USER'S MANUAL

NEC

μPD78083 SUBSERIES

8-BIT SINGLE-CHIP MICROCONTROLLER

μPD78081 μPD78081(A)
μPD78082 μPD78082(A)
μPD78P083 μPD78P083(A)
μPD78P081(A2)

Document No. U12176EJ2V0UM00 (2nd edition)
(O. D. No. IEU-886)
Date Published May 1997 N
Printed in Japan

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

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

License not needed: μPD78P083DU
The customer must judge the need for license:
μPD78081CU-×××, 78081GB-×××-3B4, 78081GB-×××-3BS-MTX
μPD78081GB(A)-×××-3B4, 78081GB(A2)-×××-3B4
μPD78082CU-×××, 78082GB-×××-3B4, 78082GB-×××-3BS-MTX
μPD78082GB(A)-×××-3B4
μPD78P083CU, 78P083GB-3B4, 78P083GB-3BS-MTX
μPD78P083CU(A), 78P083GB(A)-3B4, 78P083GB(A)-3BS-MTX

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

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

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

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

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

NEC devices are classified into the following three quality grades:

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

- Standard: Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots
- Special: Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support)
- Specific: Aircrafts, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems or medical equipment for life support, etc.

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

Anti-radioactive design is not implemented in this product.

Regional Information

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

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

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

NEC Electronics Inc. (U.S.)
Santa Clara, California
Tel: 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 Revision in This Edition

Page	Description
Throughout	The following products have been already developed μPD78081CU-×××, 78081GB-×××-3B4, 78082CU-×××, 78082GB-×××-3B4, 78P083CU, 78P083DU, 78P083GB-3B4
	The following products have been added μPD78081GB-×××-3BS-MTX, 78082GB-×××-3BS-MTX, 78P083GB-3BS-MTX, 78081GB(A)-×××-3B4, 78082GB(A)-×××-3B4, 78P083CU(A), 78P083GB(A)-3B4, 78P083GB(A)-3BS-MTX, 78081GB(A2)-×××-3B4
	Changes supply voltage to V_{DD} = 1.8 to 5.5V.
p. 9	**1.6 78K/0 Series Development** has been changed.
p. 13	**1.9 Differences between the μPD78081, 78082, and 78P083, the μPD78081(A), 78082(A), and 78P083(A), and the μPD78081(A2)** has been added.
p. 19	Cautions regarding the use of functions in common with **2.2.5 (2) (d) ASCK** has been added.
p. 72	Cautions concerning the Write to OSMS Command has been added to **5.3 (2) Oscillation mode select register (OSMS).**
p. 73	Cautions concerning external clock input in **5.4.1 Main system clock oscillator** has been changed.
p. 108	**Figure 7-3. Watchdog Timer Mode Register Format**, notes and cautions have been added.
p. 110	Description of **7.4.2 Interval timer operation** has been changed.
p. 113	Cautions with regard to rewriting TCL0 to other than same data has been added to **8.3 (1) Timer clock select register 0 (TCL0)**.
p. 120	The HSC bit has been added to the A/D Converter Mode Register in **Figure10-1. A/D Converter Block Diagram.**
p. 122, 193	**10.3 (1) A/D converter mode register (ADM), 13.1.1 Standby function,** and **Cautions** have been added.
p. 137	**Figure 11-1. Serial Interface Channel 2 Block Diagram** has been corrected.
p. 146, 155	**11.3 (4) (a), 11.4.2 (1) (d) (i) Generation of baud rate transmit/receive clock by means of main system clock** have been added. 76800 bps has been added to baud rate generated from the main system clock.
p. 161	**Figure 11-10. Receive Error Timing** has been corrected.
p. 165	**11.4.3 (1) (c) Baud rate generator control register (BRGC)** has been added.
p. 168	**11.4.3 (3) MSB/LSB switching as start bit** has been added.
p. 206	**15.1 Memory Size Switching Register** has been changed from W to R/W.
p. 205	Items and cautions have been added to **Table 15-1. Differences between the μPD78P083 and Mask ROM Versions.**
p. 214	A description of the QTOP microcontroller has been added to **15.5 Screening of One-Time PROM Versions.**
p. 232	**Figure A-1. Development Tool Configuration** has been changed.
p. 231	**APPENDIX A DEVELOPMENT TOOLS** The following Development Tools have been added: IE-78000-R-A, IE-70000-98-IF-B, IE-70000-98N-IF, IE-70000-PC-IF-B, IE-78000-R-SV3, SM78K0, ID78K0
p. 239	**A.4 OS for IBM PC** has been added.
p. 240	**Table A-2. System-Up Method from Other In-Circuit Emulator to IE-78000-R-A** has been added.
p. 244	**B.1 Real-time OS** has been added.
p. 249	**APPENDIX D REVISION HISTORY** has been added.

The mark ★ shows major revised points.

PREFACE

Readers

This manual has been prepared for user engineers who want to understand the functions of the μPD78083 subseries and design and develop its application systems and programs.

Caution

In the μPD78083 Subseries, the μPD78P083DU is not designed to maintain the reliability required for use in customers' mass-produced equipment. Please use this device only for experimentation or for evaluation of functions.

Purpose

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

Organization

The μPD78083 subseries manual is separated into two parts: this manual and the instruction edition (common to the 78K/0 Series).

μPD78083 Subseries User's Manual (This Manual)	78K/0 Series User's Manual Instruction
● Pin functions	● CPU functions
● Internal block functions	● Instruction set
● Interrupt	● Explanation of each instruction
● Other on-chip peripheral functions	

How to Read This Manual

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

○ For those who will be using this as a manual for the μPD78081(A), 78082(A), 78P083(A) and 78081(A2):
→ The μPD78081, 78082, 78P083 are explained as being representative devices.
In case this is used as a manual for the μPD78081(A), 78082(A), 78P083(A), or 78081(A2), please reread the product names as follows.
μPD78081 → μPD78081(A) or μPD78081(A2)
μPD78082 → μPD78082(A)
μPD78P083 → μPD78P083(A)

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

○ To know the μPD78083 Subseries instruction function in detail:
→ Refer to the **78K/0 Series User's Manual: Instructions (IEU-1372)**

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

○ To learn the function of a register whose register name is known:
→ Refer to **Appendix C Register Index**.

○ To know the electrical specifications of the μPD78083 Subseries:
→ Refer to separately available Data Sheet.

○ To know application examples of the functions provided in the μPD78083 Subseries:
→ Refer to Application Note separately provided.

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
Remarks	:	Additional explanatory material
Numeral representations	:	Binary ... ×××× or ××××B
		Decimal ... ××××
		Hexadecimal ... ××××H

Examples of use in this manual are prepared for “Standard” quality level devices for general electronic equipment. In the case of examples of use in this manual for devices which meet “Special” quality level requirements, please use each device only after studying each part that is actuall to be used, the circuitry and the quality level of each component before use.

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

● **Related documents for μPD78054 subseries**

Document name		Document No.	
		Japanese	English
μPD78083 Subseries User's Manual		U12176J	This Manual
μPD78081, 78082 Data Sheet		U11415J	U11415E
μPD78P083 Data Sheet		U11006J	U11006E
μPD78081(A), 78082(A), 78081(A2) Data Sheet		In preparation	To be prepared
μPD78P083(A) Data Sheet		U12175J	U12175E
μPD78083 Subseries Special Function Register Table		IEM-5599	—
78K/0 Series User's Manual—Instruction		IEU-849	IEU-1372
78K/0 Series Instruction Table		U10903J	—
78K/0 Series Instruction Set		U10904J	—
78K/0 Series Application Note	Basics (III)	IEA-767	U10182E

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

● **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	Structured assembly language	U11789J	U11789E
	Assembly language	U11801J	U11801E
	Operation	U11802J	U11802E
CC78K Series C Compiler	Operation	EEU-656	EEU-1280
	Language	EEU-655	EEU-1284
CC78K/0 C Compiler	Operation	U11517J	U11517E
	Language	U11518J	U11518E
CC78K/0 C Compiler Application Note	Programming know-how	EEA-618	EEA-1208
CC78K Series Library Source File		EEU-777	—
PG-1500 PROM Programmer		U11940J	EEU-1335
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		EEU-810	U11376E
IE-78000-R-A		U10057J	U10057E
IE-78000-R-BK		EEU-867	EEU-1427
IE-78078-R-EM		U10775J	U10775E
EP-78083		EEU-5003	EEU-1529
SM78K0 System Simulator Windows™ Base	Reference	U10181J	U10181E
SM78K Series System Simulator	External component user open interface specifications	U10092J	U10092E
ID78K0 Integrated Debugger EWS Base	Reference	U11151J	—
ID78K0 Integrated Debugger PC Base	Reference	U11539J	—
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	Basics	U11537J	—
	Installation	U11536J	—
	Technicals	U11538J	—
OS for 78K/0 Series MX78K0	Basics	EEU-5010	—
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	C11893E
Microcontroller 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

FIGURE (1/4)

FIGURE (2/4)

FIGURE (3/4)

FIGURE (4/4)

TABLE (1/2)

TABLE (2/2)

[MEMO]

CHAPTER 1 OUTLINE

1.1 Features

○ On-chip ROM and RAM

Type / Part Number	Program Memory (ROM)	Data Memory (Internal High-Speed RAM)
μPD78081	8 Kbytes	256 bytes
μPD78082	16 Kbytes	384 bytes
μPD78P083	24 Kbytes (Note)	512 bytes (Note)

Note The capacities of internal PROM and internal high-speed RAM can be changed by means of the memory size switching register (IMS).

○ Instruction execution time changeable from high speed (0.4 μs: In main system clock 5.0 MHz operation) to low speed (12.8 μs: In main system clock 5.0 MHz operation)
○ Instruction set suited to system control
  • Bit manipulation possible in all address spaces
  • Multiply and divide instructions
○ 33 I/O ports
○ 8-bit resolution A/D converter: 8 channels
○ Serial interface: 1 channel
  • 3-wire serial I/O/UART mode: 1 channel
○ Timer: 3 channels
  • 8-bit timer/event counter : 2 channels
  • Watchdog timer : 1 channel
○ Vectored Interrupt Source : 13
★ ○ Supply voltage: V_{DD} = 1.8 to 5.5 V

1.2 Applications

μPD78081, 78082, 78P083:

Airbags, CRT displays, keyboards, air conditioners, hot water dispensers, boilers, fan heaters, dashboards, etc.

★ **μPD78081(A), 78082(A), 78P083(A), 78081(A2):**

Automobile electrical control devices, gas detector cutoff devices, various safety devices, etc.

★ 1.3 Ordering Information

Part number	Package	Internal ROM
μPD78081CU-×××	42-pin plastic shrink DIP (600 mil)	Mask ROM
μPD78081GB-×××-3B4	44-pin plastic QFP (10 × 10 mm)	Mask ROM
μPD78081GB-×××-3BS-MTX	44-pin plastic QFP (10 × 10 mm)	Mask ROM
μPD78082CU-×××	42-pin plastic shrink DIP (600 mil)	Mask ROM
μPD78082GB-×××-3B4	44-pin plastic QFP (10 × 10 mm)	Mask ROM
μPD78082GB-×××-3BS-MTX	44-pin plastic QFP (10 × 10 mm)	Mask ROM
μPD78P083CU	42-pin plastic shrink DIP (600 mil)	One-Time PROM
μPD78P083DU	42-pin ceramic shrink DIP (with window) (600 mil)	EPROM
μPD78P083GB-3B4	44-pin plastic QFP (10 × 10 mm)	One-Time PROM
μPD78P083GB-3BS-MTX	44-pin plastic QFP (10 × 10 mm)	One-Time PROM
μPD78081GB(A)-×××-3B4	44-pin plastic QFP (10 × 10 mm)	Mask ROM
μPD78082GB(A)-×××-3B4	44-pin plastic QFP (10 × 10 mm)	Mask ROM
μPD78P083CU(A)	42-pin plastic shrink DIP (600 mil)	One-Time PROM
μPD78P083GB(A)-3B4	44-pin plastic QFP (10 × 10 mm)	One-Time PROM
μPD78P083GB(A)-3BS-MTX**Note**	44-pin plastic QFP (10 × 10 mm)	One-Time PROM
μPD78081GB(A2)-×××-3B4	44-pin plastic QFP (10 × 10 mm)	Mask ROM

Note Under development

Remark ××× indicates ROM code suffix.

★ 1.4 Quality Grade

Part number	Package	Quality grade
μPD78081CU-×××	42-pin plastic shrink DIP (600 mil)	Standard
μPD78081GB-×××-3B4	44-pin plastic QFP (10 × 10 mm)	Standard
μPD78081GB-×××-3BS-MTX	44-pin plastic QFP (10 × 10 mm)	Standard
μPD78082CU-×××	42-pin plastic shrink DIP (600 mil)	Standard
μPD78082GB-×××-3B4	44-pin plastic QFP (10 × 10 mm)	Standard
μPD78082GB-×××-3BS-MTX	44-pin plastic QFP (10 × 10 mm)	Standard
μPD78P083CU	42-pin plastic shrink DIP (600 mil)	Standard
μPD78P083DU	42-pin ceramic shrink DIP (with window) (600 mil)	Not applicable
μPD78P083GB-3B4	44-pin plastic QFP (10 × 10 mm)	Standard
μPD78P083GB-3BS-MTX	44-pin plastic QFP (10 × 10 mm)	Standard
μPD78081GB(A)-×××-3B4	44-pin plastic QFP (10 × 10 mm)	Special
μPD78082GB(A)-×××-3B4	44-pin plastic QFP (10 × 10 mm)	Special
μPD78P083CU(A)	42-pin plastic shrink DIP (600 mil)	Special
μPD78P083GB(A)-3B4	44-pin plastic QFP (10 × 10 mm)	Special
μPD78P083GB(A)-3BS-MTX[Note]	44-pin plastic QFP (10 × 10 mm)	Special
μPD78081GB(A)-×××-3B4	44-pin plastic QFP (10 × 10 mm)	Special

Note Under planning

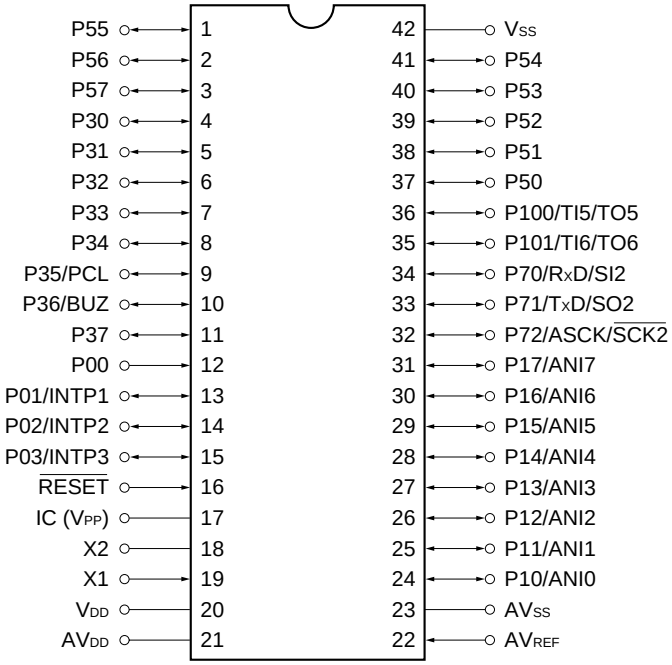
Remark ××× indicates ROM code suffix.

Please refer to “Quality grade on NEC Semiconductor Devices” (Document number 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

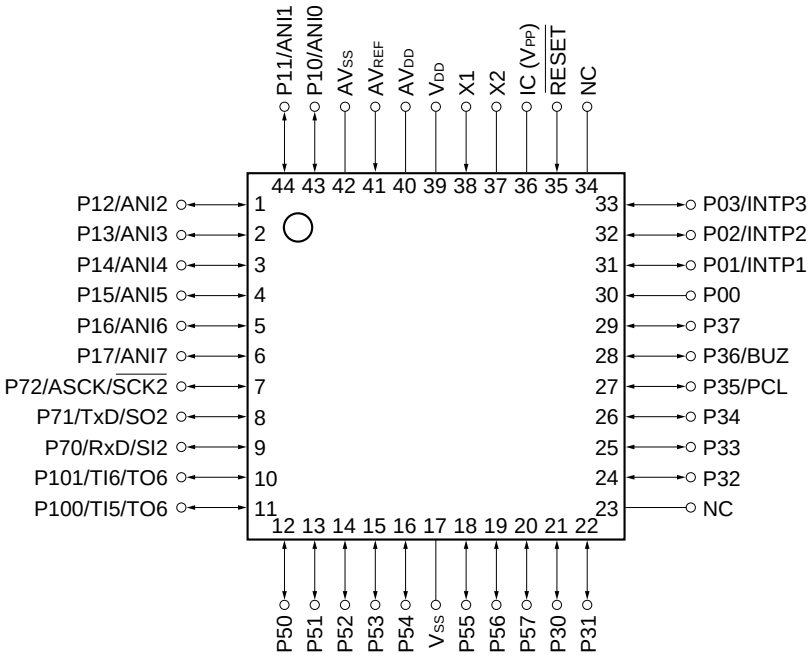
42-pin plastic shrink DIP (600 mil)
μPD78081CU-×××, 78082CU-×××, 78P083CU, 78P083CU(A)
42-pin ceramic shrink DIP (with window) (600 mil)
μPD78P083DU



Cautions 1. Be sure to connect IC (Internally Connected) pin to VSS directly.
2. Connect AVDD pin to VDD.
3. Connect AVSS pin to VSS.

Remark Pin connection in parentheses is intended for the μPD78P083.

- **44-pin plastic QFP (10 × 10 mm)**
  μPD78081GB-×××-3B4, 78081GB-×××-3BS-MTX
  μPD78082GB-×××-3B4, 78082GB-×××-3BS-MTX
  μPD78P083GB-3B4, 78P083GB-3BS-MTX
  μPD78081GB(A)-×××-3B4, 78082GB(A)-×××-3B4
  μPD78P083GB(A)-3B4, 78P083GB(A)-3BS-MTX**Note**
  μPD78P081GB(A2)-×××-3B4



Note Under development

Cautions 1. Be sure to connect IC (Internally Connected) pin to V_{SS} directly.
2. Connect AV_{DD} pin to V_{DD}.
3. Connect AV_{SS} pin to V_{SS}.
4. Connect NC pin to V_{SS} for noise protection (It can be left open).

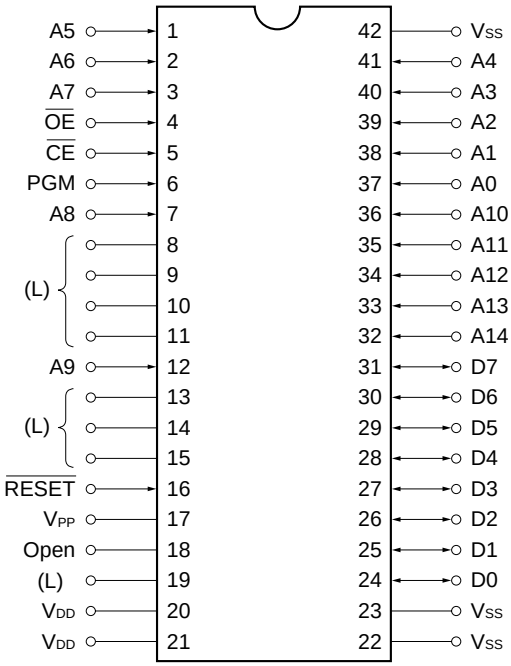
Remark Pin connection in parenthesis is intended for the μPD78P083.

