Hardware (1/2)

IE-78000-R-A In-Circuit Emulator (supporting integrated debugger)	This in-circuit emulator debugs hardware and software when an application system using the 78K/0 series is developed. It supports the integrated debugger (ID78K0). This emulator is used in combination with an emulation probe and an interface adapter that connects the emulator with the host machine.
IE-70000-98-IF-B Interface Adapter	Adapter necessary when using the PC-9800 series (except the notebook type) as the host machine of the IE-78000-R-A.
IE-70000-98N-IF Interface Adapter	Adapter and cable necessary when using the notebook type PC-9800 series as the host machine of the IE-78000-R-A.
IE-70000-PC-IF-B Interface Adapter	Adapter necessary when using IBM PC/AT as the host machine of the IE-78000-R-A.
IE-78000-R-SV3 Interface Adapter	Adapter and cable necessary when using an EWS as the host machine of the IE-78000-R-A. This cable is connected to the board in the IE-78000-R-A. As Ethernet™, 10Base-5 is supported. If the other methods are used, a commercially available conversion adapter is necessary.
IE-78000-R In-Circuit Emulator (supporting screen debugger)	This in-circuit emulator debugs hardware and software when an application system using the 78K/0 series is developed. It supports the screen debugger (SD78K/0). This emulator is used in combination with an emulation probe. When this emulator is connected with a host machine and PROM programmer, efficient debugging can be performed.
★ IE-78064-R-EM**Note** IE-780308-R-EM Emulation Board	This board emulates the peripheral hardware peculiar to a device (IE-78064-R-EM: 3 to 5.5 V, IE-780308-R-EM: 2.0 to 5.0 V). It is used in combination with an in-circuit emulator.

★ **Note** For maintenance use

B.3.1 Hardware (2/2)

EP-78064GC-R Emulation Probe		This is a probe to connect an in-circuit emulator and target system. It is for a 100-pin plastic QFP (GC-7EA, GC-8EU type). One 100-pin conversion adapter, TGC-100SDW, which facilitates development of the target system is supplied as an accessory.
	TGC-100SDW conversion adapter	This conversion adapter connects the board of the target system created for mounting a 100-pin plastic QFP (GC-7EA, GC-8EU type) and the EP-78064GC-R. This is a product of Tokyo Eletech Co. (Tokyo 03-5295-1661). When purchasing, consult your NEC distributor.
EP-78064GF-R Emulation Probe		This probe connects an in-circuit emulator and target system. It is for a 100-pin plastic QFP (GF-3BA type). One 100-pin conversion adapter, EV-9200GF-100, which facilitates development of the target system is supplied as an accessory.
	EV-9200GF-100 conversion socket	This conversion socket connects the board of the target system created for mounting a 100-pin plastic QFP (GF-3BA type) and the EP-78064GF-R.

★

Remark The TGC-100SDW is available singly.
Five EV-9200GF-100s are available as a set.

B.3.2 Software (1/3)

SM78K0 (System Simulator)	This simulator simulates operations of the target system from a Windows-installed host computer, enabling debuggung in C source level or assembler level. By using SM78K0, logical and performance verification processes can be performed independently of hardware development work without using in-circuit emulator in-circuit emulator, which leads to reduction in development workload and improvement in software quality. This system simulator is used together with the DF78064 device file (DF78064, option).
	Part number : μS××××SM78K0

Remark ×××× of the part number differs depending on the host machine and OS used. Refer to the table below.

μS×××× SM78K0

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	MS-DOS (ver. 3.30 to 6.2)**Note** + Windows (ver. 3.0 and 3.1)	3.5-inch 2HD
AB13	IBM PC/AT or compatible machine (on Japanese Windows)	Refer to **B.4**.	3.5-inch 2HC
BB13	IBM PC/AT or compatible machine (on English Wondows)		

Note The task swap function is not available with this software though the function is provided in MS-DOS version 5.0 or later.