Pin Identifications

ANI0 to ANI7	:	Analog Input	P100, P101	:	Port 10
ASCK	:	Asynchronous Serial Clock	PCL	:	Programmable Clock
AV_{DD}	:	Analog Power Supply	$\overline{RESET}$	:	Reset
AV_{REF}	:	Analog Reference Voltage	RxD	:	Receive Data
AV_{SS}	:	Analog Ground	$\overline{SCK2}$	:	Serial Clock
BUZ	:	Buzzer Clock	SI2	:	Serial Input
IC	:	Internally Connected	SO2	:	Serial Output
INTP1 to INTP3	:	Interrupt from Peripherals	TI5, TI6	:	Timer Input
NC	:	Non-connection	TO5 to TO6	:	Timer Output
P00 to P03	:	Port 0	TxD	:	Transmit Data
P10 to P17	:	Port 1	V_{DD}	:	Power Supply
P30 to P37	:	Port 3	V_{PP}	:	Programming Power Supply
P50 to P57	:	Port 5	V_{SS}	:	Ground
P70 to P72	:	Port 7	X1, X2	:	Crystal (Main System Clock)

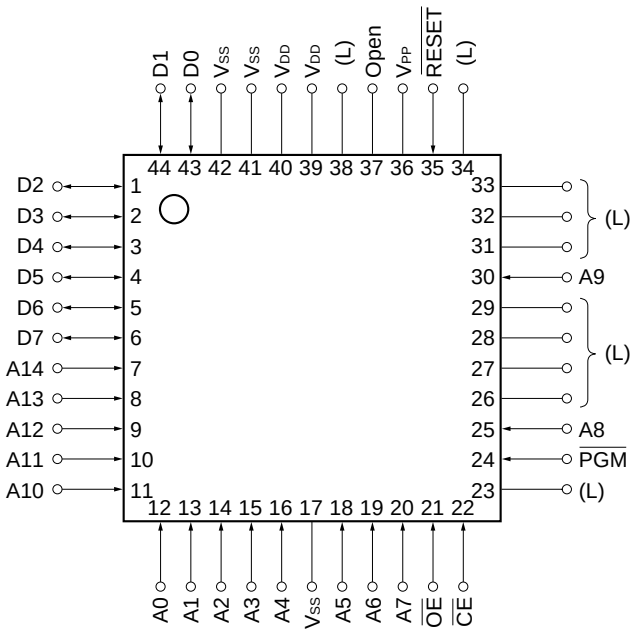
(2) PROM programming mode

- **42-pin plastic shrink DIP (600 mil)**
  μPD78P083CU, 78P083CU(A)
- **42-pin ceramic shrink DIP (with window) (600 mil)**
  μPD78P083DU



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

- **44-pin plastic QFP (10 × 10 mm)**
  μPD78P083GB-3B4, 78P083GB-3BS-MTX
  μPD78P083GB(A)-3B4, 78P083GB(A)-3BS-MTX**Note**



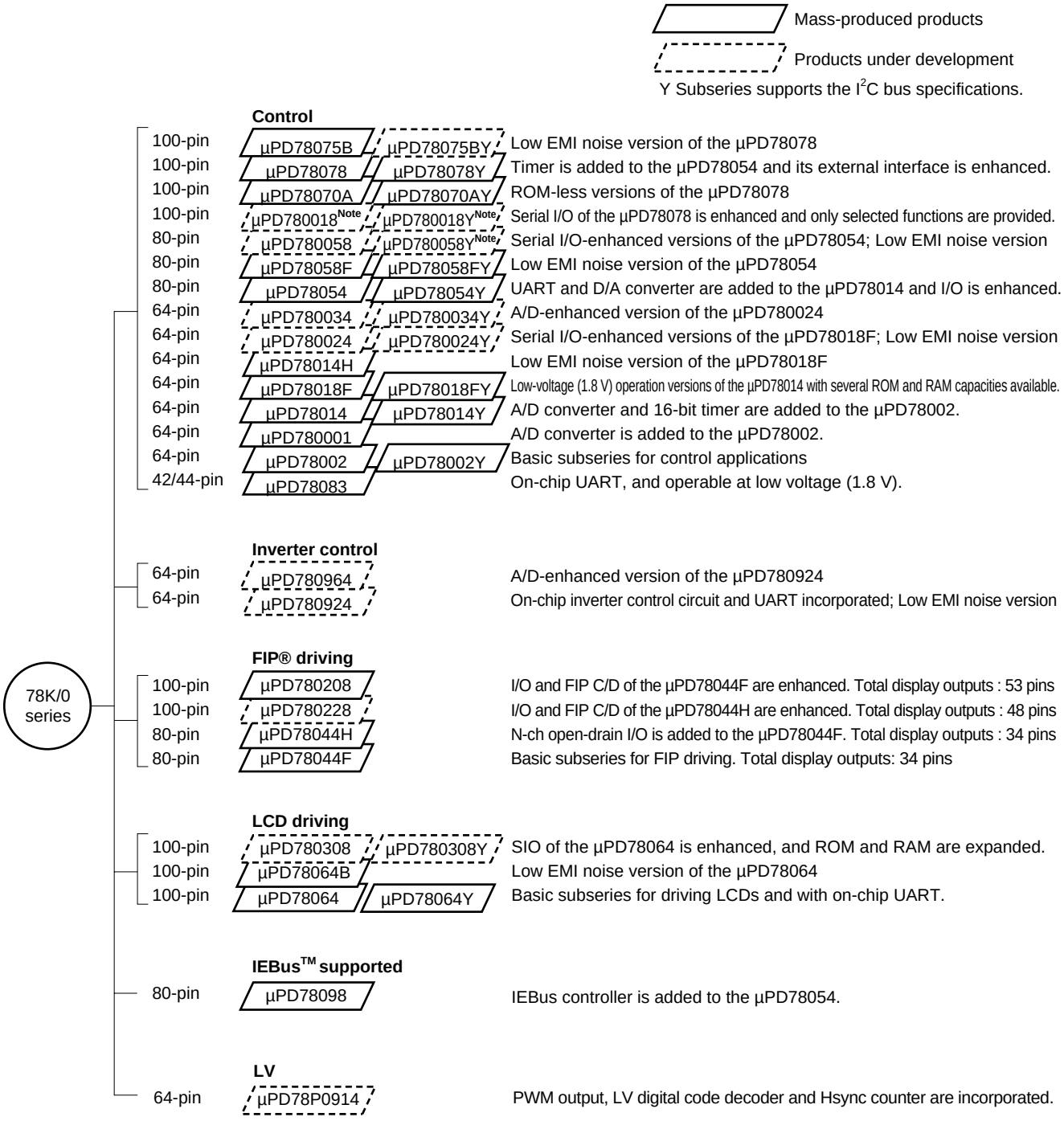
Note Under development

Cautions 1. (L) : Connect individually to VSS via a pull-down resistor.
2. VSS : Connect to the ground.
3. RESET : Set to the low level.
4. Open : Do not connect anything.

A0 to A14 : Address Bus
CE : Chip Enable
D0 to D7 : Data Bus
OE : Output Enable
PGM : Program
RESET : Reset
VDD : Power Supply
VPP : Programming Power Supply
VSS : Ground

1.6 78K/0 Series Development

The following shows the 78K/0 Series products development. Subseries names are shown inside frames.



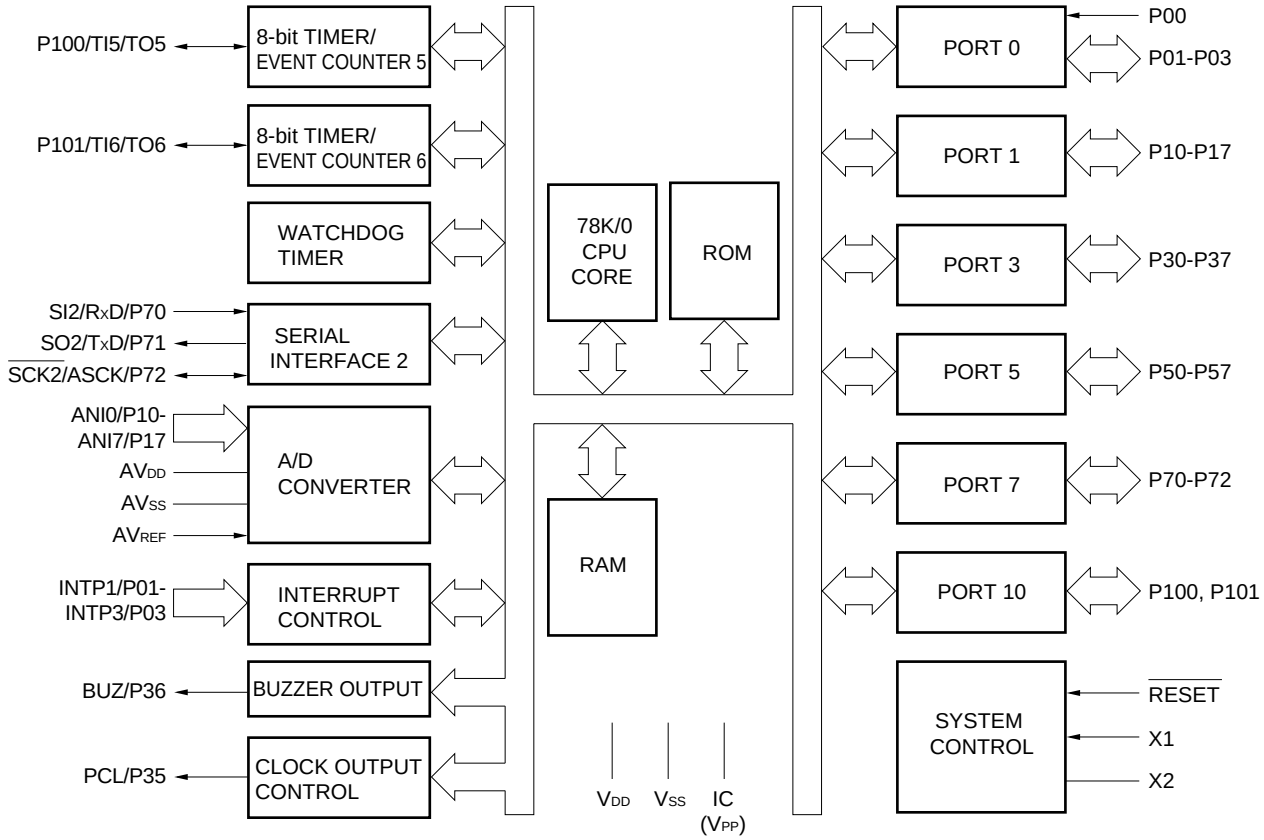
Note Under planning

The following table shows the differences among subseries functions.

Subseries name \ Function		ROM capacity	Timer 8-bit	Timer 16-bit	Timer Watch	Timer WDT	8-bit A/D	10-bit A/D	8-bit D/A	Serial interface	I/O	V_{DD} MIN. value	External expansion
Control	μPD78075B	32K to 40K	4 ch	1 ch	1 ch	1 ch	8 ch	—	2 ch	3 ch (UART: 1 ch)	88	1.8 V	Available
	μPD78078	48K to 60K											
	μPD78070A	—									61	2.7 V	
	μPD780018	48K to 60K							—	2 ch (Time division 3-wire: 1 ch)	88		
	μPD780058	24K to 60K	2 ch						2 ch	3 ch (Time division UART: 1 ch)	68	1.8 V	
	μPD78058F	48K to 60K								3 ch (UART: 1 ch)	69	2.7 V	
	μPD78054	16K to 60K										2.0 V	
	μPD780034	8K to 32K					—	8 ch	—	3 ch (UART: 1 ch, Time division 3-wire: 1 ch)	51	1.8 V	
	μPD780024						8 ch	—					
	μPD78014H									2 ch	53		
	μPD78018F	8K to 60K											
	μPD78014	8K to 32K										2.7 V	
	μPD780001	8K		—	—					1 ch	39		—
	μPD78002	8K to 16K			1 ch		—				53		Available
	μPD78083				—		8 ch			1 ch (UART: 1 ch)	33	1.8 V	—
Inverter control	μPD780964	8K to 32K	3 ch	**Note**	—	1 ch	—	8 ch	—	2 ch (UART: 2 ch)	47	2.7 V	Available
	μPD780924						8 ch	—					
FIP driving	μPD780208	32K to 60K	2 ch	1 ch	1 ch	1 ch	8 ch	—	—	2 ch	74	2.7 V	—
	μPD780228	48K to 60K	3 ch	—	—					1 ch	72	4.5 V	
	μPD78044H	32K to 48K	2 ch	1 ch	1 ch						68	2.7 V	
	μPD78044F	16K to 40K								2 ch			
LCD driving	μPD780308	48K to 60K	2 ch	1 ch	1 ch	1 ch	8 ch	—	—	3 ch (Time division UART: 1 ch)	57	2.0 V	—
	μPD78064B	32K								2 ch (UART: 1 ch)			
	μPD78064	16K to 32K											
IEBus supported	μPD78098	32K to 60K	2 ch	1 ch	1 ch	1 ch	8 ch	—	2 ch	3 ch (UART: 1 ch)	69	2.7 V	Available
LV	μPD78P0914	32K	6 ch	—	—	1 ch	8 ch	—	—	2 ch	54	4.5 V	Available

Note 10 bits timer: 1 channel

1.7 Block Diagram



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

2. Pin connection in parentheses is intended for the μPD78P083.

1.8 Outline of Function

Item \ Part Number		μPD78081	μPD78082	μPD78083
Internal memory	ROM	Mask ROM		PROM
		8 Kbytes	16 Kbytes	24 Kbytes[Note]
	High-speed RAM	256 bytes	384 bytes	512 bytes[Note]
Memory space		64 Kbytes		
General register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)		
Instruction cycle		Instruction execution time variable function is integrated. 0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (@5.0-MHz operation with main system clock)		
Instruction set		• 16-bit operation • Multiply/divide (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjust, etc.		
I/O ports		Total : 33 • CMOS input : 1 • CMOS input/output : 32		
A/D converter		• 8-bit resolution × 8 channels		
Serial interface		• 3-wire serial I/O/UART mode selectable: 1 channel		
Timer		• 8-bit timer/event counter: 2 channels • Watchdog timer: 1 channel		
Timer output		2 pins (8-bit PWM output enable)		
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, and 5.0 MHz (@ 5.0-MHz operation with main system clock)		
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, and 9.8 kHz (@ 5.0-MHz operation with main system clock)		
Vectored interrupt source	Maskable	Internal : 8 external : 3		
	Non-maskable	Internal : 1		
	Software	1		
★ Supply voltage		V_{DD} = 1.8 to 5.5 V		
Operating ambient temperature		T_A = –40 to +85°C		
Package		• 42-pin plastic shrink DIP (600 mil) • 44-pin plastic QFP (10 × 10 mm) • 42-pin ceramic shrink DIP (with window) (600 mil) (μPD78P083 only)		

Note Internal PROM and high-speed RAM capacities can be changed by setting the internal memory size switching register (IMS).

★ 1.9 Differences between the μPD78081, 78082 and 78P083, the μPD78081(A), 78082(A) and 78P083(A), and the μPD78081(A2)

Table 1-1 Differences between the μPD78081, 78082 and 78P083, the μPD78081(A), 78082(A) and 78P083(A), and the μPD78081(A2)

Part Number / Item	μPD78081 μPD78082 μPD78P083	μPD78081(A) μPD78082(A) μPD78P083(A)	μPD78081(A2)
Quality grade	Standard	Special	
Supply voltage	V_{DD} = 1.8 to 5.5 V		V_{DD} = 4.5 to 5.5 V
Operating ambient temperature	T_A = –40 to +85°C		T_A = –40 to +125°C
Electrical specifications	Please refer to the individual data sheets.		

[MEMO]

CHAPTER 2 PIN FUNCTION

2.1 Pin Function List

2.1.1 Normal operating mode pins

(1) Port pins

Pin Name	Input/Output	Function		After Reset	Alternate Function
P00	Input	Port 0	Input only	Input	—
P01	Input/output	4-bit input/output port	Input/output is specifiable bit-wise. When used as the input port, it is possible to connect a pull-up resistor by software.	Input	INTP1
P02					INTP2
P03					INTP3
P10-P17	Input/output	Port 1 8-bit input/output port Input/output is specifiable bit-wise. When used as the input port, it is possible to connect a pull-up resistor by software. Note		Input	ANI0-ANI7
P30-P34	Input/output	Port 3		Input	—
P35		8-bit input/output port			PCL
P36		Input/output is specifiable bit-wise.			BUZ
P37		When used as the input port, it is possible to connect a pull-up resistor by software.			—
P50-P57	Input/output	Port 5 8-bit input/output port A maximum of 7 out of 8 ports can drive LEDs directly. Input/output is specifiable bit-wise. When used as the input port, it is possible to connect a pull-up resistor by software.		Input	—
P70	Input/output	Port 7		Input	SI2/RxD
P71		3-bit input/output port			SO2/TxD
P72		Input/output is specifiable bit-wise.			$\overline{SCK2}$/ASCK
		When used as the input port, it is possible to connect a pull-up resistor by software.			
P100	Input/output	Port 10		Input	TI5/TO5
P101		2-bit input/output port			TI6/TO6
		Input/output is specifiable bit-wise.			
		When used as the input port, it is possible to connect a pull-up resistor by software.			

Note When P10/ANI0-P17/ANI7 pins are used as the analog inputs for the A/D converter, set the port 1 to the input mode. The on-chip pull-up resistor is automatically disabled.

(2) Pins other than port pins

Pin Name	Input/Output	Function	After Reset	Alternate Function
INTP1	Input	External interrupt request input by which the active edge (rising edge, falling edge, or both rising and falling edges) can be specified.	Input	P01
INTP2				P02
INTP3				P03
SI2	Input	Serial interface serial data input.	Input	P70/RxD
SO2	Output	Serial interface serial data output.	Input	P71/TxD
$\overline{SCK2}$	Input/output	Serial interface serial clock input/output.	Input	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/$\overline{SCK2}$
TI5	Input	External count clock input to 8-bit timer (TM5).	Input	P100/TO5
TI6		External count clock input to 8-bit timer (TM6).		P101/TO6
TO5	Output	8-bit timer output. (also used for 8-bit PWM output)	Input	P100/TI5
TO6				P101/TI6
PCL	Output	Clock output. (for main system clock trimming)	Input	P35
BUZ	Output	Buzzer output.	Input	P36
ANI0-ANI7	Input	A/D converter analog input.	Input	P10-P17
AV_{REF}	Input	A/D converter reference voltage input.	–	–
AV_{DD}	–	A/D converter analog power supply. Connected to V_{DD}.	–	–
AV_{SS}	–	A/D converter ground potential. Connected to V_{SS}.	–	–
$\overline{RESET}$	Input	System reset input.	–	–
X1	Input	Main system clock oscillation crystal connection.	–	–
X2	–		–	–
V_{DD}	–	Positive power supply.	–	–
V_{PP}	–	High-voltage applied during program write/verification. Connected directly to V_{SS} in normal operating mode.	–	–
V_{SS}	–	Ground potential.	–	–
IC	–	Internal connection. Connect directly to V_{SS}.	–	–
NC	–	Does not internally connected. Connect to V_{SS}. (It can be left open)	–	–

2.1.2 PROM programming mode pins (μPD78P083 only)

Pin Name	Input/Output	Function
$\overline{RESET}$	Input	PROM programming mode setting. When +5 V or +12.5 V is applied to the V_{PP} pin or a low level voltage is applied to the $\overline{RESET}$ pin, the PROM programming mode is set.
V_{PP}	Input	High-voltage application for PROM programming mode setting and program write/verify.
A0 to A14	Input	Address bus
D0 to D7	Input/output	Data bus
$\overline{CE}$	Input	PROM enable input/program pulse input
$\overline{OE}$	Input	Read strobe input to PROM
$\overline{PGM}$	Input	Program/program inhibit input in PROM programming mode
V_{DD}	—	Positive power supply
V_{SS}	—	Ground potential

2.2 Description of Pin Functions

2.2.1 P00 to P03 (Port 0)

These are 4-bit input/output ports. Besides serving as input/output ports, they function as an external interrupt request input.

The following operating modes can be specified bit-wise.

(1) Port mode

P00 functions as input-only port and P01 to P03 function as input/output ports.

P01 to P03 can be specified for input or output ports bit-wise with a port mode register 0 (PM0). When they are used as input ports, on-chip pull-up resistors can be used to them by defining the pull-up resistor option register L (PUOL).

(2) Control mode

INTP1 to INTP3 function as external interrupt request input pins which are capable of specifying the valid edges (rising edge, falling edge, and both rising and falling edges).

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, on-chip 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 on-chip pull-up resistor is automatically disabled when the pins specified for analog input.

2.2.3 P30 to P37 (Port 3)

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

The following operating modes can be specified bit-wise.

(1) Port mode

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

(2) Control mode

These ports function as clock output, and buzzer output.

(a) PCL

Clock output pin.

(b) BUZ

Buzzer output pin.

2.2.4 P50 to P57 (Port 5)

These are 8-bit input/output ports. They can be specified bit-wise as input/output ports with port mode register 5 (PM5). When they are used as input ports, on-chip pull-up resistors can be used by defining the pull-up resistor option register L (PUOL). A maximum of 7 out of 8 ports can drive LEDs directly.

2.2.5 P70 to P72 (Port 7)

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

The following operating modes can be specified bit-wise.

(1) Port mode

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

(2) Control mode

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

(a) SI2, SO2

Serial interface serial data input/output pins

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

Serial interface serial clock input/output pin.

(c) RxD, TxD

Asynchronous serial interface serial data input/output pins.

(d) ASCK

Asynchronous serial interface serial clock input 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, see the operation mode setting list in Table 11-2 “Serial Interface Channel 2 Operating Mode Settings”

2.2.6 P100 to P101 (Port 10)

These are 2-bit input/output ports. Besides serving as input/output ports, they function as timer input/output.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 2-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 10 (PM10). When they are used as input ports, on-chip pull-up resistors can be used by defining the pull-up resistor option register H (PUOH).

(2) Control mode

These ports function as timer input/output.

(a) TI5, TI6

Pin for external clock input to the 8-bit timer/event counter 5 and 6.

(b) TO5, TO6

Timer output pins.

2.2.7 AV_{REF}

A/D converter reference voltage input pin.

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

2.2.8 AV_{DD}

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

2.2.9 AV_{SS}

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

2.2.10 $\overline{\text{RESET}}$

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

2.2.11 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.

2.2.12 V_{DD}

Positive power supply pin

2.2.13 V_{SS}

Ground potential pin

2.2.14 V_{PP} (μPD78P083 only)

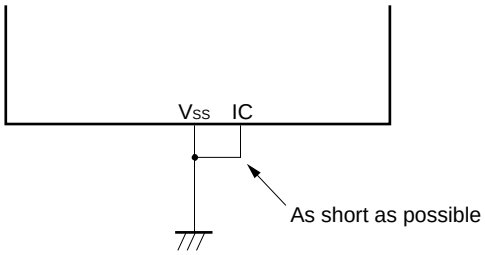
High-voltage apply pin for PROM programming mode setting and program write/verify. Connect directly to V_{SS} in normal operating mode.

2.2.15 IC (Mask ROM version only)

The IC (Internally Connected) pin is provided to set the test mode to check the μPD78083 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.



2.2.16 NC (44-pin plastic QFP versions only)

Not internally connected. Please connect to Vss (open is also possible)

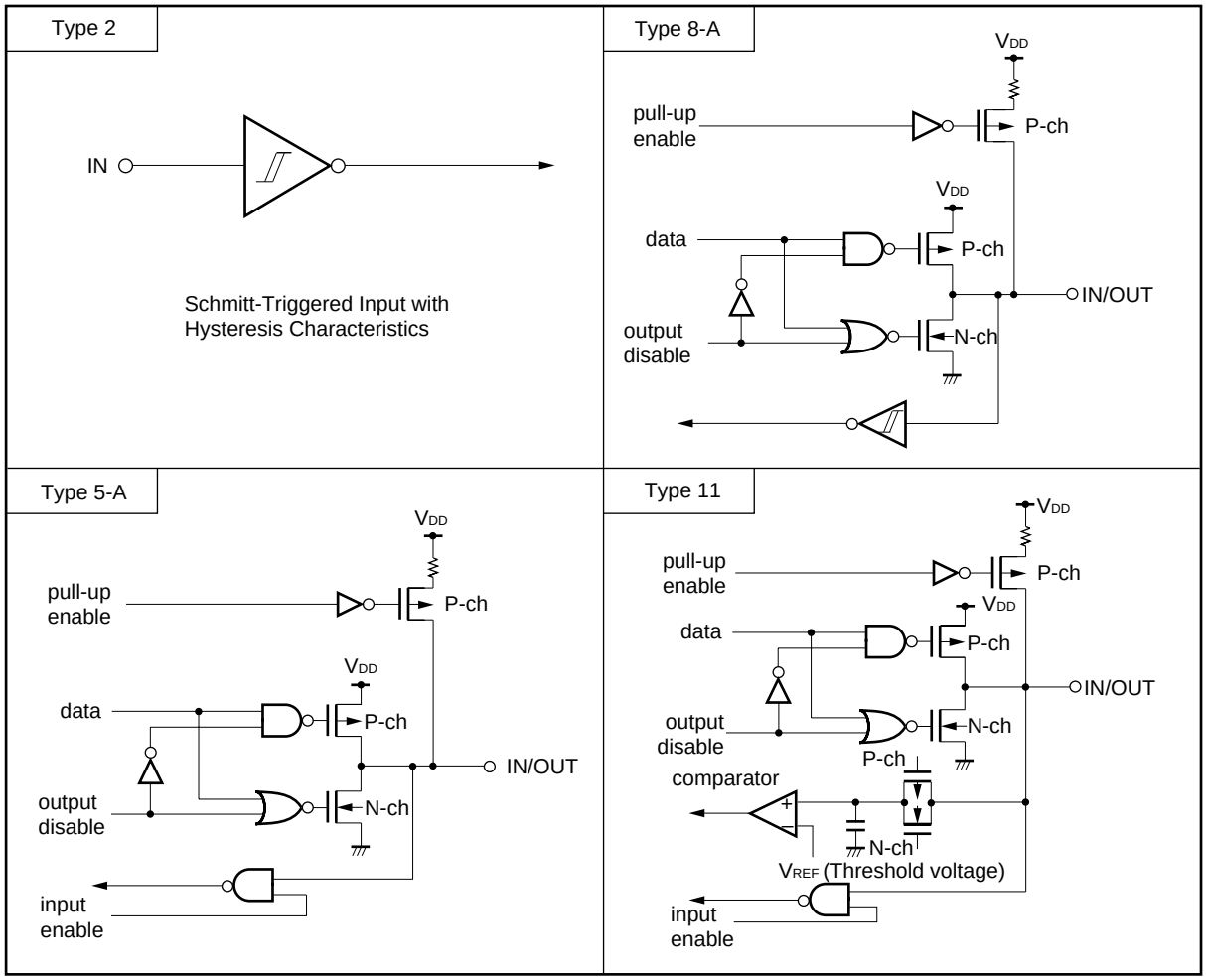
2.3 Pin Input/Output Circuits and Recommended Connection of Unused Pins

Types of input/output circuits of the pins and recommeded connection of unused pins are shown in Table 2-1. For the configuration of each type of input/output circuit, see Figure 2-1.

Table 2-1. Type of Input/Output Circuit of Each Pin

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection for Unused Pins
P00	2	Input	Connect to V_{SS}.
P01/INTP1	8-A	Input/Output	Independently connect to V_{SS} via a resistor.
P02/INTP2			
P03/INTP3			
P10/ANI0-P17/ANI7	11	Input/Output	Independently connect to V_{DD} or V_{SS} via a resistor.
P30-P32	5-A		
P33, P34	8-A		
P35/PCL	5-A		
P36/BUZ			
P37			
P50-P57	5-A		
P70/SI2/RxD	8-A		
P71/SO2/TxD	5-A		
P72/$\overline{\text{SCK2}}$/ASCK	8-A		
P100/TI5/TO5	8-A		
P101/TI6/TO6			
$\overline{\text{RESET}}$	2	Input	–
AV_{REF}	–	–	Connect to V_{SS}.
AV_{DD}			Connect to V_{DD}.
AV_{SS}			Connect to V_{SS}.
V_{PP} (μPD78P083)			Connect directly to V_{SS}.
NC (44-pin plastic QFP version)			Connect to V_{SS} (can also leave open)
IC (Mask ROM version)			Connect directly to V_{SS}.

Figure 2-1. Pin Input/Output Circuit of List



[MEMO]

CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Spaces

Figures 3-1 to 3-3 shows memory maps.

Figure 3-1. Memory Map (μPD78081)

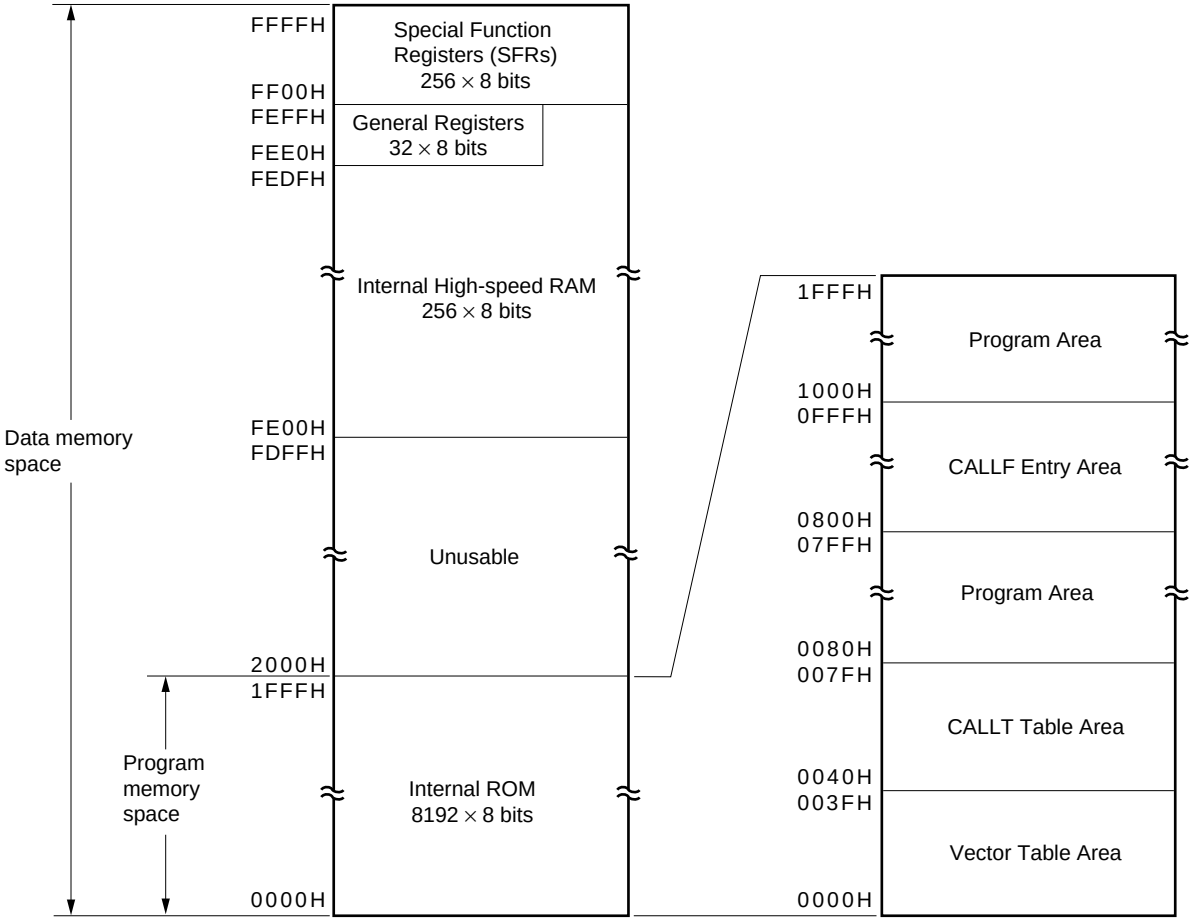


Figure 3-2. Memory Map (μPD78082)

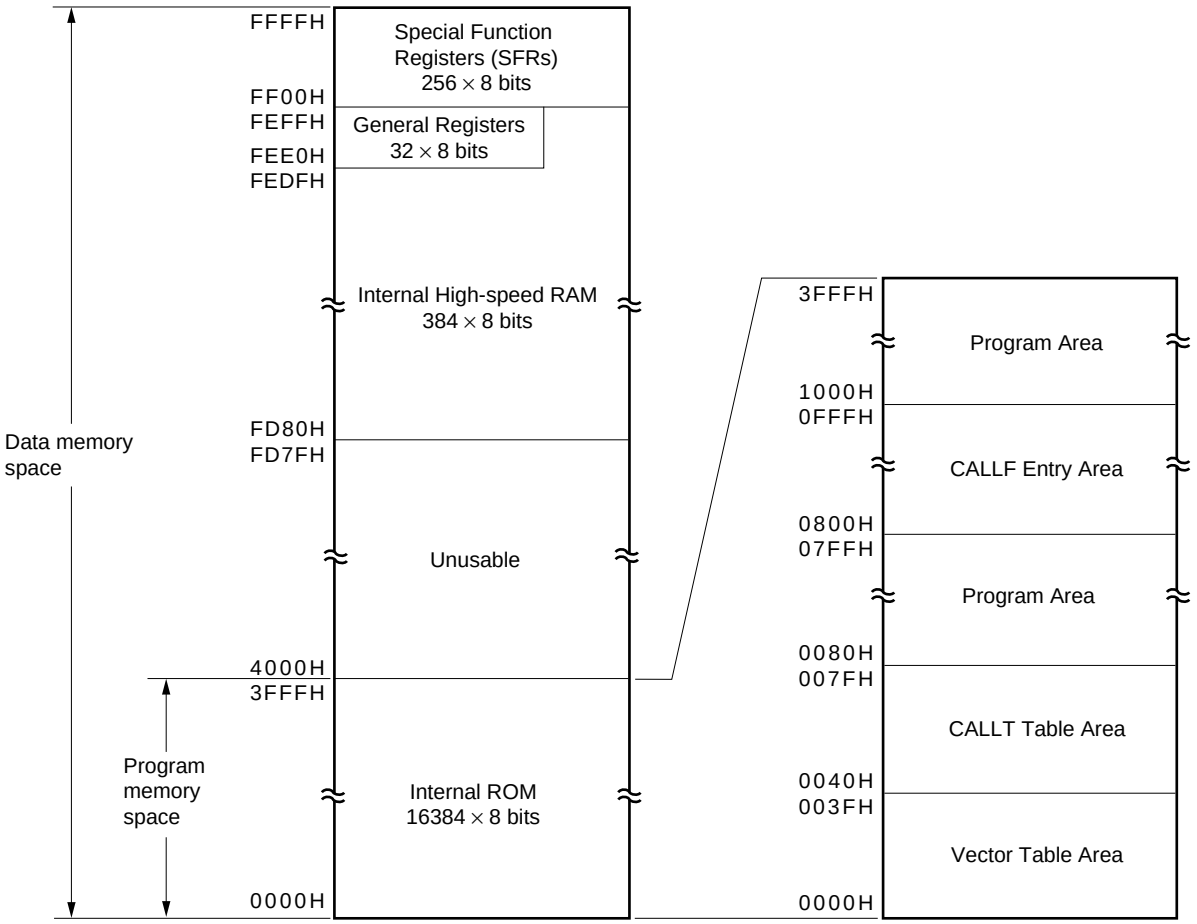
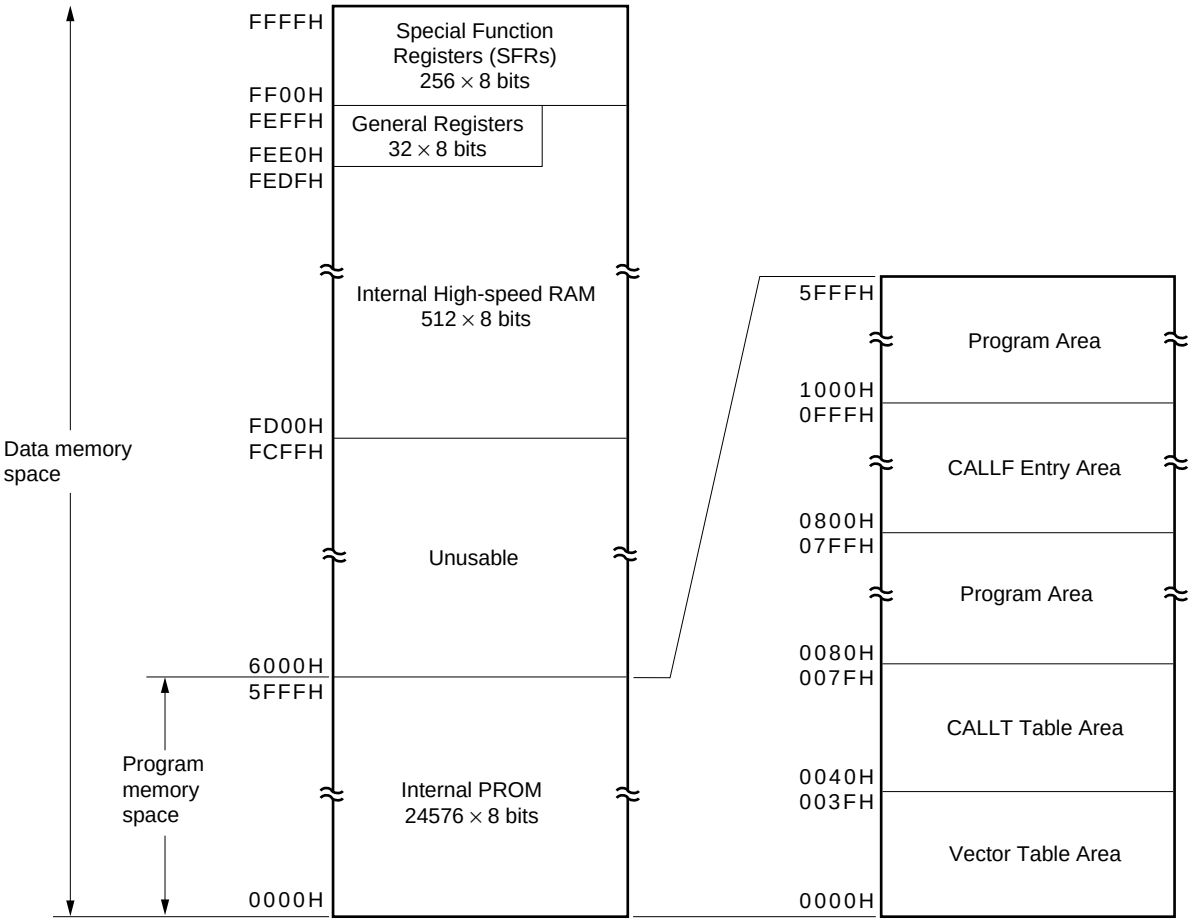


Figure 3-3. Memory Map (μPD78P083)



3.1.1 Internal program memory space

The internal program memory is mask ROM with a 8192 × 8-bit configuration in the μPD78081, and a 16384 × 8-bit configuration in the μPD78082, and PROM with a 24576 × 8-bit configuration in the μPD78P083.

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

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 3-1. Vector Table

Vector Table Address	Interrupt Request
0000H	$\overline{\text{RESET}}$ input
0004H	INTWDT
0008H	INTP1
000AH	INTP2
000CH	INTP3
0018H	INTSER
001AH	INTSR/INTCSI2
001CH	INTST
0028H	INTAD
002AH	INTTM5
002CH	INTTM6
003EH	BRK

(2) CALLT instruction table area

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

(3) CALLF instruction entry area

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

3.1.2 Internal data memory space

The internal high speed RAM configuration is 256 × 8-bit in the μPD78081, 384 × 8-bit in the μPD78082 and 512 × 8-bit in the μPD8P083. 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.

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 **Table 3-2. Special-Function Register List** in **3.2.3 Special Function Register (SFR)**).

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

3.1.4 Data memory addressing

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

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

To address the memory that is manipulated when an instruction is executed, the μPD78083 Subseries is provided with many addressing modes with a high operability. Especially at addresses corresponding to data memory area, particular addressing modes are possible to meet the functions of the special function registers (SFRs) and general registers. This area is between FE00H and FFFFH for the μPD78081, FD80H and FFFFH for the μPD78082, and between FD00H and FFFFH for the μPD78P083. The data memory space is the entire 64K-byte space (0000H to FFFFH). Figure 3-4 to 3-6 show the data memory addressing modes. For details of each addressing, refer to **3.4 Operand Address Addressing.**

Figure 3-4. Data Memory Addressing (μPD78081)

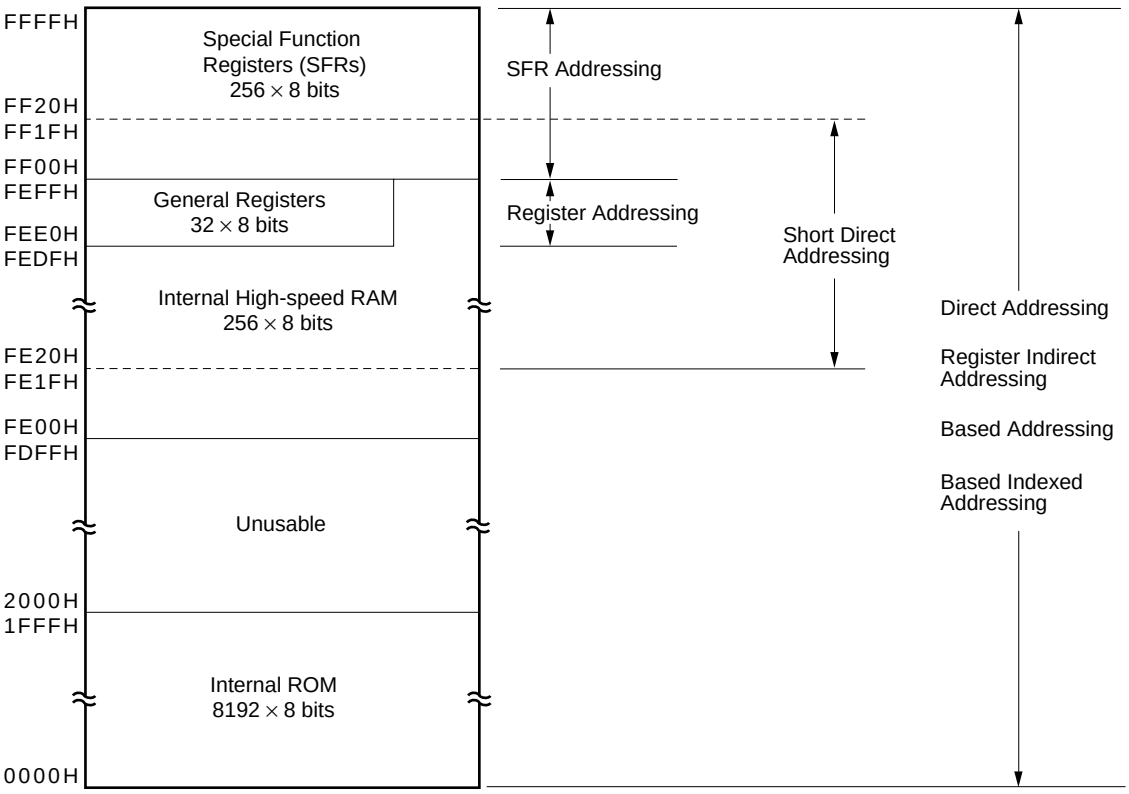


Figure 3-5. Data Memory Addressing (μPD78082)