B.3.2 Software (2/3)

ID78K0 Integrated Debugger	This is a control program that debugs the 78K/0 series. It employs Windows on a personal computer and OSF/MOtif™ on EWS as a graphical user interface, and provides the environment and operability conforming to these interfaces. Moreover, C language debugging functions are improved, and the trace result can be displayed at C language level by using a window integrating function that associates the source program, disassemble display, and memory display with the trace result. In addition, the debugging of a program can be made more efficient by using a real-time OS by incorporating function expansion modules such as a task debugger and system performance analyzer. This debugger is used in combination with an optional device file (DF78064).
	Part number : μS××××ID78K0

Remark ×××× in the parts number differs depending on the host machine and OS used.

μS×××× SM78K0

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	MS-DOS (Ver. 3.30 to Ver. 6.2[Note]) + Windows (Ver. 3.1)	3.5-inch 2HD
AB13	IBM PC/AT or compatible machine (Japanese Windows)	Refer to **B.4**.	3.5-inch 2HC
BB13	IBM PC/AT or compatible machine (English Windows)		
3P16	HP9000 Series 700	HP-UX (rel. 9.01)	Digital audio tape (DAT)
3K15	SPARCstation	SunOS (rel. 4.1.1)	Cartridge tape (QIC-24)
3K13			3.5-inch 2HC
3R16	NEWS™ (RISC)	NEWS-OS™ (6.1x)	1/4-inch CGMT
3R13			3.5-inch 2HC
3M15	EWS4800 series (RISC)	EWS-UX/V (rel. 4.0)	Cartridge tape (QIC-24)

Note MS-DS Ver. 5.0 or above has a task swap function, but this function cannot be used with the above software.

B.3.2 Software (3/3)

SD78K/0 Screen Debugger	This debugger is a program which controls the IE-78000-R in-circuit emulator from the host computer. The in-circuit emulator must be connected to the host computer via a serial interface (RS-232-C) cable. This debugger is used together with the DF78064 device file (option).
	Part number : μS××××SD78K0
DF78064[Note] Device File	File containing information unique to the devices. This device file is used together with the SM78K0, CC78K/0, RA78K0, ID78K0, and SD78K/0 (option).
	Part number : μS××××DF78064

Note This device file can be used for any of RA78K/0, CC78K/0, SM78K0, ID78K0 and SD78K/0.

Remark ×××× of the part number differs depending on the host machine and OS used. Refer to the table below.

μS×××× SD78K0
μS×××× DF78064

××××	Host Machine	OS	Supply Medium
5A13	PC-9800 series	MS-DOS (ver. 3.30 to 6.2)[Note]	3.5-inch 2HD
5A10			5-inch 2HD
7B13	IBM PC/AT or compatible machine	Refer to **B.4**.	3.5-inch 2HC
7B10			5-inch 2HC

Note The task swap function is not available with this software though the function is provided in MS-DOS version 5.0 or later.

B.4 Operating Systems for IBM PC

The following operating systems are available for IBM PC.

If SM78K0, ID78K0, and FE9200 (refer to **C.2 Fuzzy Inference Development Support System**) are to be operated, Windows version 3.0 or 3.1 is also required.

OS	Version
PC DOS	Version 5.02 through 6.3
	J6.1/V through J6.3/V[Note]
IBM DOS™	J5.02/V[Note]
MS-DOS	Version 5.0 through 6.22
	5.0/V[Note] through 6.2V[Note]

Note Supports English versions only.

Caution The task swap function is not available with this software though the function is provided in MS-DOS version 5.0 or later.

B.5 System-Up Method from Other In-Circuit Emulator to 78K/0 Series In-Circuit Emulator

When you already have an in-circuit emulator for the 78K series or the 75X/XL series, you can use that in-circuit emulator as the equivalent of a 78K/0 series in-circuit emulator IE-78000-R or IE-78000-R-A by replacing the internal break board with the IE-78000-R-BK.