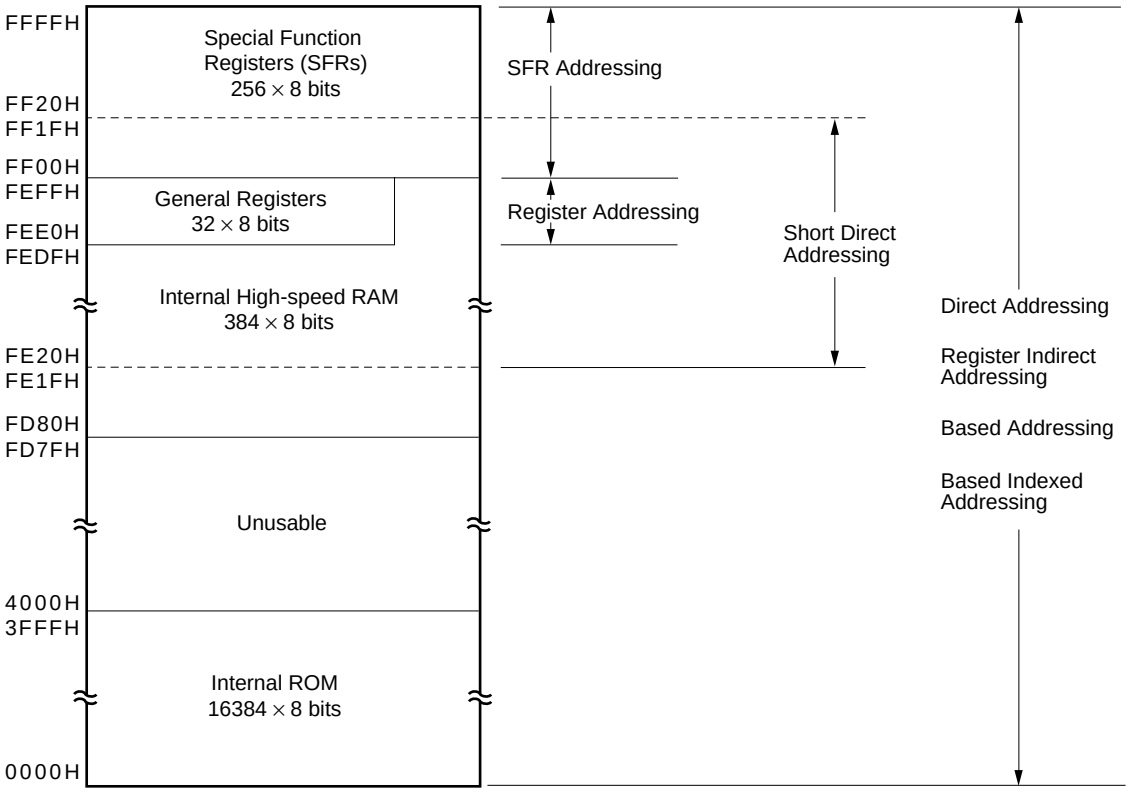
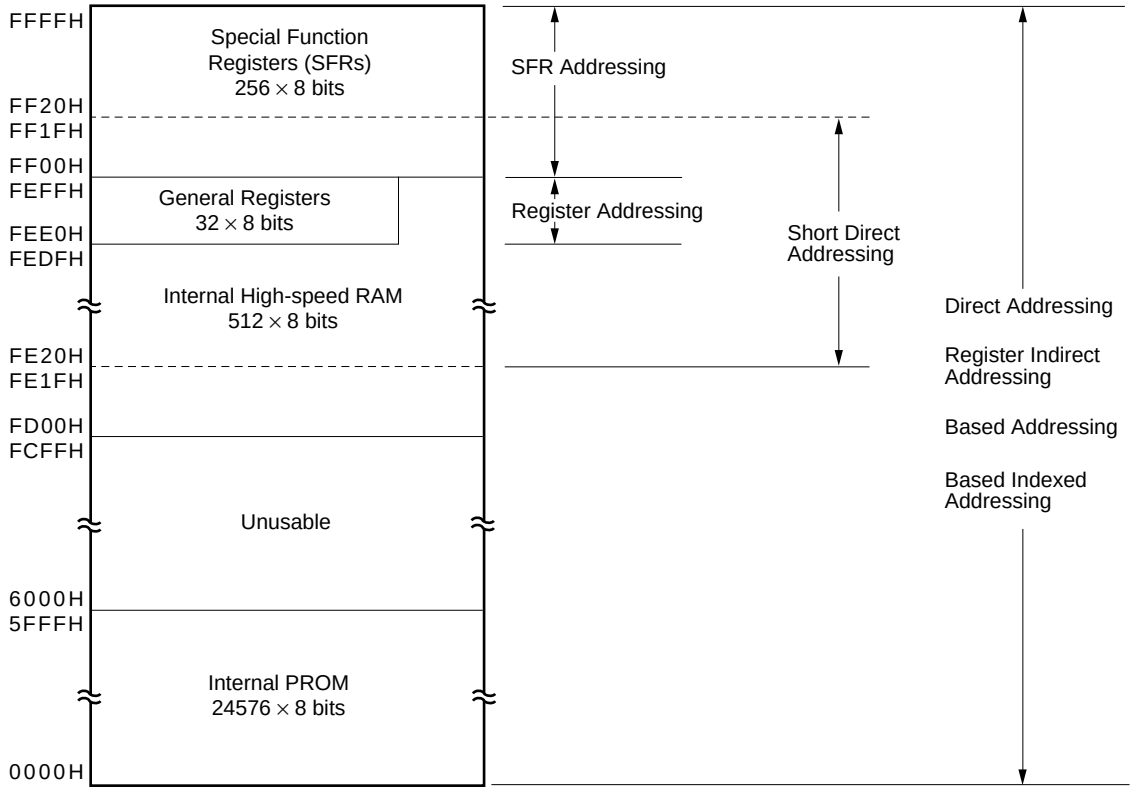


Figure 3-6. Data Memory Addressing (μPD78P083)



3.2 Processor Registers

The μPD78083 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, a program status word and a stack pointer.

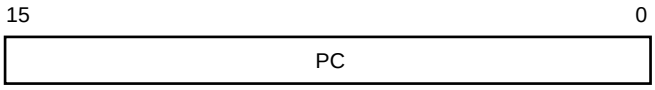
(1) Program counter (PC)

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 3-7. Program Counter Configuration

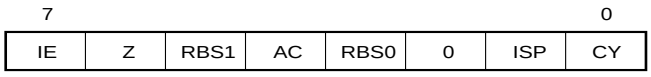


(2) Program status word (PSW)

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

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

Figure 3-8. Program Status Word Configuration



(a) Interrupt enable flag (IE)

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

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

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

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

(b) Zero flag (Z)

When the operation result is zero, this flag is set (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 request whose priority is specified by the priority specify flag registers (PR0L, PR0H, and PR1L) (Refer to **12.3 (3) Priority specify flag registers (PR0L, PR0H, and PR1L)**) to be low is disabled. Whether the interrupt request is actually acknowledged is controlled by the status of the interrupt enable flag (IE).

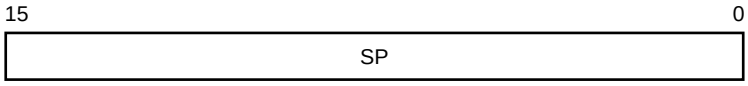
(f) Carry flag (CY)

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

(3) Stack pointer (SP)

This is a 16-bit register to hold the start address of the memory stack area. Only the internal high-speed RAM area (FE00H-FEFFH for the μPD78081, FD80H-FEFFH for the μPD78082, and FD00H-FEFFH for the μPD78P083) can be set as the stack area.

Figure 3-9. Stack Pointer Configuration



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

Each stack operation saves/resets data as shown in Figures 3-10 and 3-11.

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

Figure 3-10. Data to be Saved to Stack Memory

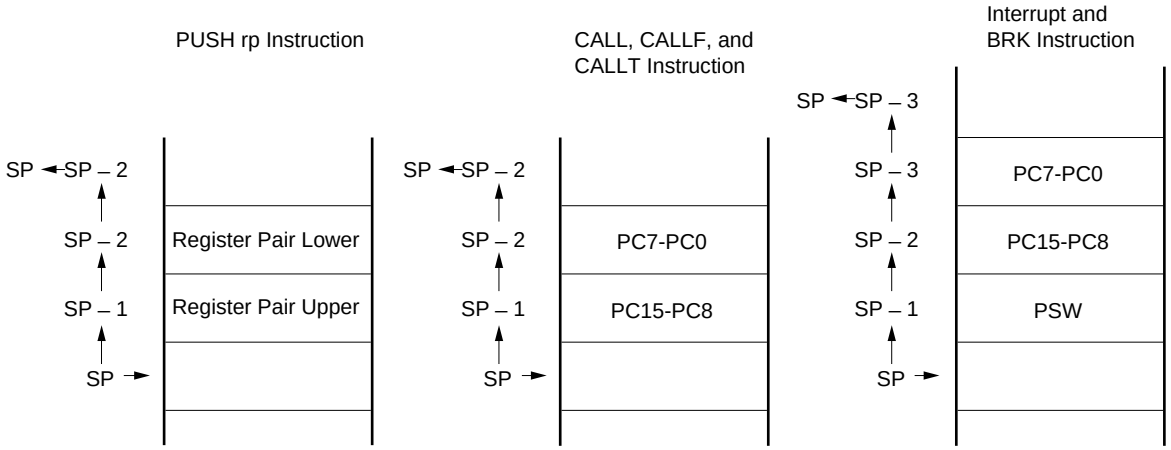
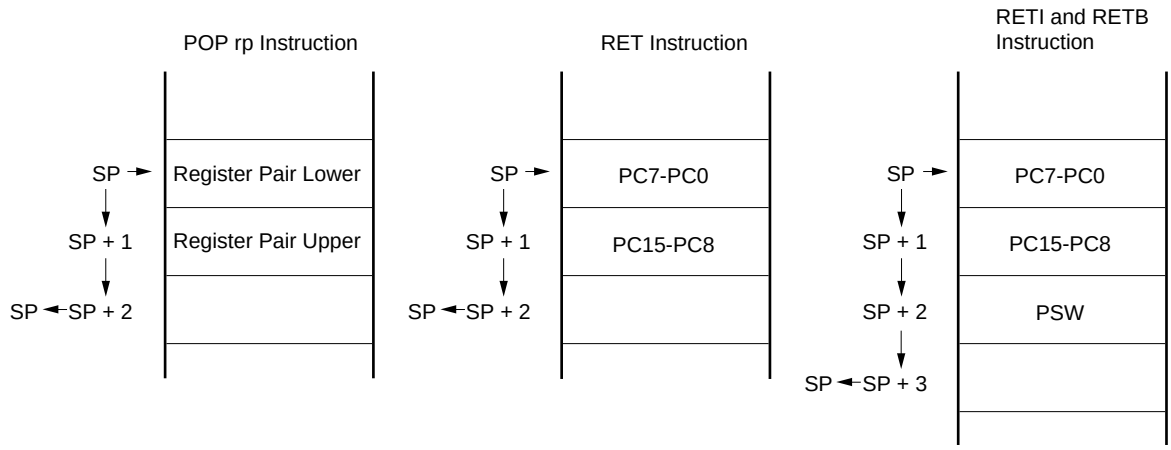


Figure 3-11. Data to be Reset from Stack Memory



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).

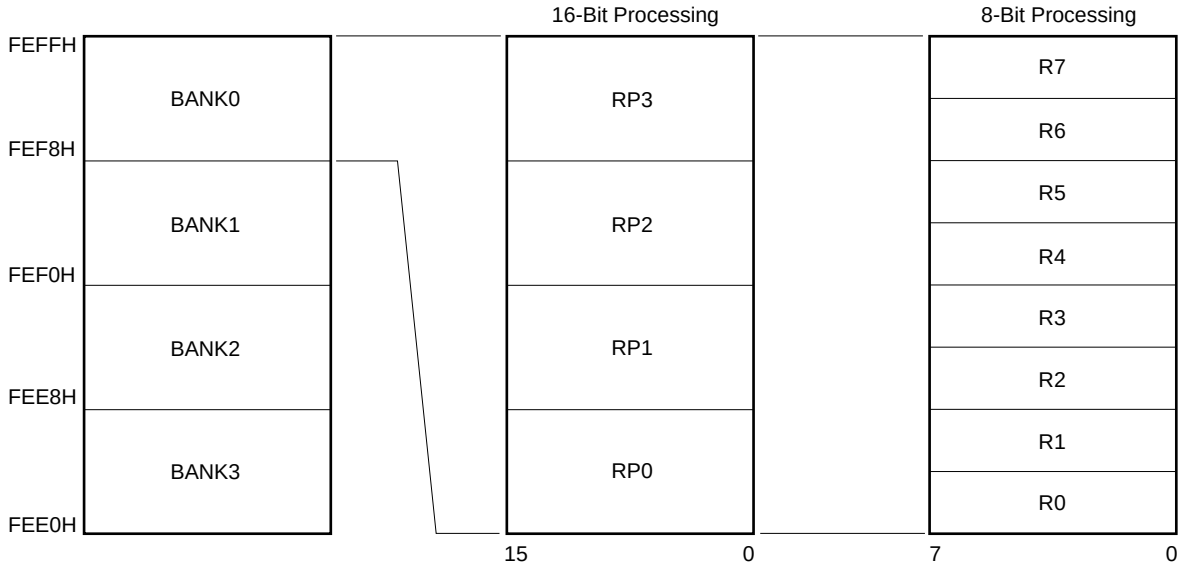
Each register can also be used as an 8-bit register. Two 8-bit registers can be used in pairs as a 16-bit register (AX, BC, DE and HL).

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).

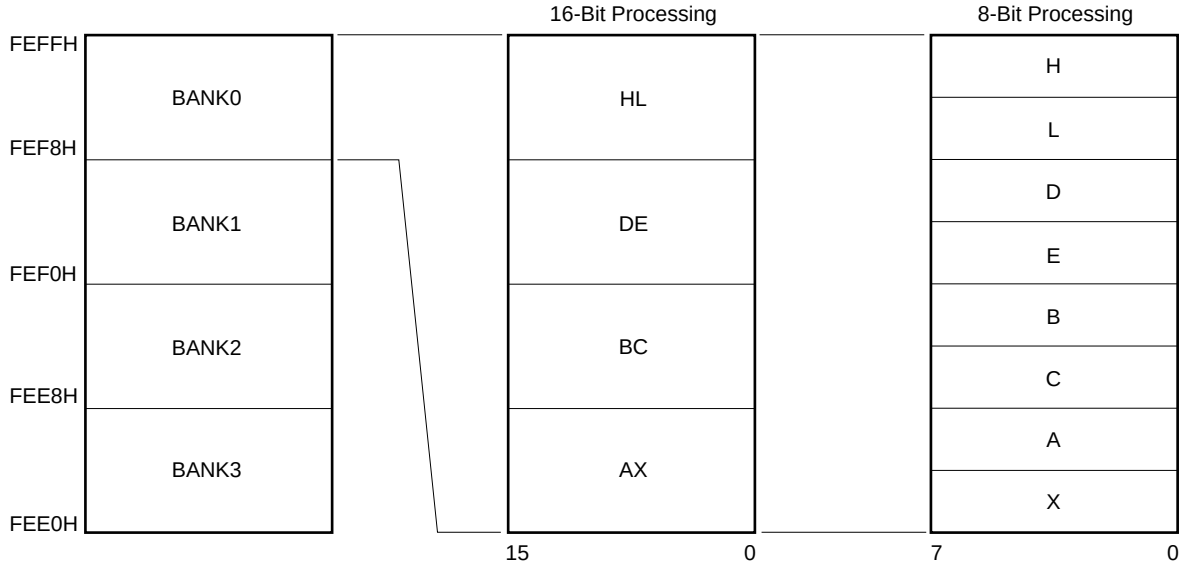
Register banks to be used for instruction execution are set with the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interruption request for each bank.

Figure 3-12. 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 3-2 gives a list of special-function registers. The meaning of items in the table is as follows.

- Symbol

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

Table 3-2. 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	Port0	P0	R/W	√	√	—	00H
FF01H	Port1	P1		√	√	—	
FF03H	Port3	P3		√	√	—	
FF05H	Port5	P5		√	√	—	
FF07H	Port7	P7		√	√	—	
FF0AH	Port10	P10		√	√	—	
FF1FH	A/D conversion result register	ADCR	R	√	√	—	Undefined
FF20H	Port mode register 0	PM0	R/W	√	√	—	FFH
FF21H	Port mode register 1	PM1		√	√	—	
FF23H	Port mode register 3	PM3		√	√	—	
FF25H	Port mode register 5	PM5		√	√	—	
FF27H	Port mode register 7	PM7		√	√	—	
FF2AH	Port mode register 10	PM10		√	√	—	
FF40H	Timer clock select register 0	TCL0		√	√	—	00H
FF42H	Timer clock select register 2	TCL2		—	√	—	
FF50H	Compare Register 50	CR50		—	√	—	
FF51H	8-bit timer register 5	TM5	R	—	√	—	
FF52H	Timer clock select register 5	TCL5	R/W	—	√	—	
FF53H	8-bit timer mode control register 5	TMC5		√	√	—	
FF54H	Compare Register 60	CR60		—	√	—	
FF55H	8-bit timer register 6	TM6	R	—	√	—	
FF56H	Timer clock select register 6	TCL6	R/W	—	√	—	
FF57H	8-bit timer mode control register 6	TMC6		√	√	—	
FF70H	Asynchronous serial interface mode register	ASIM		√	√	—	
FF71H	Asynchronous serial interface status register	ASIS	R	√	√	—	
FF72H	Serial operating mode register 2	CSIM2	R/W	√	√	—	
FF73H	Baud rate generator control register	BRGC		—	√	—	
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
FFE0H	Interrupt request flag register 0L	IF0, IF0L		√	√	√	
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		√	√		

Table 3-2. 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	
FFEAH	Priority order specify flag register 1L	PR1L	R/W	√	√	—	FFH
FFECH	External interrupt mode register 0	INTM0		—	√	—	00H
FFEDH	External interrupt mode register 1	INTM1		—	√	—	
★ FFF0H	Memory size switching register	IMS		—	√	—	(Note)
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		√	√	—	

Note The value after reset depends on products.
μPD78081 : 82H, μPD78082 : 64H, μPD78P083 : 46H.

3.3 Instruction Address Addressing

An instruction address is determined by program counter (PC) contents. The contents of PC are normally incremented (+1 for each byte) automatically according to the number of bytes of an instruction to be fetched each time another instruction is executed. When a branch instruction is executed, the branch destination information is set to the PC and branched by the following addressing. (For details of instructions, refer to **78K/0 USER'S MANUAL: Instruction (IEU-1372)**.

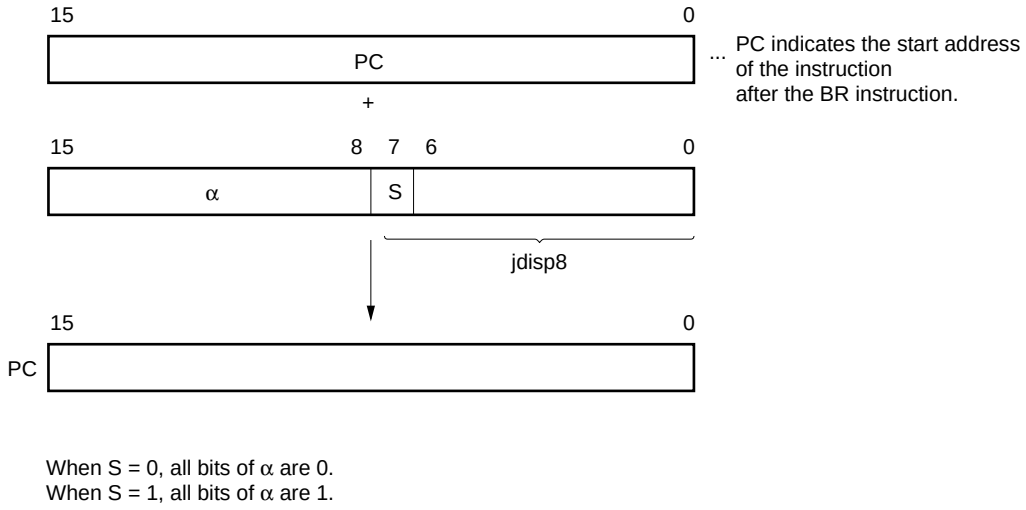
3.3.1 Relative addressing

[Function]

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

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

[Illustration]



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

3.3.2 Immediate addressing

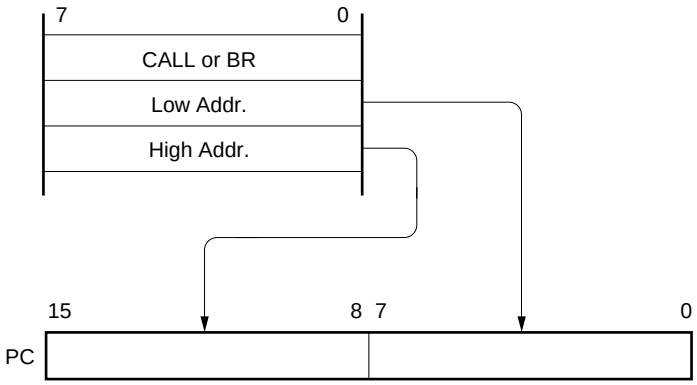
[Function]

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

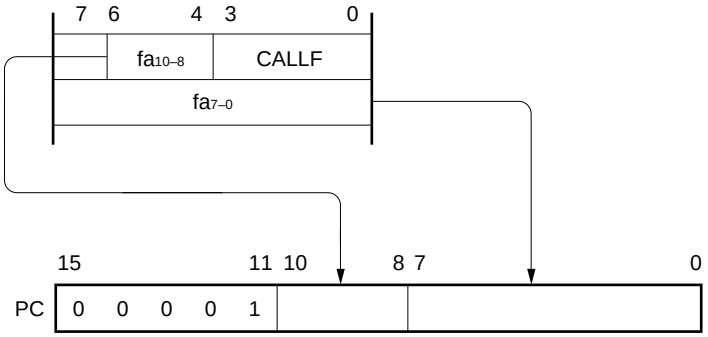
This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed. The CALL !addr16 and BR !addr16 instruction can branch in the entire memory space. The CALLF !addr11 instruction branches to an area of addresses 0800H through 0FFFH.

[Illustration]

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



In the case of CALLF !addr11 instruction



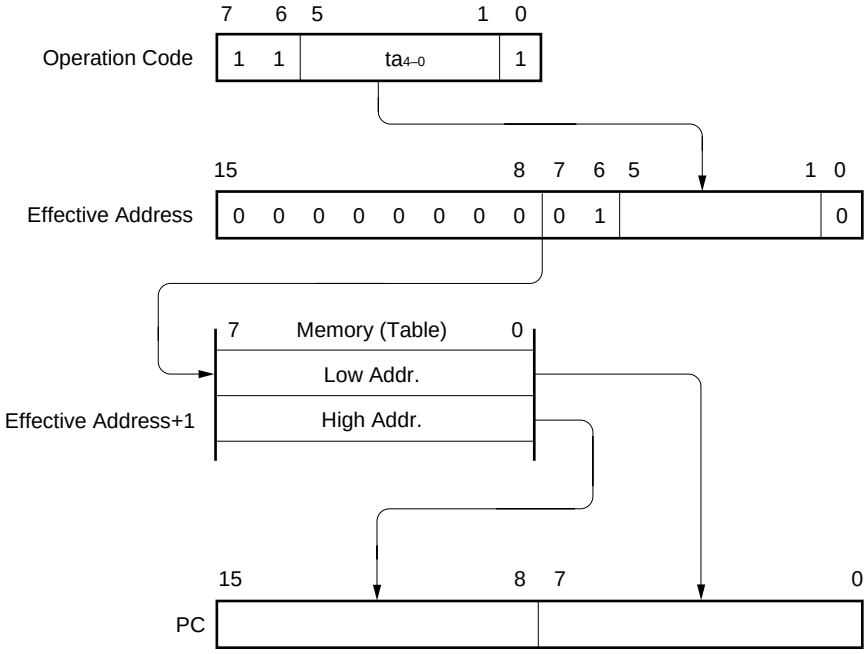
3.3.3 Table indirect addressing

[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.

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

[Illustration]



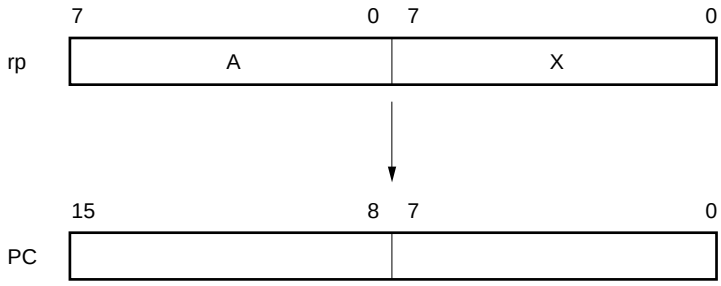
3.3.4 Register addressing

[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 (illicitly) addressed.

Of the μPD78083 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 accesses a general register as an operand. The general register accessed is specified by the register bank select flags (RBS0 and RBS1) and register specify code (Rn or RPn) in an instruction code. Register addressing is carried out when an instruction with the following operand format is executed. When an 8-bit register is specified, one of the eight registers is specified with 3 bits in the operation code.

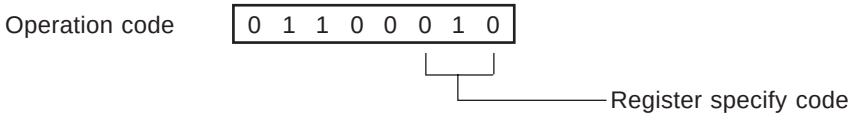
[Operand format]

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

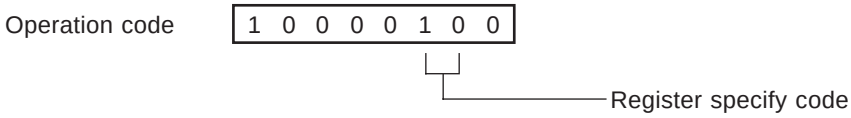
'r' and 'rp' can be described with function names (X, A, C, B, E, D, L, H, AX, BC, DE and HL) as well as absolute names (R0 to R7 and RP0 to RP3).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



3.4.3 Direct addressing

[Function]

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

[Operand format]

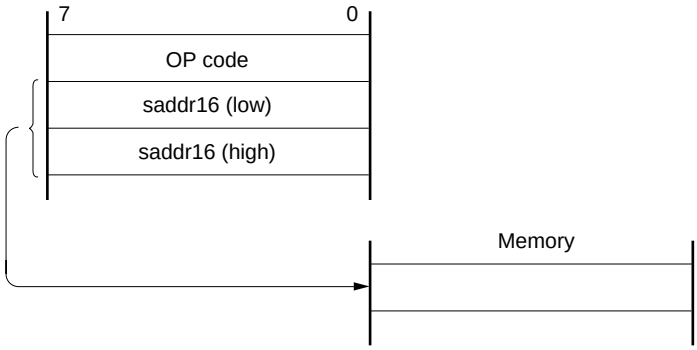
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

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

[Illustration]



3.4.4 Short direct addressing

[Function]

The memory to be manipulated in the fixed space is directly addressed with 8-bit data in an instruction word. The fixed space to which this address is applied is a 256-byte space of addresses FE20H through 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 through FF1FH) to which short direct addressing is applied is a part of the entire SFR area. To this area, ports frequently accessed by the program, and the compare registers and capture registers of timer/event counters are mapped. These SFRs can be manipulated with a short byte length and a few clocks.

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

[Operand format]

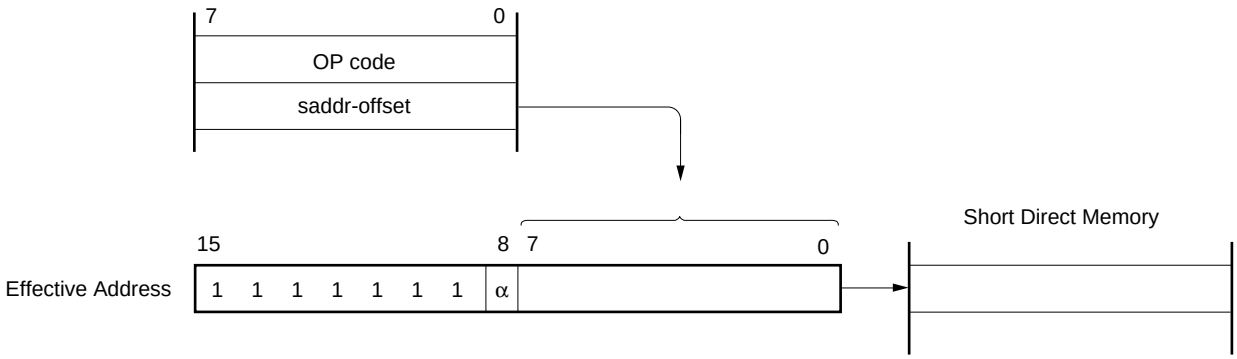
Identifier	Description
saddr	Label of FE20H to FF1FH immediate data
saddrp	Label of FE20H to FF1FH immediate data (even address only)

[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, $\alpha = 0$

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

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]

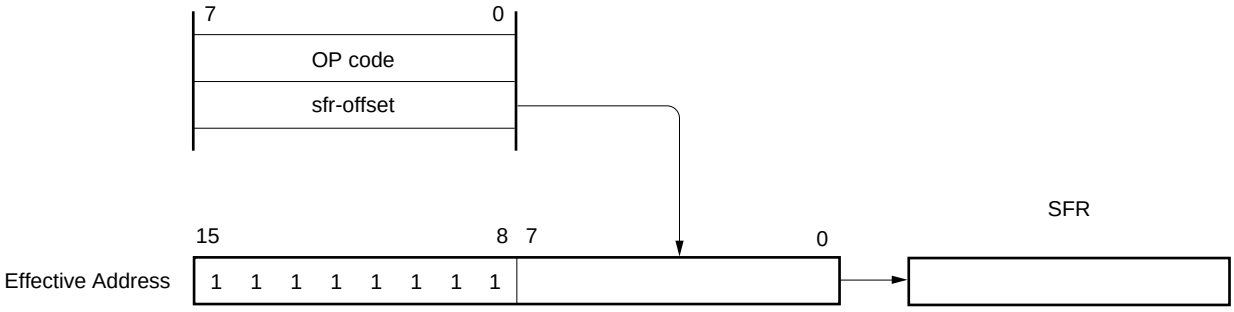
Identifier	Description
sfr	Special-function register name
sfrp	16-bit manipulatable special-function register name (even address only)

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

[Operand format]

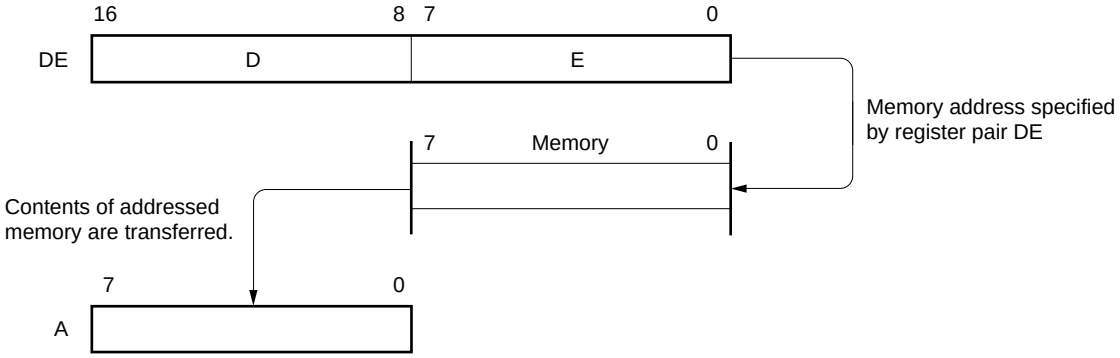
Identifier	Description
—	[DE], [HL]

[Description example]

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

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

[Illustration]



3.4.7 Based addressing

[Function]

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

[Operand format]

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

[Description example]

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

Operation code

1	0	1	0	1	0	1	1

3.4.9 Stack addressing

[Function]

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

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

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

[Description example]

In the case of PUSH DE

Operation code

1	0	1	1	0	1	0	1

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The μPD78083 Subseries units incorporate an input port and thirty-two 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

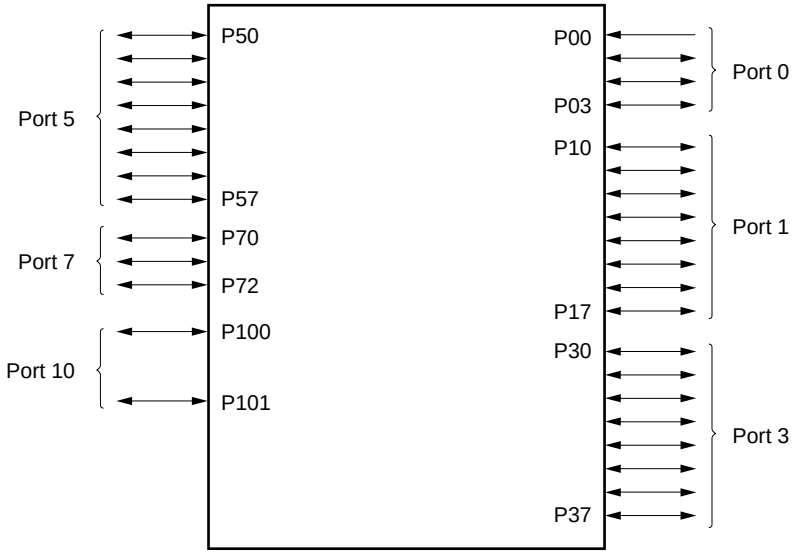


Table 4-1. Port Functions

Pin Name	Input/Output	Function		Dual-Function Pin
P00	Input	Port 0	Input only	—
P01	Input/output	4-bit input/output port	Input/output is specifiable bit-wise. When used as the input port, it is possible to connect a pull-up resistor by software.	INTP1
P02				INTP2
P03				INTP3
P10-P17	Input/output	Port 1 8-bit input/output port Input/output is specifiable bit-wise. When used as the input port, it is possible to connect a pull-up resistor by software.		ANI0-ANI7
P30-P34	Input/output	Port 3		—
P35		8-bit input/output port		PCL
P36		Input/output is specifiable bit-wise.		BUZ
P37		When used as the input port, it is possible to connect a pull-up resistor by software.		—
P50-P57	Input/output	Port 5 8-bit input/output port A maximum of 7 out of 8 ports can drive LEDs directly. Input/output is specifiable bit-wise. When used as the input port, it is possible to connect a pull-up resistor by software.		—
P70	Input/output	Port 7		SI2/RxD
P71		3-bit input/output port		SO2/TxD
P72		Input/output is specifiable bit-wise. When used as the input port, it is possible to connect a pull-up resistor by software.		$\overline{\text{SCK2}}$/ASCK
P100	Input/output	Port 10		TI5/TO5
P101		2-bit input/output port Input/output is specifiable bit-wise. When used as the input port, it is possible to connect a pull-up resistor by software.		TI6/TO6

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, 1, 3, 5, 7, 10) Pull-up resistor option register (PUOH, PUOL)
Port	Total: 33 ports (1 input, 32 inputs/outputs)
Pull-up resistor	Total: 32 (software specifiable)

4.2.1 Port 0

Port 0 is an 4-bit input/output port with output latch. P01 to P03 pins can specify the input mode/output mode in 1-bit units with the port mode register 0 (PM0). P00 pin is input-only port. When P01 to P03 pins are used as input ports, an on-chip pull-up resistor can be used to them in 3-bit units with a pull-up resistor option register L (PUOL).

Dual-functions include external interrupt request input.

$\overline{\text{RESET}}$ 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. P00 Block Diagram

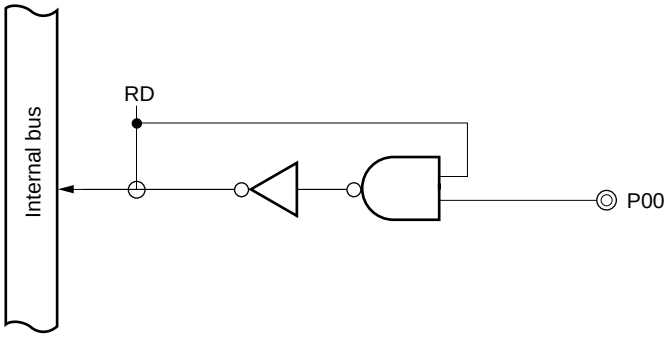
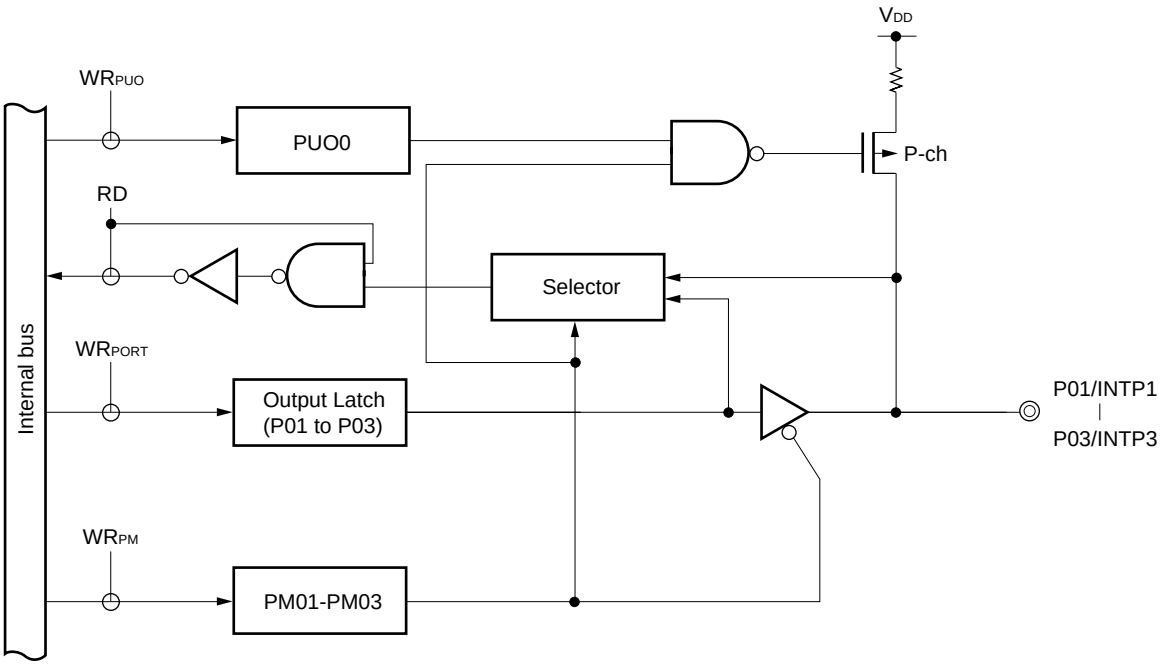


Figure 4-3. P01 to P03 Block Diagram



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 on-chip pull-up resistor can be used to them in 8-bit units with a pull-up resistor option register L (PUOL).

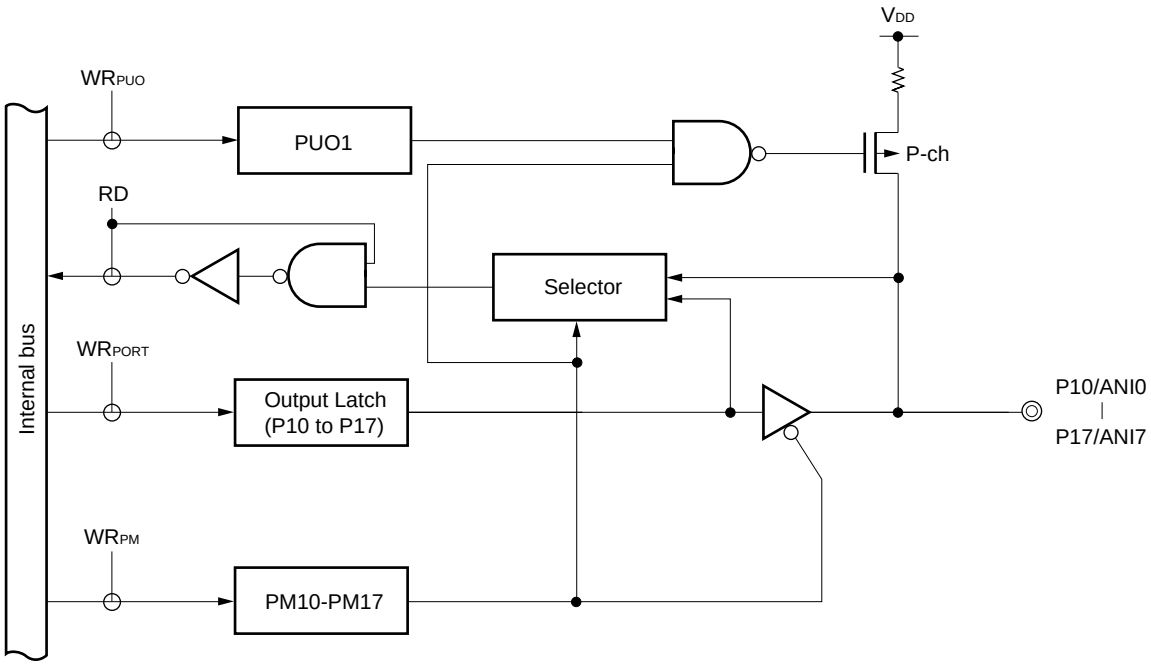
Dual-functions include an A/D converter analog input.