Table B-1. System-Up Method from Other In-Circuit Emulator to IE-78000-R

Series Name	In-circuit Emulator Owned	Board to be Purchased
75X/XL Series	IE-75000-R[Note], IE-75001-R	IE-78000-R-BK
78K/I Series	IE-78130-R, IE-78140-R	
78K/II Series	IE-78230-R[Note], IE-78230-R-A IE-78240-R[Note], IE-78240-R-A	
78K/III Series	IE-78320-R[Note], IE-78327-R IE-78330-R, IE-78350-R	

Note Available for maintenance use only.

Table B-2. System-Up Method from Other In-Circuit Emulator to IE-78000-R-A

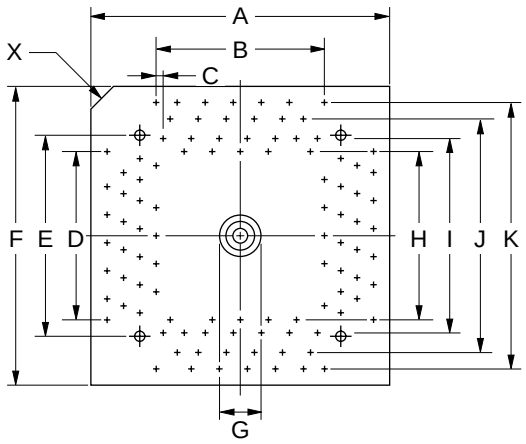
Series Name	In-circuit Emulator Owned	Board to be Purchased
75X/XL Series	IE-75000-R[Note 1], IE-75001-R	IE-78000-R-BK[Note 2]
78K/I Series	IE-78130-R, IE-78140-R	
78K/II Series	IE-78230-R[Note 1], IE-78230-R-A IE-78240-R[Note 1], IE-78240-R-A	
78K/III Series	IE-78320-R[Note 1], IE-78327-R IE-78330-R, IE-78350-R	
78K/0 Series	IE-78000-R	—[Note 2]

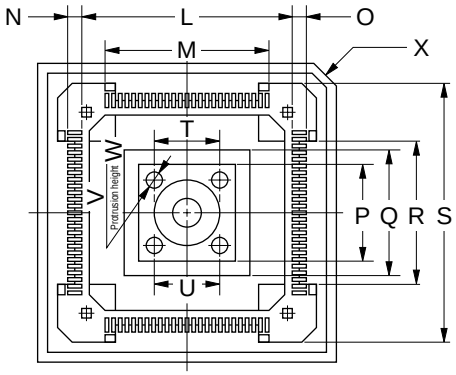
Notes 1. Available for maintenance use only.

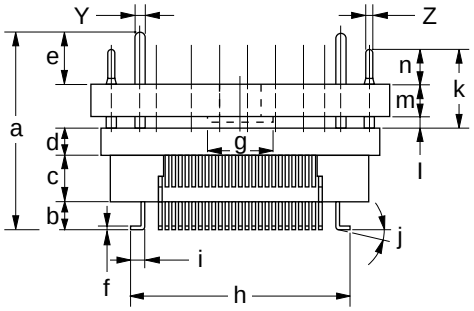
2. It is necessary to submit the board to modify a part of in-circuit emulator main unit to adapt control/trace board for superviser board.

Drawing for Conversion Adapter (TGC-100SDW)

★ **Figure B-2. Conversion Adapter (TGC-100SDW) Drawing (For Reference Only)**







ITEM	MILLIMETERS	INCHES
A	21.55	0.848
B	0.5x24=12	0.020x0.945=0.472
C	0.5	0.020
D	0.5x24=12	0.020x0.945=0.472
E	15.0	0.591
F	21.55	0.848
G	ϕ3.55	ϕ0.140
H	10.9	0.429
I	13.3	0.524
J	15.7	0.618
K	18.1	0.713
L	13.75	0.541
M	0.5x24=12.0	0.020x0.945=0.472
N	1.125±0.3	0.044±0.012
O	1.125±0.2	0.044±0.008
P	7.5	0.295
Q	10.0	0.394
R	11.3	0.445
S	18.1	0.713
T	ϕ5.0	ϕ0.197
U	5.0	0.197
V	4-ϕ1.3	4-ϕ0.051
W	1.8	0.071
X	C 2.0	C 0.079
Y	ϕ0.9	ϕ0.035
Z	ϕ0.3	ϕ0.012