$\overline{\text{RESET}}$ 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. P10 to P17 Block Diagram



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

4.2.3 Port 3

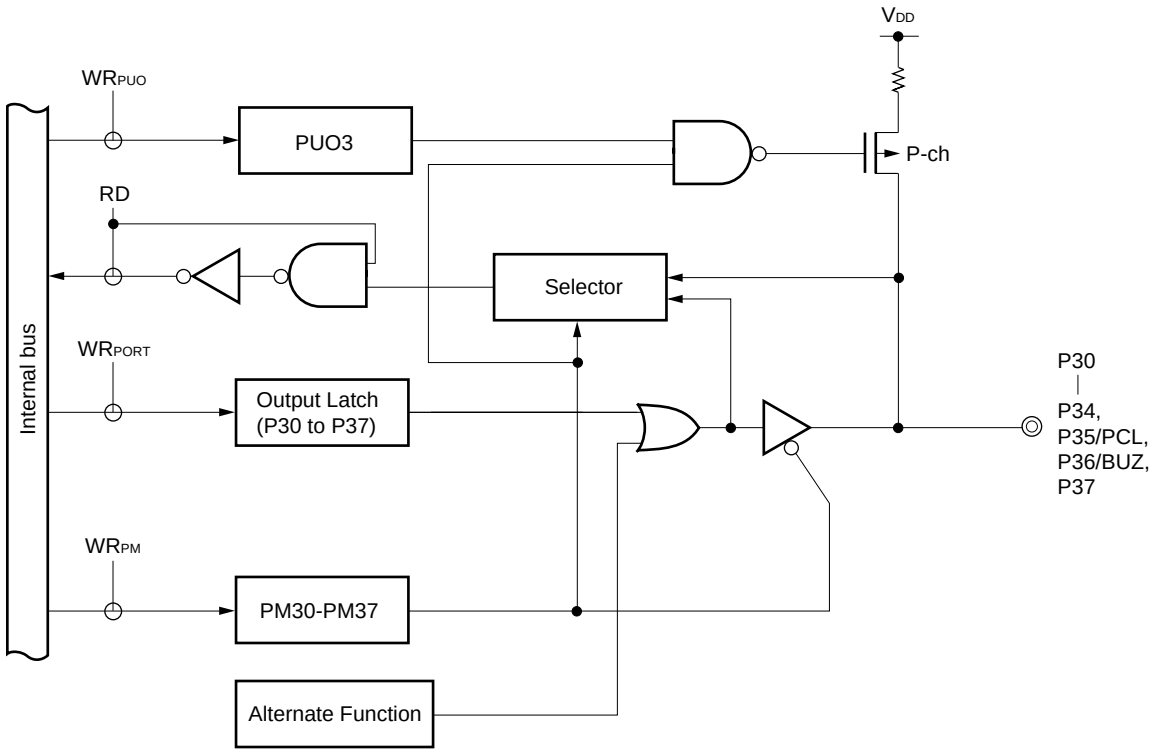
Port 3 is an 8-bit input/output port with output latch. P30 to P37 pins can specify the input mode/output mode in 1-bit units with the port mode register 3 (PM3). When P30 to P37 pins are used as input ports, an on-chip pull-up resistor can be used to them in 8-bit units with a pull-up resistor option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

Dual-functions include clock output and buzzer output.

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

Figure 4-5 shows a block diagram of port 3.

Figure 4-5. P30 to P37 Block Diagram



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

4.2.4 Port 5

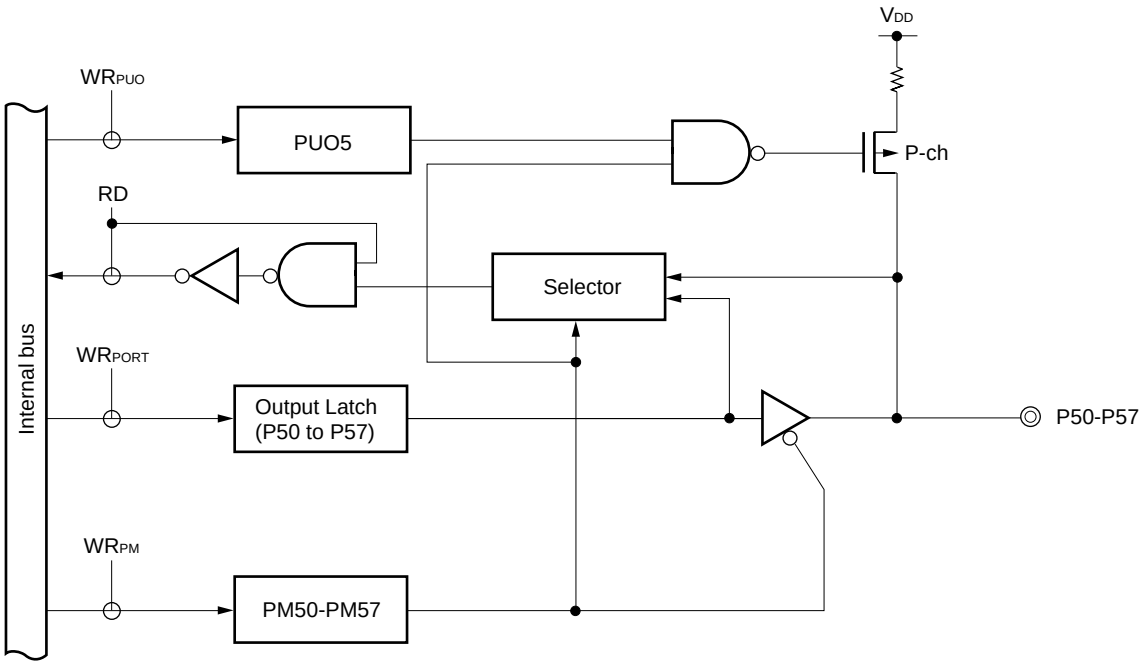
Port 5 is an 8-bit input/output port with output latch. P50 to P57 pins can specify the input mode/output mode in 1-bit units with the port mode register 5 (PM5). When P50 to P57 pins are used as input ports, an on-chip pull-up resistor can be used to them in 8-bit units with a pull-up resistor option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

A maximum of 7 out of 8 ports can drive LEDs directly.

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

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

Figure 4-6. P50 to P57 Block Diagram



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

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

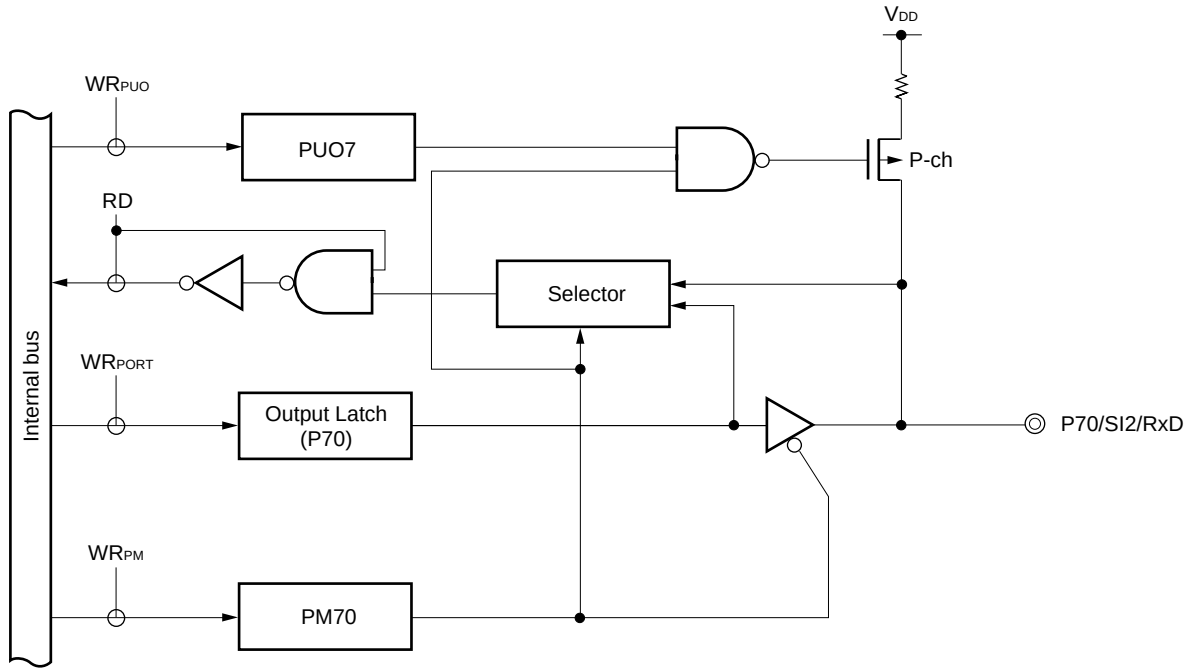
Dual-functions include serial interface channel 2 data input/output and clock input/output.

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

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

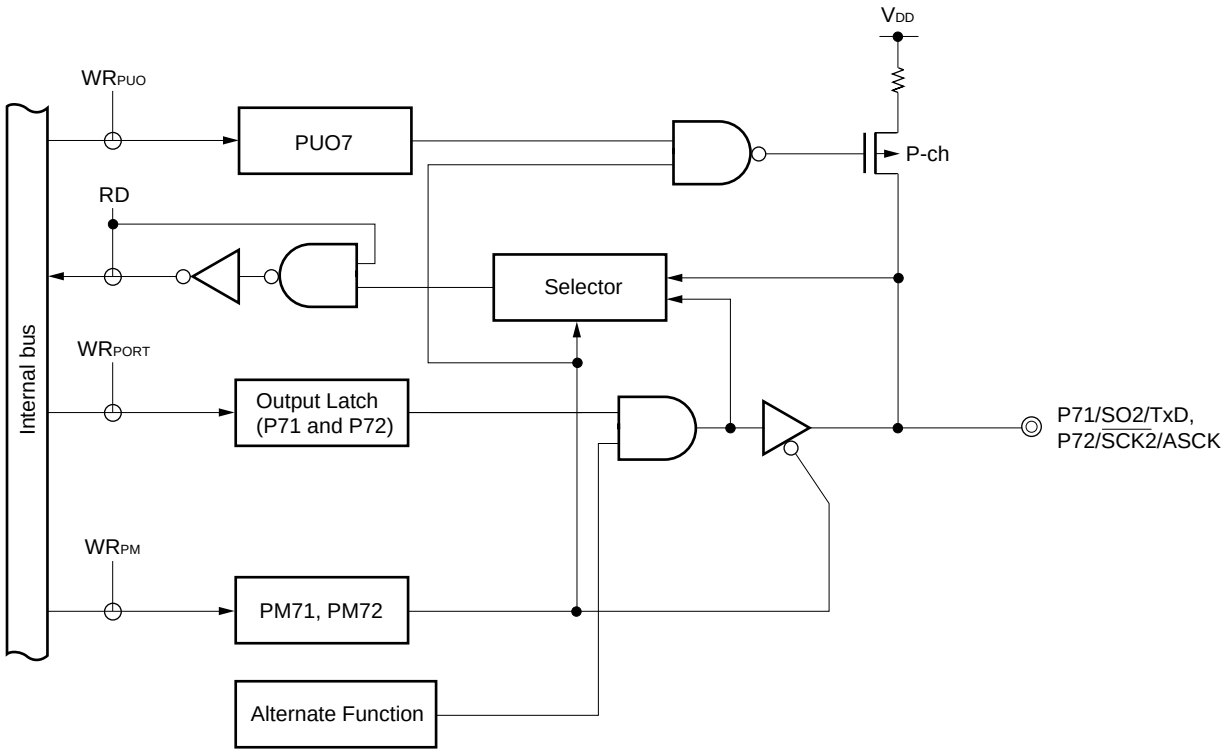
Caution When used as a serial interface, set the input/output and output latch according to its functions. For the setting method, refer to Table 11-2 Serial Interface Channel 2 Operating Mode Settings.

Figure 4-7. P70 Block Diagram



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

Figure 4-8. P71 and P72 Block Diagram



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

4.2.6 Port 10

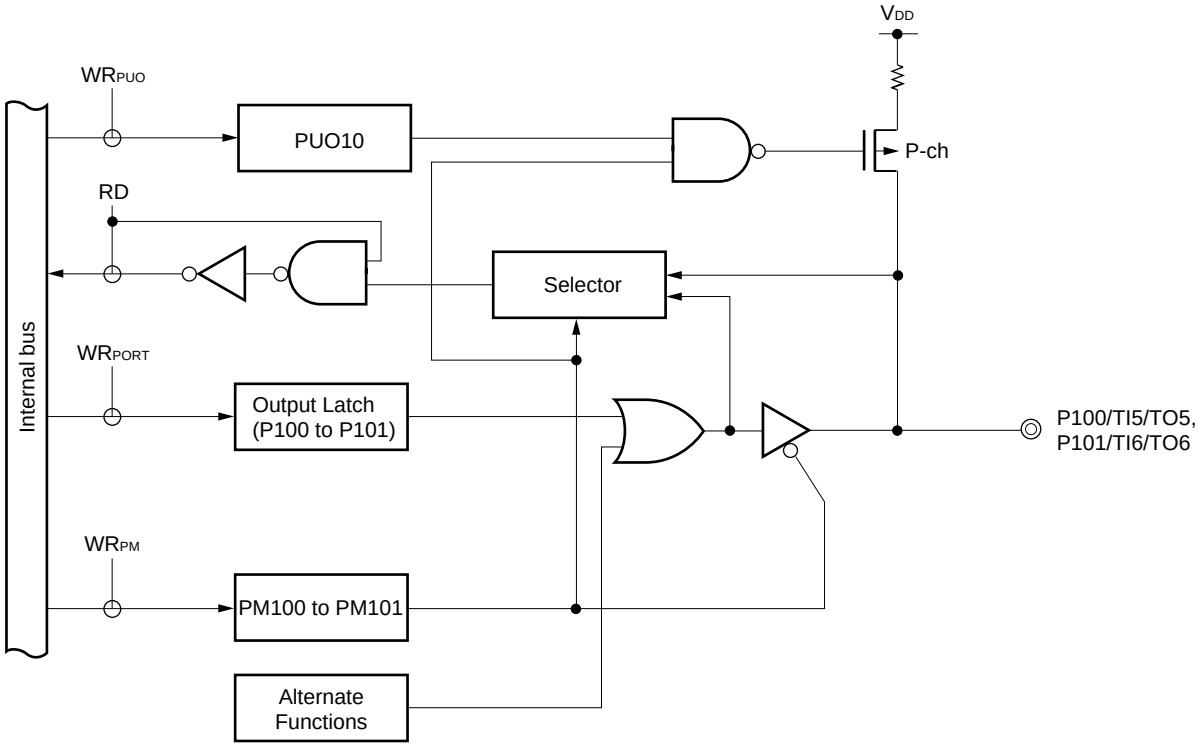
This is an 2-bit input/output port with output latches. Input mode/output mode can be specified bit-wise by means of port mode register 10 (PM10). When pins P100 to P101 are used as input port pins, an on-chip pull-up resistor can be used as an 2-bit unit by means of pull-up resistor option register H (PUOH).

These pins are dual function pins and serve as timer inputs/outputs.

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

The port 10 block diagram is shown in Figure 4-9.

Figure 4-9. P100 to P101 Block Diagram



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

4.3 Port Function Control Registers

The following two types of registers control the ports.

- Port mode registers (PM0, PM1, PM3, PM5, PM7, PM10)
- Pull-up resistor option register (PUOH, PUOL)

(1) Port mode registers (PM0, PM1, PM3, PM5, PM7, PM10)

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

PM0, PM1, PM3, PM5, PM7, PM10 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. P00 pin is input-only pin.

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		
P01 to P03	INTP1 to INTP3	Input	1	×
P10 to P17[Note]	ANI0 to ANI7	Input	1	×
P35	PCL	Output	0	0
P36	BUZ	Output	0	0
P100	TI5	Input	1	×
	TO5	Output	0	0
P101	TI6	Input	1	×
	TO6	Output	0	0

Note If a read instruction is performed to these pins when they are used as an alternate function, read data is to be undefined.

Caution When port 7 is used for serial interface, the I/O latch or output latch must be set according to its function. For the setting methods, see Table 11-2 "Serial Interface Channel 2 Operating Mode Settings."

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

Figure 4-10. Port Mode Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
PM0	1	1	1	1	PM03	PM02	PM01	1	FF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FF25H	FFH	R/W
PM7	1	1	1	1	1	PM72	PM71	PM70	FF27H	FFH	R/W
PM10	1	1	1	1	1	1	PM101	PM100	FF2AH	FFH	R/W

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

Caution Set 1 to the bits 0, 4 to 7 of PM0, bits 3 to 7 of PM7 and bits 2 to 7 of PM10.

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

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

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

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

Cautions 1. P00 pin does not incorporate a pull-up resistor.

2. When port 1 is used as dual-function pin, an on-chip pull-up resistor cannot be used even if 1 is set in PUOL bit 1 (PUO1).

Figure 4-11. 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	0	PUO10	0	0	FFF3H	00H	R/W
	⑦	6	⑤	4	③	2	①	⓪			
PUOL	PUO7	0	PUO5	0	PUO3	0	PUO1	PUO0	FFF7H	00H	R/W

PUOm	Pm Internal Pull-up Resistor Selection (m = 0, 1, 3, 5, 7, 10)
0	Internal pull-up resistor not used
1	Internal pull-up resistor used

Caution Set 0 to the bits 0, 1, 3 to 7 of PUOH and bits 2, 4, 6 of PUOL.

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 type of system clock oscillator is available.

Main system clock oscillator

This circuit oscillates at frequencies of 1 to 5.0 MHz. Oscillation can be stopped by executing the STOP instruction.

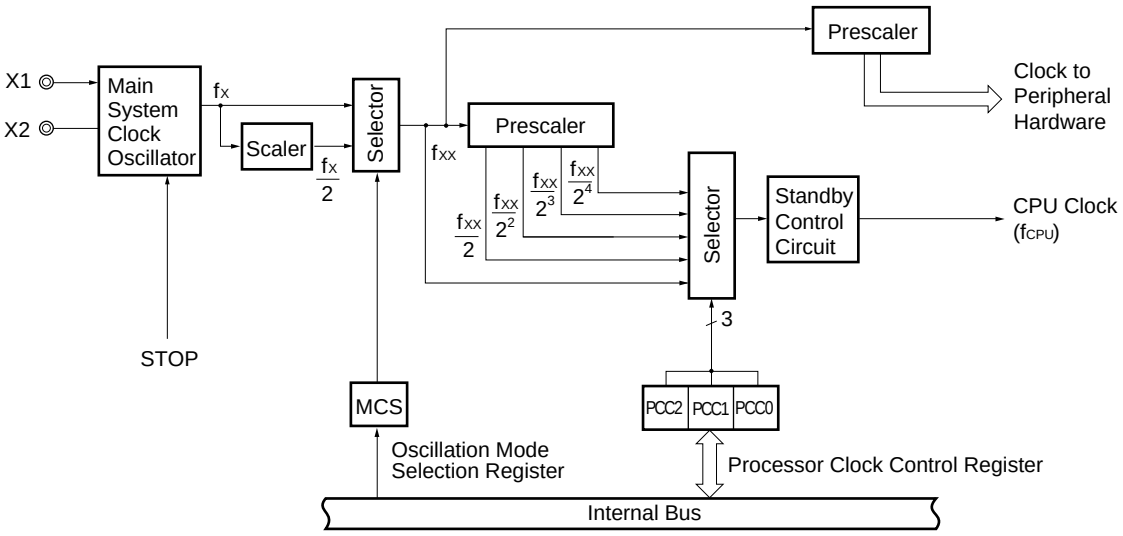
5.2 Clock Generator Configuration

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

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 and the ratio of division.

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. Processor Clock Control Register Format

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

PCC2	PCC1	PCC0	CPU Clock Selection (f_{CPU})		
				MCS=1	MCS=0
0	0	0	f_{XX}	f_X (0.4 μs)	$f_X/2$ (0.8 μs)
0	0	1	$f_{XX}/2$	$f_X/2$ (0.8 μs)	$f_X/2^2$ (1.6 μs)
0	1	0	$f_{XX}/2^2$	$f_X/2^2$ (1.6 μs)	$f_X/2^3$ (3.2 μs)
0	1	1	$f_{XX}/2^3$	$f_X/2^3$ (3.2 μs)	$f_X/2^4$ (6.4 μs)
1	0	0	$f_{XX}/2^4$	$f_X/2^4$ (6.4 μs)	$f_X/2^5$ (12.8 μs)
Other than above			Setting prohibited		

Caution Set 0 to the bits 3 to 7.

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

2. f_X : Main system clock oscillator frequency

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

4. Figures in parentheses indicate minimum instruction execution time : $2f_{CPU}$ when operating at f_X = 5.0 MHz.

(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.

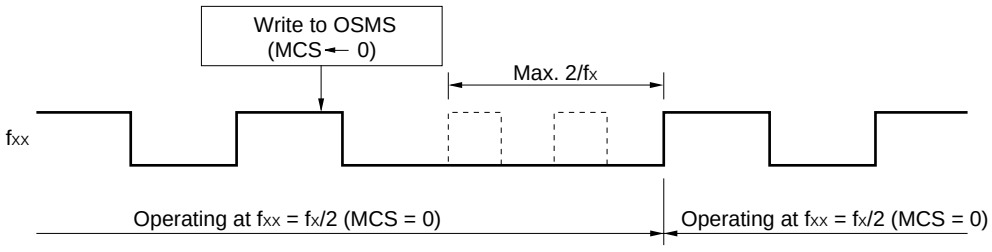
Figure 5-3. Oscillation Mode Selection Register Format

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 5-4 below, writing data (including same data as previous) to OSMS cause delay of main system clock cycle up to 2/f_x during the write operation. Therefore, if this register is written during the operation, in peripheral hardware which operates with the main system clock, a temporary error occurs in the count clock cycle of timer, etc. In addition, because the oscillation mode is changed by this register, the clocks for peripheral hardware as well as that for the CPU are switched.**

Figure 5-4. Main System Clock Waveform due to Writing to OSMS



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

Remarks f_{xx} : Main system clock frequency (fx or fx/2)

f_x : Main system clock oscillation frequency

5.4 System Clock Oscillator

5.4.1 Main system clock oscillator

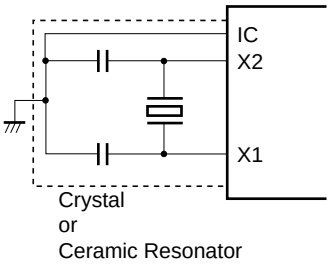
The main system clock oscillator oscillates with a crystal resonator or a ceramic resonator (standard: 5.0 MHz) connected to the X1 and X2 pins.

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

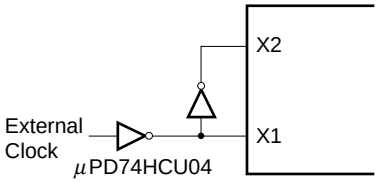
Figure 5-5 shows an external circuit of the main system clock oscillator.

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

(a) Crystal and ceramic oscillation



(b) External clock



★ **Cautions 1. Do not execute the STOP instruction if an external clock is used. This is because the X2 pin is connected to V_{DD} via a pull-up register.**

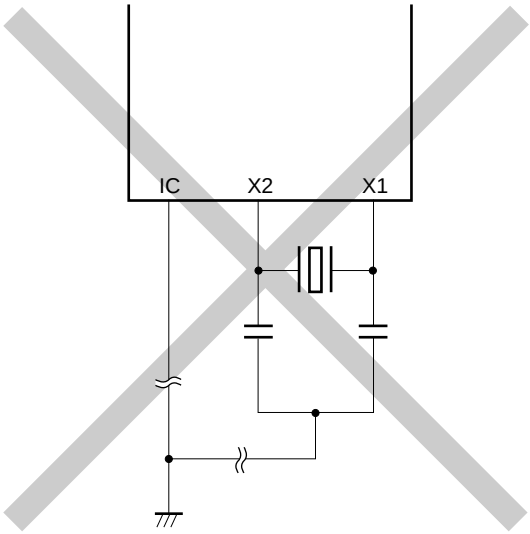
2. When using a main system clock oscillator, carry out wiring in the broken line area in Figure 5-5 to prevent any effects from wiring capacities.

- **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.**

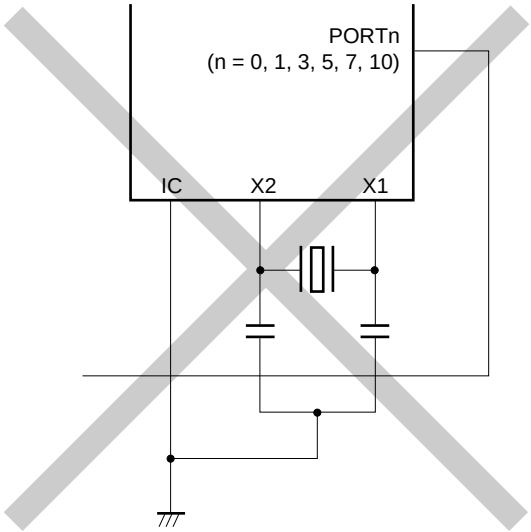
Figure 5-6 shows examples of oscillator having bad connection.

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

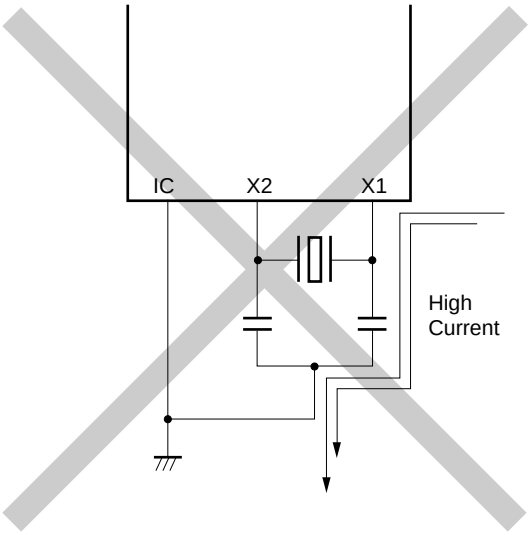
(a) Wiring of connection circuits is too long



(b) Signal conductors intersect with each other



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



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

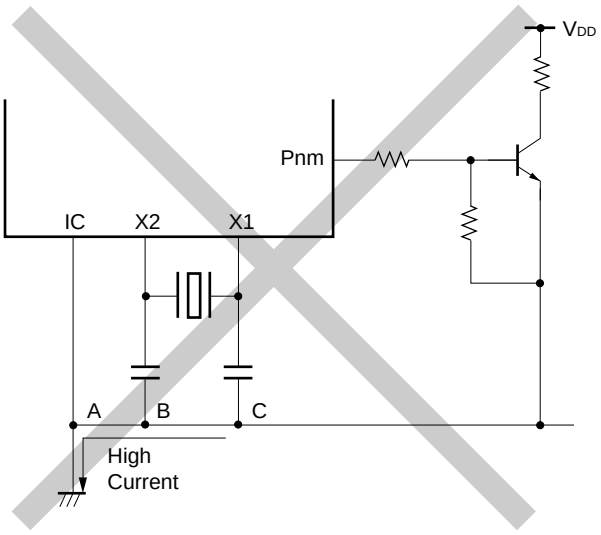
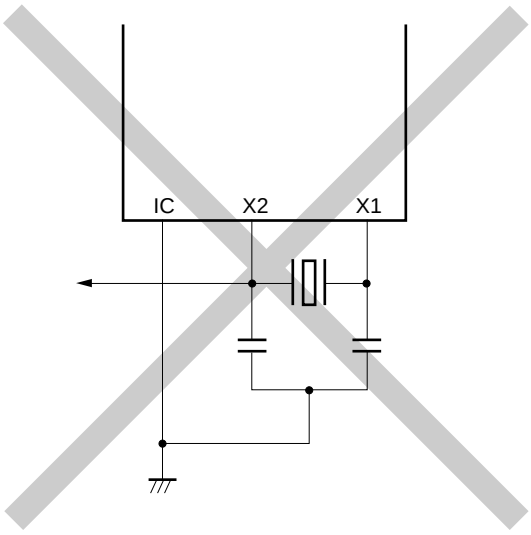


Figure 5-6. Examples of Oscillator with Bad Connection (2/2)

(c) Signals are fetched



5.4.2 Scaler

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

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}
- 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) The six types of CPU clocks (0.4 μs. 0.8 μs, 1.6 μs, 3.2 μs, 6.4μs, 12.8 μs : 5.0 MHz) can be selected by setting the PCC and OSMS.

(c) Two standby modes, the STOP and HALT modes, are available.

(d) The main system clock is divided and supplied to the peripheral hardware. Thus, the peripheral hardware also stops if the main system clock is stopped. (Except external input clock operation)

5.6 Changing CPU Clock Settings

5.6.1 Time required for CPU clock switchover

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

The actual switchover operation is not performed directly after writing to the PCC, but operation continues on the pre-switchover clock for several instructions (see **Table 5-2**).

Table 5-2. Maximum Time Required for CPU Clock Switchover

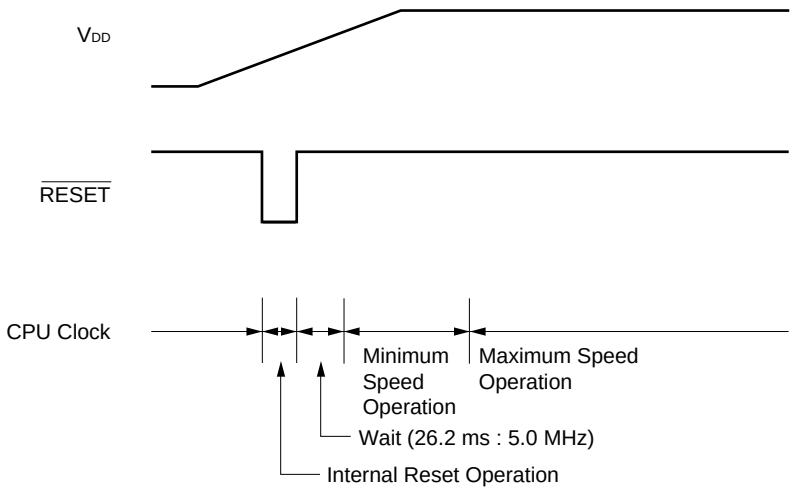
Set Values before Switchover			Set Values After Switchover														
PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0
			0	0	0	0	0	1	0	1	0	0	1	1	1	0	0
0	0	0				16 instructions			16 instructions			16 instructions			16 instructions		
0	0	1	8 instructions						8 instructions			8 instructions			8 instructions		
0	1	0	4 instructions			4 instructions						4 instructions			4 instructions		
0	1	1	2 instructions			2 instructions			2 instructions						2 instructions		
1	0	0	1 instruction			1 instruction			1 instruction			1 instruction					

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

5.6.2 CPU clock switching procedure

This section describes CPU clock switching procedure.

Figure 5-7. 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.

CHAPTER 6 8-BIT TIMER/EVENT COUNTERS 5 AND 6

The timers incorporated into the μPD78083 subseries are outlined below.

(1) 8-bit timers/event counters 5 and 6 (TM5 and TM6)

This can be used to serve as an interval timer, an external event counter, square wave output with any selected frequency PWM, etc. It cannot be used as a 16-bit timer/event counter (See **CHAPTER 6 8-BIT TIMER/EVENT COUNTERS 5 AND 6**).

(2) Watchdog timer (WDTM)

WDTM can perform the watchdog timer function or generate non-maskable interrupt requests, maskable interrupt requests and $\overline{\text{RESET}}$ at the preset time intervals (See **CHAPTER 7 WATCHDOG TIMER**).

(3) Clock output control circuit

This circuit supplies a clock obtained by dividing the main system clock, to other devices (See **CHAPTER 8 CLOCK OUTPUT CONTROL CIRCUIT**).

(4) Buzzer output control circuit

This circuit outputs the buzzer frequency obtained by dividing the main system clock (See **CHAPTER 9 BUZZER OUTPUT CONTROL CIRCUIT**).

Table 6-1 Timer/Event Counter Types and Functions

		8-bit Timer/Event Counters 5 and 6	Watchdog Timer
Type	Interval timer	2 channels	1 channel[Note]
	External event counter	√	—
Function	Timer output	√	—
	PWM output	√	—
	Square-wave output	√	—
	Interrupt request	√	√

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

6.1 8-Bit Timer/Event Counters 5 and 6 Functions

The 8-bit timer/event counters 5 and 6 (TM5 and TM6) have the following functions.

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

(1) 8-bit interval timer

Interrupt requests are generated at the preset time intervals.

Table 6-2. 8-Bit Timer/Event Counters 5 and 6 Interval Times

Minimum Interval Time		Maximum Interval Time		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
—	$1/f_X$ (200 ns)	—	$2^8 \times 1/f_X$ (51.2 μs)	—	$1/f_X$ (200 ns)
$1/f_X$ (200 ns)	$2 \times 1/f_X$ (400 ns)	$2^8 \times 1/f_X$ (51.2 μs)	$2^9 \times 1/f_X$ (102.4 μs)	$1/f_X$ (200 ns)	$2 \times 1/f_X$ (400 ns)
$2 \times 1/f_X$ (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 6-3. 8-Bit Timer/Event Counters 5 and 6 Square-Wave Output Ranges

Minimum pulse width		Maximum pulse width		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
—	$1/f_X$ (200 ns)	—	$2^8 \times 1/f_X$ (51.2 μs)	—	$1/f_X$ (200 ns)
$1/f_X$ (200 ns)	$2 \times 1/f_X$ (400 ns)	$2^8 \times 1/f_X$ (51.2 μs)	$2^9 \times 1/f_X$ (102.4 μs)	$1/f_X$ (200 ns)	$2 \times 1/f_X$ (400 ns)
$2 \times 1/f_X$ (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.

(4) PWM Output

TM5 and TM6 can generate 8-bit resolution PWM output.

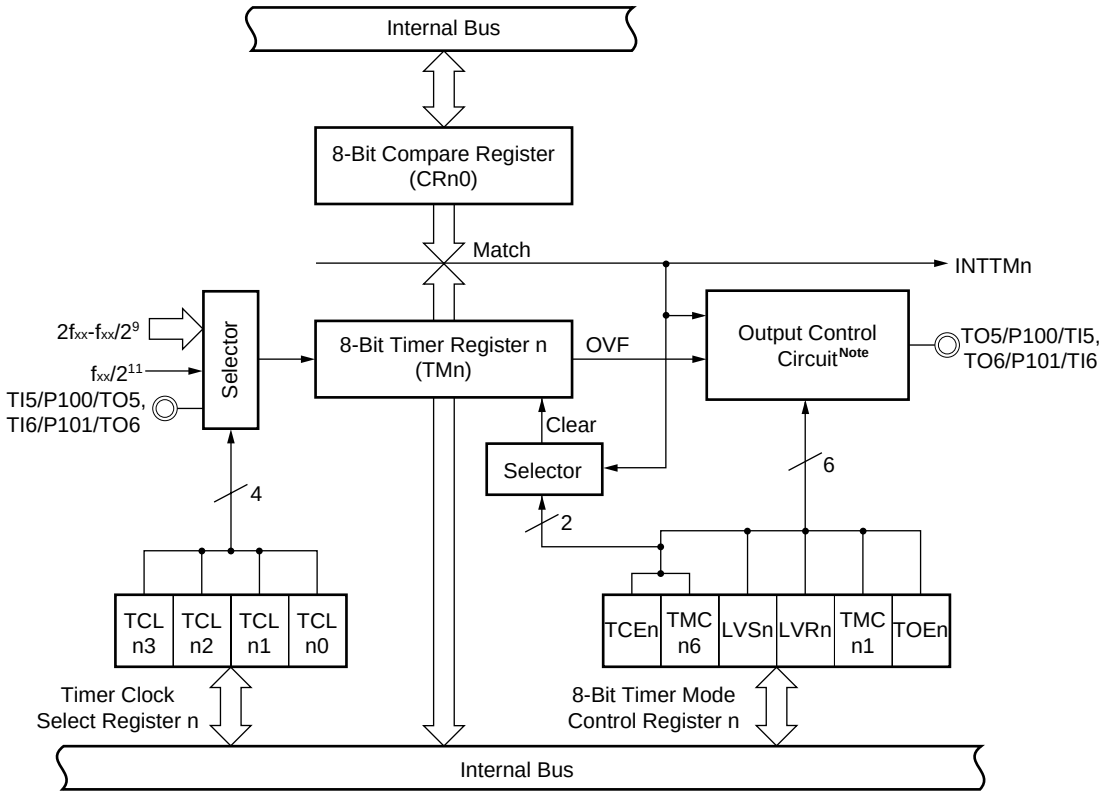
6.2 8-Bit Timer/Event Counters 5 and 6 Configurations

The 8-bit timer/event counters 5 and 6 consist of the following hardware.

Table 6-4. 8-Bit Timer/Event Counters 5 and 6 Configurations

Item	Configuration
Timer register	8 bits × 2 (TM5, TM6)
Register	Compare register: 8 bits × 2 (CR50, CR60)
Timer output	2 (TO5, TO6)
Control register	Timer clock select register 5 and 6 (TCL5, TCL6) 8-bit timer mode control register 5 and 6 (TMC5, TMC6) Port mode register 10 (PM10)

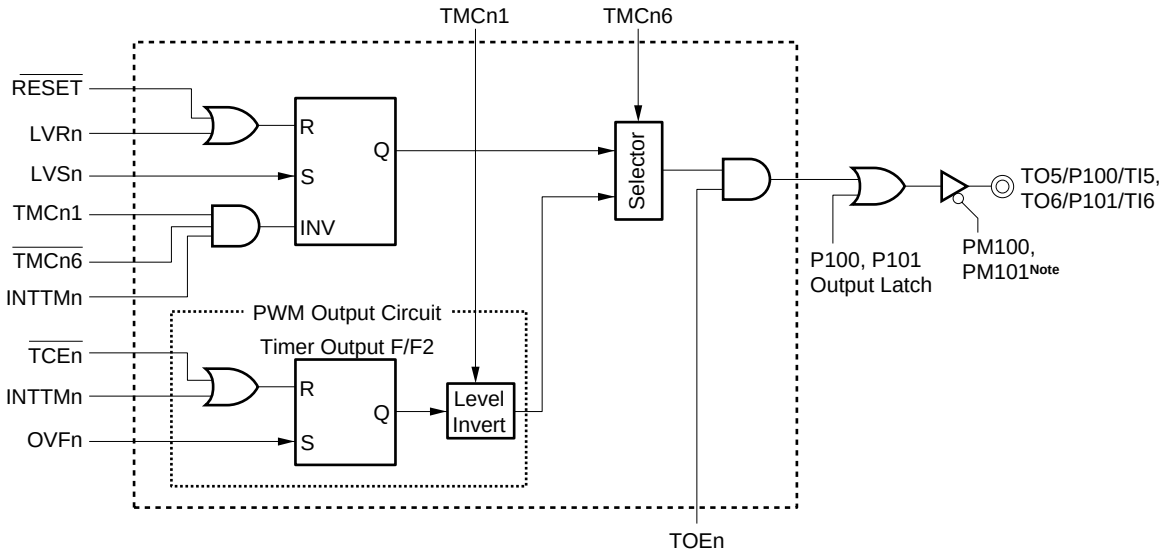
Figure 6-1. 8-Bit Timer/Event Counters 5 and 6 Block Diagram



Note Refer to Figures 6-2 for details of configurations of 8-bit timer/event counters 5 and 6 output control circuits.

Remark n = 5, 6

Figure 6-2. Block Diagram of 8-Bit Timer/Event Counters 5 and 6 Output Control Circuit



Note PM100 : Bit 0 of port mode register 10 (PM10)
PM101 : Bit 1 of PM10

Remarks 1. The section in the broken line is an output control circuit.
2. n = 5, 6

(1) Compare registers 50 and 60 (CR50, CR60)

These are 8-bit registers to compare the value set to CR50 to the 8-bit timer register 5 (TM5) count value, and the value set to CR60 to the 8-bit timer register 6 (TM6) count value, and, if they match, generate an interrupt request (INTTM5 and INTTM6, respectively).

CR50 and CR60 are set with an 8-bit memory manipulation instruction. They cannot be set with a 16-bit memory manipulation instruction. The 00H to FFH values can be set.

$\overline{\text{RESET}}$ input sets CR50 and CR60 to 00H.

Caution When using the PWM mode, please set the CRn0 value before setting TMCn (n=5, 6) to the PWM mode.

(2) 8-bit timer registers 5 and 6 (TM5, TM6)

These are 8-bit registers to count count pulses.

TM5 and TM6 are read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TM5 and TM6 to 00H.

6.3 8-Bit Timer/Event Counters 5 and 6 Control Registers

The following three types of registers are used to control the 8-bit timer/event counter 5 and 6.

- Timer clock select register 5 and 6 (TCL5, TCL6)
- 8-bit timer mode control register 5 and 6 (TMC5, TMC6)
- Port mode register 10 (PM10)

(1) Timer clock select register 5 (TCL5)

This register sets count clocks of 8-bit timer register 5.

TCL5 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TCL5 to 00H.

Figure 6-3. Timer Clock Select Register 5 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL5	0	0	0	0	TCL53	TCL52	TCL51	TCL50	FF52H	00H	R/W

TCL53	TCL52	TCL51	TCL50	8-Bit Timer Register 5 Count Clock Selection		
					MCS=1	MCS=0
0	0	0	0	TI5 falling edge[Note]		
0	0	0	1	TI5 rising edge[Note]		
0	1	0	0	$2f_{XX}$	Setting prohibited	f_X (5.0 MHz)
0	1	0	1	f_{XX}	f_X (5.0 MHz)	$f_X/2$ (2.5 MHz)
0	1	1	0	$f_{XX}/2$	$f_X/2$ (2.5 MHz)	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
1	1	1	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)
1	1	1	1	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)
Other than above				Setting prohibited		

Note The timer output (PWM output) cannot be used in cases where the clock is being input from an external source.

Caution When rewriting TCL5 to other data, stop the timer operation beforehand.

Remarks 1. f_{XX} : Main system clock frequency (f_X or $f_X/2$)
2. f_X : Main system clock oscillation frequency
3. TI5 : 8-bit timer register 5 input pin
4. MCS : Oscillation mode selection register (OSMS) bit 0
5. Values in parentheses when operated at f_X = 5.0 MHz

(2) Timer clock select register 6 (TCL6)

This register sets count clocks of 8-bit timer register 6.

TCL6 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TCL6 to 00H.

Figure 6-4. Timer Clock Select Register 6 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset
TCL6	0	0	0	0	TCL63	TCL62	TCL61	TCL60	FF56H	00H

TCL63	TCL62	TCL61	TCL60	8-bit Timer Register 6 Count Clock Selection		
					MCS=1	MCS=0
0	0	0	0	TI6 falling edge[Note]		
0	0	0	1	TI6 rising edge[Note]		
0	1	0	0	$2f_{XX}$	Setting prohibited	f_X (5.0 MHz)
0	1	0	1	f_{XX}	f_X (5.0 MHz)	$f_X/2$ (2.5 MHz)
0	1	1	0	$f_{XX}/2$	$f_X/2$ (2.5 MHz)	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
1	1	1	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)
1	1	1	1	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)
Other than above				Setting prohibited		

Note When clock is input from the external, timer output (PWM output) cannot be used.