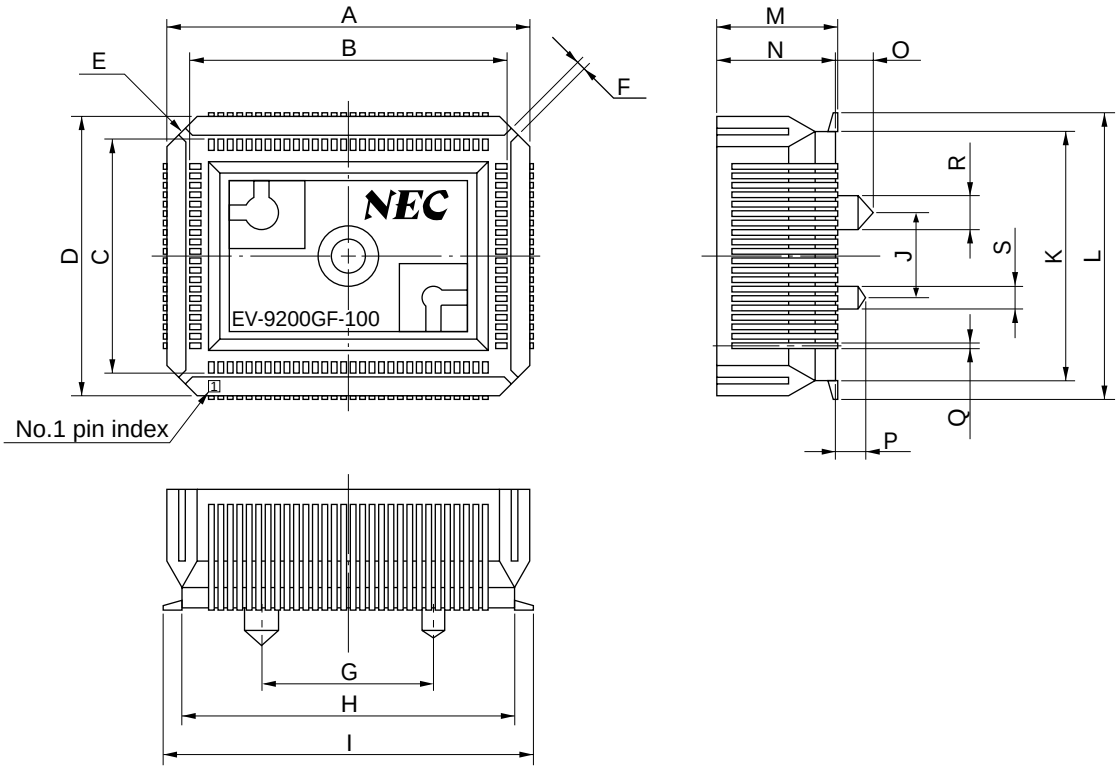
ITEM	MILLIMETERS	INCHES
a	14.45	0.569
b	1.85±0.25	0.073±0.010
c	3.5	0.138
d	2.0	0.079
e	3.9	0.154
f	0.25	0.010
g	ϕ4.5	ϕ0.177
h	16.0	0.630
i	1.125±0.3	0.044±0.012
j	0~5°	0.000~0.197°
k	5.9	0.232
l	0.8	0.031
m	2.4	0.094
n	2.7	0.106

TGC-100SDW-G1E

note: Product by TOKYO ELETECH CORPORATION.

Drawing and Recommended Print Board Mounting for Conversion Socket (EV-9200GF-100)

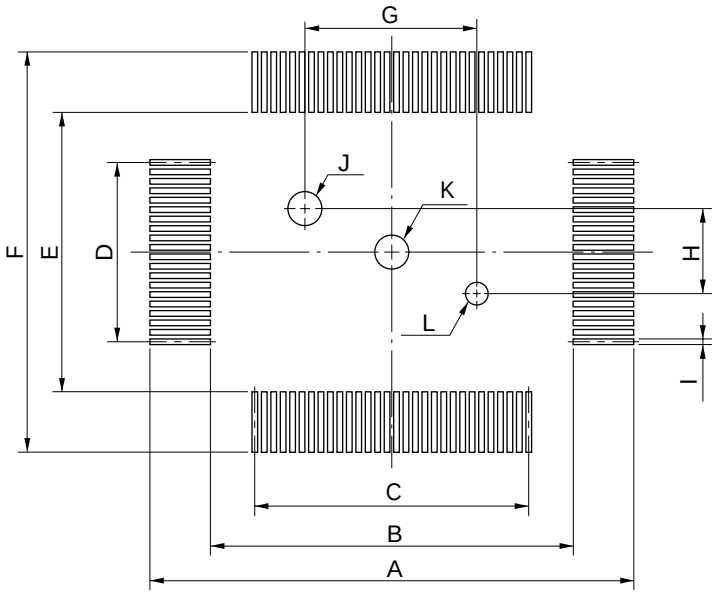
Figure B-3. Conversion Socket (EV-9200GF-100) Drawing (For Reference Only)



EV-9200GF-100-G0E

ITEM	MILLIMETERS	INCHES
A	24.6	0.969
B	21	0.827
C	15	0.591
D	18.6	0.732
E	4-C 2	4-C 0.079
F	0.8	0.031
G	12.0	0.472
H	22.6	0.89
I	25.3	0.996
J	6.0	0.236
K	16.6	0.654
L	19.3	076
M	8.2	0.323
N	8.0	0.315
O	2.5	0.098
P	2.0	0.079
Q	0.35	0.014
R	ϕ2.3	ϕ0.091
S	ϕ1.5	ϕ0.059

Figure B-4. Conversion Socket (EV-9200GF-100) Recommended Print Board Mounting Pattern (For Reference Only)



EV-9200GF-100-P1E

ITEM	MILLIMETERS	INCHES
A	26.3	1.035
B	21.6	0.85
C	$0.65\pm0.02\times29=18.85\pm0.05$	$0.026^{+0.001}_{-0.002}\times1.142=0.742^{+0.002}_{-0.002}$
D	$0.65\pm0.02\times19=12.35\pm0.05$	$0.026^{+0.001}_{-0.002}\times0.748=0.486^{+0.003}_{-0.002}$
E	15.6	0.614
F	20.3	0.799
G	12 ± 0.05	$0.472^{+0.003}_{-0.002}$
H	6 ± 0.05	$0.236^{+0.003}_{-0.002}$
I	0.35 ± 0.02	$0.014^{+0.001}_{-0.001}$
J	$\phi2.36\pm0.03$	$\phi0.093^{+0.001}_{-0.002}$
K	$\phi2.3$	$\phi0.091$
L	$\phi1.57\pm0.03$	$\phi0.062^{+0.001}_{-0.002}$

Caution Dimensions of mount pad for EV-9200 and that for target device (QFP) may be different in some parts. For the recommended mount pad dimensions for QFP, refer to "SEMICONDUCTOR DEVICE MOUNTING TECHNOLOGY MANUAL" (C10535E).

[MEMO]

APPENDIX C EMBEDDED SOFTWARE

This section describes the embedded software which are provided for the μPD78064B subseries to allow users to develop and maintain the application program for these subseries.

C.1 Real-time OS (1/2)

RX78K/0 Real-Time OS	RX78K/0 is a real-time OS which is based on the μITRON specification. Supplied with the RX78K/0 nucleus and a tool to prepare multiple information tables (configurator). When using the RX78K/0, the RA78K/0 assembler package and DF78064 device file (option) are necessary.
	Part number: μS××××RX78013-ΔΔΔΔ

Caution When purchasing the RX78K/0, fill in the purchase application form in advance, and sign the Use Approval Contract.

Remark ×××× and ΔΔΔΔ of the part number differs depending on the host machine and OS used. Refer to the table below.

μS××××RX78013-ΔΔΔΔ

ΔΔΔΔ	Product Outline	Max. No. for Use in Mass Production
001	Evaluation object	Do not use for mass production
100K	Mass-production object	100,000
001M		1,000,000
010M		10,000,000
S01	Source program	Source program for mass-production object

××××	Host Machine	OS	Supply Medium
5A13	PC-9800 series	MS-DOS (ver. 3.30 to 6.2)[Note]	3.5-inch 2HD
5A10			5-inch 2HD
7B13	IBM PC/AT or compatible machine	Refer to **B.4**.	3.5-inch 2HC
7B10			5-inch 2HC
3H15	HP9000 series 300	HP-UX (rel.7.05B)	Cartridge tape (QIC-24)
3P16	HP9000 series 700	HP-UX (rel.9.01)	Digital tape (DAT)
3K15	SPARCstation	SunOS (rel.4.1.1)	Cartridge tape (QIC-24)
3M15	EWS4800 series (RISC)	EWS-UX/V (rel.4.0)	

Note The task swap function is not available with this software though the function is provided in MS-DOS version 5.0 or later.

C.1 Real-time OS (2/2)

MX78K0 OS	MX78K/0 is an OS for subsets based on the μITRON specification. Supplied with the MX78K0 nucleus. This OS manages tasks, events, and time. In task management operation, it controls the execution orders of tasks, and switches processing to the task to be executed next.
	Part number: μS××××MX78K0-ΔΔΔ

Remark ×××× and ΔΔΔ of the part number differs depending on the host machine and OS used. Refer to the table below.

μS××××MX78K0-ΔΔΔ

ΔΔΔ	Product outline	Remark
001	Evaluation object	Use for preproduction.
xx	Mass-production object	Use for mass-production.
S01	Source program	Available only when purchasing mass-production object

××××	Host Machine	OS	Supply Medium
5A13	PC-9800 series	MS-DOS (ver. 3.30 to 6.2)**Note**	3.5-inch 2HD
5A10			5-inch 2HD
7B13	IBM PC/AT or compatible machine	Refer to **B.4**.	3.5-inch 2HC
7B10			5-inch 2HC
3H15	HP9000 series 300	HP-UX (rel.7.05B)	Cartridge tape (QIC-24)
3P16	HP9000 series 700	HP-UX (rel.9.01)	Digital tape (DAT)
3K15	SPARCstation	SunOS (rel.4.1.1)	Cartridge tape (QIC-24)
3M15	EWS4800 series (RISC)	EWS-UX/V (rel.4.0)	

Note The task swap function is not available with this software though the function is provided in MS-DOS version 5.0 or later.

C.2 Fuzzy Inference Development Support System

FE9000/FE9200 (Fuzzy Knowledge Data Creation tool)	Program supporting input of fuzzy knowledge data (fuzzy rule and membership function), editing (edit), and evaluation (simulation). FE9200 operates on Windows.
	Part number: μS××××FE9000 (PC-9800 series) μS××××FE9200 (IBM PC/AT or compatible machine)
FT9080/FT9085 (Translator)	Program converting fuzzy knowledge data obtained by using fuzzy knowledge data preparation tool to RA78K/0 assembler source program.
	Part number: μS××××FT9080 (PC-9800 series) μS××××FT9085 (IBM PC/AT or compatible machine)
FI78K0 (Fuzzy Inference Module)	Program executing fuzzy inference. Fuzzy inference is executed by linking fuzzy knowledge data converted by translator.
	Part number: μS××××FI78K0 (PC-9800 series, IBM PC/AT or compatible machine)
FD78K0 (Fuzzy Inference Debugger)	Support software evaluating and adjusting fuzzy knowledge data at hardware level by using in-circuit emulator.
	Part number: μS××××FD78K0 (PC-9800 series, IBM PC/AT or compatible machine)

Remark ×××× of the part number differs depending on the host machine and OS used. Refer to the table below.

μS××××FE9000
μS××××FT9080
μS××××FI78K0
μS××××FD78K0

××××	Host Machine	OS	Supply Medium
5A13	PC-9800 series	MS-DOS	3.5-inch 2HD
5A10		(ver. 3.30 to 6.2)**Note**	5-inch 2HD

μS××××FE9200
μS××××FT9085
μS××××FI78K0
μS××××FD78K0

××××	Host Machine	OS	Supply Medium
7B13	IBM PC/AT	Refer to **B.4**.	3.5-inch 2HC
7B10	or compatible machine		5-inch 2HC

Note The task swap function is not available with this software though the function is provided in MS-DOS version 5.0 or later.

[MEMO]

APPENDIX D REGISTER INDEX

D.1 Register Name Index

D.2 Register Symbol Index

[T]

[W]

★

APPENDIX E REVISION HISTORY

Here is the revision history of this manual. “Chapter” indicates the chapters in the old edition where revision has been made in this edition.

Edition	Revision from Previous Edition	Chapter
2nd edition	• Addition of μPD78064B(A) • Addition of 100-pin plastic LQFP (fine pitch) (14 × 14 mm) package to μPD78064B and 78P064B	Throughout
	Addition of 1.4 Quality Grade	CHAPTER 1 OVERVIEW
	Addition of notes on differences between AV_{DD} and AV_{SS} pins to 1.5 Pin Configuration (Top View)	
	Addition of following subseries to 1.6 Development of 78K/0 Series: μPD78075B, 78075BY, 780018, 780018Y, 780058, 780058Y, 780034, 780034Y, 780024, 780024Y, 78014H, 780964, 780924, 780228, 78044H, 78044F, 78098B, 780973, and 780805 subseries	
	Addition of 1.10 Differences between μPD78064B and μPD78064B(A)	
	Addition of notes on AV_{DD} and AV_{SS} pins to 2.1. Normal operating mode pins	CHAPTER 2 PIN FUNCTION
	Addition of Figures 7-10 and 7-13 Timing of Square Wave Output Operation	CHAPTER 7 8-BIT TIMER/ EVENT COUNTER
	Addition of notes on changing operation mode to 13.1 Serial Interface Channel 0 Functions and (2) Serial operation mode register (CSIM0) in 13.3 Serial Interface Channel 0 Control Registers	CHAPTER 13 SERIAL INTERFACE CHANNEL 0
	Addition of notes to (f) Busy signal (BUSY), ready signal (READY) in (2) Definition of SBI in 13.4.3 SBI mode operation	
	Addition of notes to (11) Notes on SBI mode in 13.4.3 (2) (a) Bus release signal (REL), (b) command signal (CMD)	
	Correction of Figure 14-10 Receive Error Timing	CHAPTER 14 SERIAL INTERFACE CHANNEL 2
	Addition of (3) Changing MSB/LSB first to 14.4.3 3-wire serial I/O mode	
	Addition of 14.4.4 Limitations when using UART mode	
	Addition of APPENDIX A DIFFERENCES BETWEEN μPD78064 AND 78064B SUBSERIES	APPENDIX A DIFFERENCES BETWEEN μPD78064 AND 78064B SUBSERIES
	• Addition of PA-78P0308GC, PA-78P0308GF, and IE-780308-R-EM • Change of name of conversion adapter EV-9500GC-100 to TGC-100SDW • Deletion of Windows-compatible 5" FD	APPENDIX B DEVELOPMENT TOOLS
	Addition of APPENDIX E REVISION HISTORY	APPENDIX E REVISION HISTORY

[MEMO]

NEC

Facsimile Message

From:

Name

Company

Tel. FAX

Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: 1-800-729-9288
1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Technical Documentation Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-889-1689

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

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

Taiwan
NEC Electronics Taiwan Ltd.
Fax: 02-719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Corporation
Semiconductor Solution Engineering Division
Technical Information Support Dept.
Fax: 044-548-7900

I would like to report the following error/make the following suggestion:

Document title: _____

Document number: _____ Page number: _____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	❑	❑	❑	❑
Technical Accuracy	❑	❑	❑	❑
Organization	❑	❑	❑	❑