Caution When rewriting TCL6 to other data, stop the timer operation beforehand.

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

2. f_X : Main system clock oscillation frequency

3. TI6 : 8-bit timer register 6 input pin

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

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

(3) 8-bit timer mode control register 5 (TMC5)

This register enables/stops operation of 8-bit timer register 5, sets the operating mode of 8-bit timer register 5 and controls operation of 8-bit timer/event counter 5 output control circuit.

It sets R-S type flip-flop (timer output F/F 1,2) setting/resetting, the active level in PWM mode, inversion enabling/disabling in modes other than PWM mode and 8-bit timer/event counter 5 timer output enabling/disabling.

TMC5 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMC5 to 00H.

Figure 6-5. 8-Bit Timer Mode Control Register 5 Format

Symbol	⑦	6	5	4	③	②	1	⓪	Address	After Reset	R/W
TMC5	TCE5	TMC56	0	0	LVS5	LVR5	TMC51	TOE5	FF53H	00H	R/W

TOE5	8-Bit Timer/Event Counter 5 Output Control
0	Output disabled (Port mode)
1	Output enabled

TMC51	In PWM Mode	In Other Modes
	Active level selection	Timer output F/F1 control
0	Active high	Inversion operation disabled
1	Active low	Inversion operation enabled

LVS5	LVR5	8-Bit Timer/Event Counter 5 Timer Output F/F1 Status Setting
0	0	No change
0	1	Timer output F/F1 reset (0)
1	0	Timer output F/F1 set (1)
1	1	Setting prohibited

TMC56	8-Bit Timer/Event Counter 5 Operating Mode Selection
0	Clear & start mode on match of TM5 and CR50
1	PWM mode (free-running)

TCE5	8-Bit Timer Register 5 Operation Control
0	Operation stop (TM5 clear to 0)
1	Operation enable

Cautions 1. Timer operation must be stopped before setting TMC5.

2. If LVS5 and LVR5 are read after data are set, they will be 0.

3. Set 0 to the bits 4 and 5.

(4) 8-bit timer mode control register 6 (TMC6)

This register enables/stops operation of 8-bit timer register 6, sets the operating mode of 8-bit timer register 6 and controls operation of 8-bit timer/event counter 6 output control circuit.

It sets R-S type flip-flop (timer output F/F 1,2) setting/resetting, active level in PWM mode, inversion enabling/disabling in modes other than PWM mode and 8-bit timer/event counter 6 timer output enabling/disabling.

TMC6 is set with a 1-bit or 8-bit memory manipulation instruction.

RESET input sets TMC6 to 00H.

Figure 6-6. 8-Bit Timer Mode Control Register 6 Format

Symbol	⑦	6	5	4	③	②	1	⓪	Address	After Reset	R/W
TMC6	TCE6	TMC66	0	0	LVS6	LVR6	TMC61	TOE6	FF57H	00H	R/W

TOE6	8-Bit Timer/Event Counter 6 Output Control
0	Output disabled (Port mode)
1	Output enabled

TMC61	In PWM Mode	In Other Modes
	Active level selection	Timer output F/F1 control
0	Active high	Inversion operation disabled
1	Active low	Inversion operation enabled

LVS6	LVR6	8-Bit Timer/Event Counter 6 Timer Output F/F1 Status Setting
0	0	No change
0	1	Timer output F/F1 reset (0)
1	0	Timer output F/F1 set (1)
1	1	Setting prohibited

TMC66	8-Bit Timer/Event Counter 6 Operating Mode Selection
0	Clear & start mode on match of TM6 and CR60
1	PWM mode (free-running)

TCE6	8-Bit Timer Register 6 Operation Control
0	Operation Stop (TM6 clear to 0)
1	Operation Enable

Cautions 1. Timer operation must be stopped before setting TMC6.

2. If LVS6 and LVR6 are read after data are set, they will be 0.

3. Set 0 to the bits 4 and 5.

(5) Port mode register 10 (PM10)

This register sets port 10 input/output in 1-bit units.

When using the P100/TI5/TO5 and P101/TI6/TO6 pins for timer output, set PM100, PM101, and output latches of P100 and P101 to 0.

PM10 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM10 to FFH.

Figure 6-7. Port Mode Register 10 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
PM10	1	1	1	1	1	1	PM101	PM100	FF2AH	FFH	R/W

PM10n	P10n Pin Input/Output Mode Selection (n = 0, 1)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

Caution Set 1 to the bits 2 to 7.

6.4 8-Bit Timer/Event Counters 5 and 6 Operations

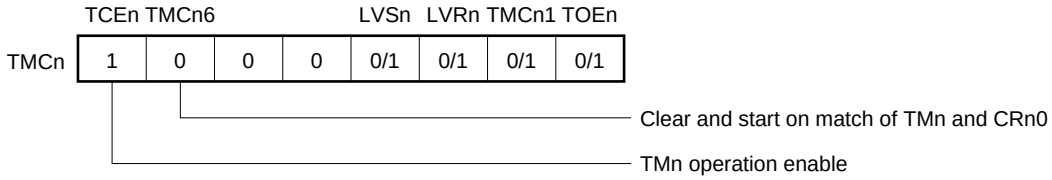
6.4.1 Interval timer operations

By setting the 8-bit timer mode control registers 5 and 6 (TMC5 and TMC6) as shown in Figure 6-8, it can be operated as an interval timer. The 8-bit timer/event counters 5 and 6 operate as interval timers which generate interrupt requests repeatedly at intervals of the count value preset to 8-bit compare registers 50 and 60 (CR50 and CR60).

When the count values of the 8-bit timer registers 5 and 6 (TM5 and TM6) match the values set to CR50 and CR60, counting continues with the TM5 and TM6 values cleared to 0 and the interrupt request signals (INTTM5 and INTTM6) are generated.

Count clock of TM5 can be selected with the timer clock select register 5 (TCL5). Count clock of TM6 can be selected with the timer clock select register 6 (TCL6).

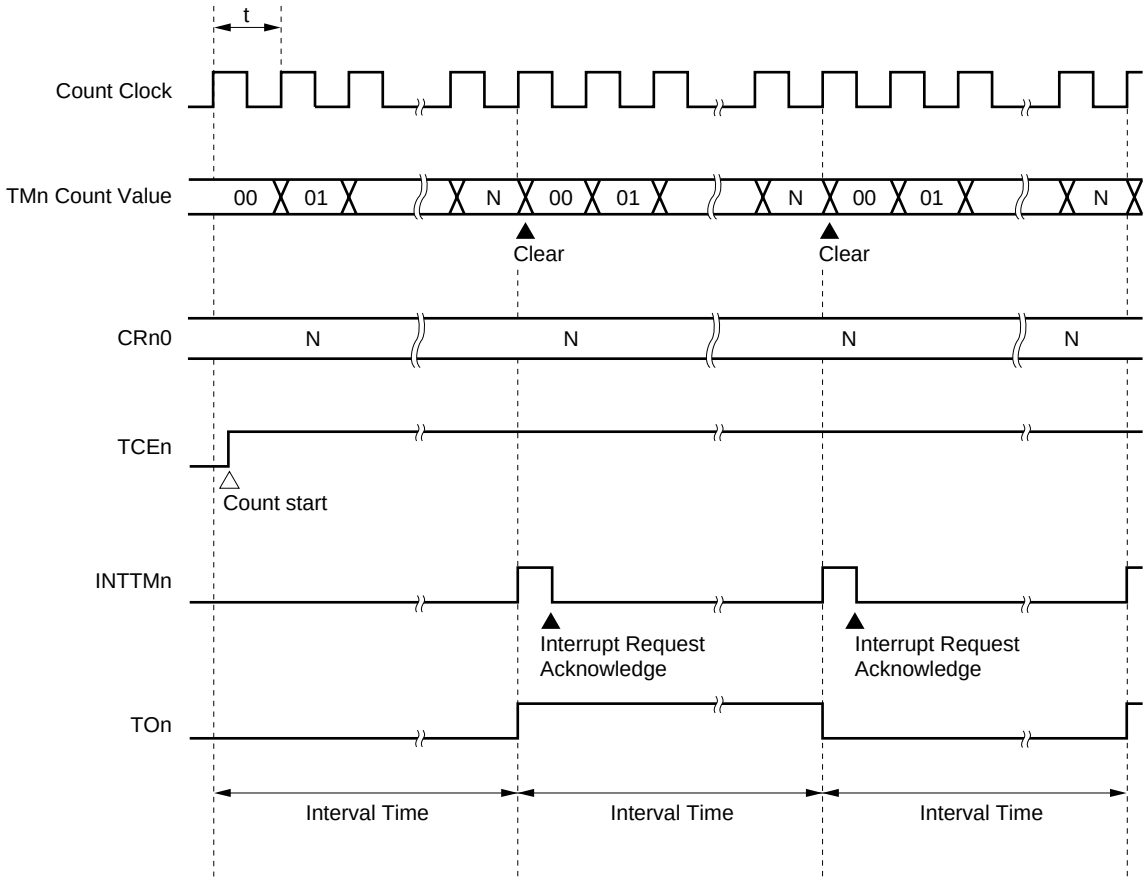
Figure 6-8. 8-Bit Timer Mode Control Register Settings for Interval Timer Operation



Remarks 1. 0/1 : Setting 0 or 1 allows another function to be used simultaneously with the interval timer. See 6.3 (3), (4) for details.

2. n = 5, 6

Figure 6-9. Interval Timer Operation Timings



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

2. n = 5, 6

Table 6-5. 8-Bit Timer/Event Counters 5 and 6 Interval Times

TCLn3	TCLn2	TCLn1	TCLn0	Minimum Interval Time		Maximum Interval Time		Resolution	
				MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
0	0	0	0	TIn input cycle		$2^8 \times$ TIn input cycle		TIn input edge cycle	
0	0	0	1	TIn input cycle		$2^8 \times$ TIn input cycle		TIn input edge cycle	
0	1	0	0	Setting prohibited	$1/f_X$ (200 ns)	Setting prohibited	$2^8 \times 1/f_X$ (51.2 μs)	Setting prohibited	$1/f_X$ (200 ns)
0	1	0	1	$1/f_X$ (200 ns)	$2 \times 1/f_X$ (400 ns)	$2^8 \times 1/f_X$ (51.2 μs)	$2^9 \times 1/f_X$ (102.4 μs)	$1/f_X$ (200 ns)	$2 \times 1/f_X$ (400 ns)
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. Values in parentheses when operated at f_X = 5.0 MHz.

4. n = 5, 6

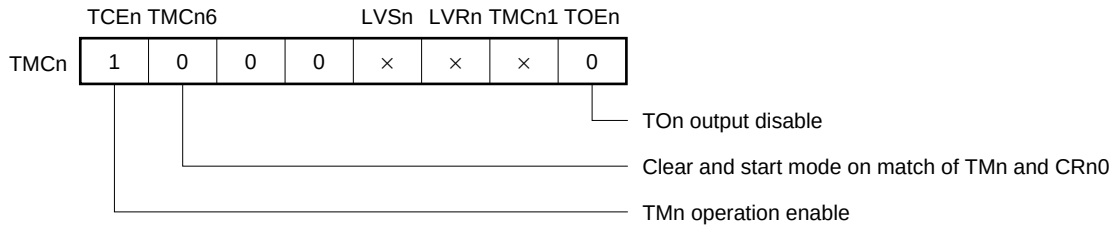
6.4.2 External event counter operation

The external event counter counts the number of external clock pulses to be input to the TI5/PI00/TO5 and TI6/P101/TO6 pins with 8-bit timer registers 5 and 6 (TM5 and TM6).

TM5 and TM6 are incremented each time the valid edge specified with the timer clock select register 5 and 6 (TCL5 and TCL6) is input. Either the rising or falling edge can be selected.

When the TM5 and TM6 counted values match the values of 8-bit compare registers 50 and 60 (CR50 and CR60), TM5 and TM6 are cleared to 0 and the interrupt request signals (INTTM5 and INTTM6) are generated.

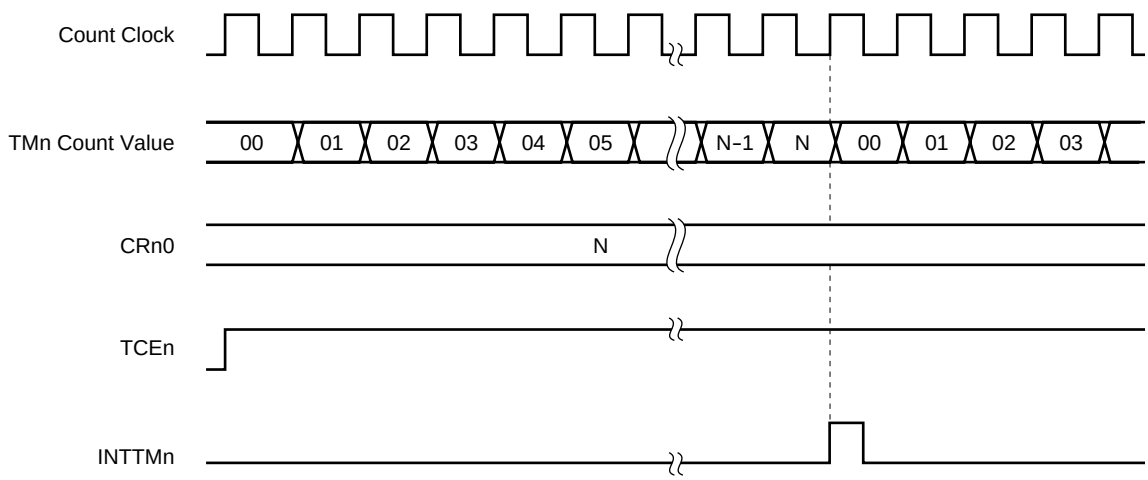
Figure 6-10. 8-Bit Timer Mode Control Register Setting for External Event Counter Operation



Remarks 1. n = 5, 6

2. × : don't care

Figure 6-11. External Event Counter Operation Timings (with Rising Edge Specification)



Remarks 1. N = 00H to FFH

2. n = 5, 6

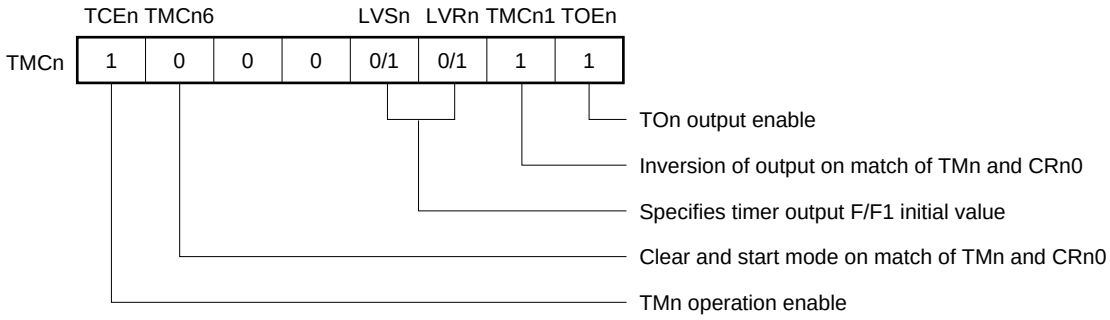
6.4.3 Square-wave output

This makes the value set in advance in the 8-bit conveyor register 50, 60 (CR50, CR60) to be the interval. It operates as a square wave output at the desired frequency.

The TO5/P100/TI5 or TO6/P101/TI6 pin output status is reversed at intervals of the count value preset to CR50 or CR60 by setting bit 1 (TMC51) and bit 0 (TOE5) of 8-bit timer output control register 5 (TMC5), or bit 1 (TMC61) and bit 0 (TOE6) of 8-bit timer mode control register 6 (TMC6) to 1.

This enables a square wave of any selected frequency to be output.

Figure 6-12. 8-Bit Timer Mode Control Register Settings for Square-Wave Output Operation



Caution When TI5/P100/TO5 or TI6/P101/TO6 pin is used as the timer output, set 0 to port mode register (PM100 or PM101) and output latch (P100 or P101).

Remark n = 5, 6

Table 6-6. 8-Bit Timer/Event Counters 5 and 6 Square-Wave Output Ranges

Minimum Pulse Width		Maximum Pulse Width		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
—	$1/f_X$ (200 ns)	—	$2^8 \times 1/f_X$ (51.2 μs)	—	$1/f_X$ (200 ns)
$1/f_X$ (200 ns)	$2 \times 1/f_X$ (400 ns)	$2^8 \times 1/f_X$ (51.2 μs)	$2^9 \times 1/f_X$ (102.4 μs)	$1/f_X$ (200 ns)	$2 \times 1/f_X$ (400 ns)
$2 \times 1/f_X$ (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.

6.4.4 PWM output operations

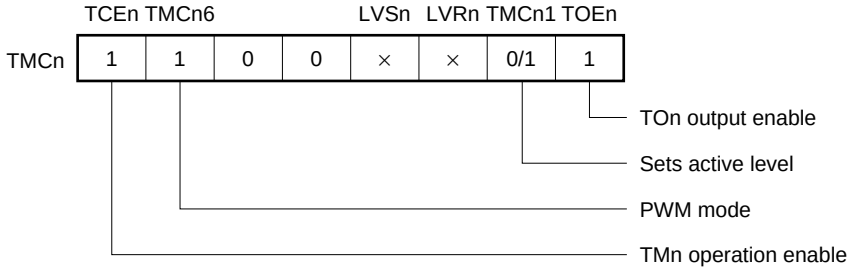
Setting the 8-bit timer mode control registers 5 and 6 (TMC5 and TMC6) as shown in Figure 6-13 allows operation as PWM output. Pulses with the duty ratio determined by the values preset in the 8-bit compare registers 50 and 60 (CR50 and CR60) output from the TO5/P100/TI5 or TO6/P101/TI6 pin.

Select the active level of PWM pulse with bit 1 (TMC51) of 8-bit timer mode control register 5 (TMC5) or bit 1 (TMC61) of 8-bit timer mode control register 6 (TMC6).

This PWM pulse has an 8-bit resolution. The pulse can be converted into an analog voltage by integrating it with an external low-pass filter (LPF). Count clock of 8-bit timer register 5 (TM5) can be selected with timer clock select register 5 (TCL5) and count clock of 8-bit timer register 6 (TM6) can be selected with timer clock select register 6 (TCL6).

PWM output enable/disable can be selected with bit 0 (TOE5) of TMC5 or bit 0 (TOE6) of TMC6.

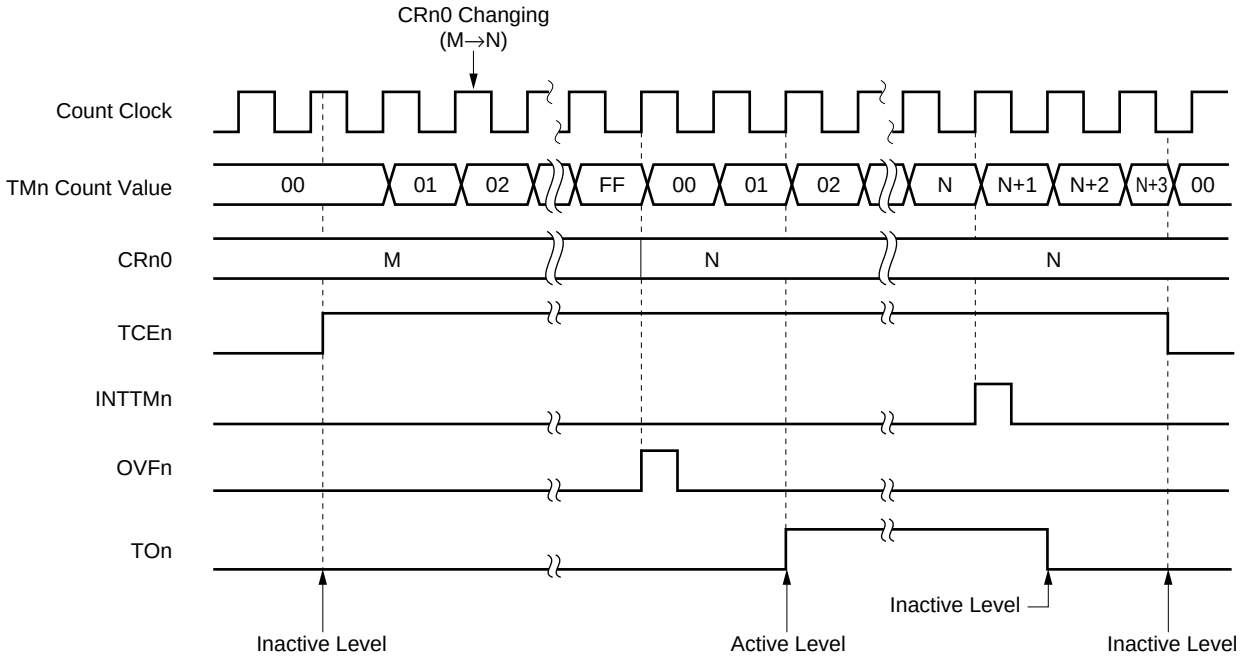
Figure 6-13. 8-Bit Timer Mode Control Register Settings for PWM Output Operation



Remarks 1. n = 5, 6

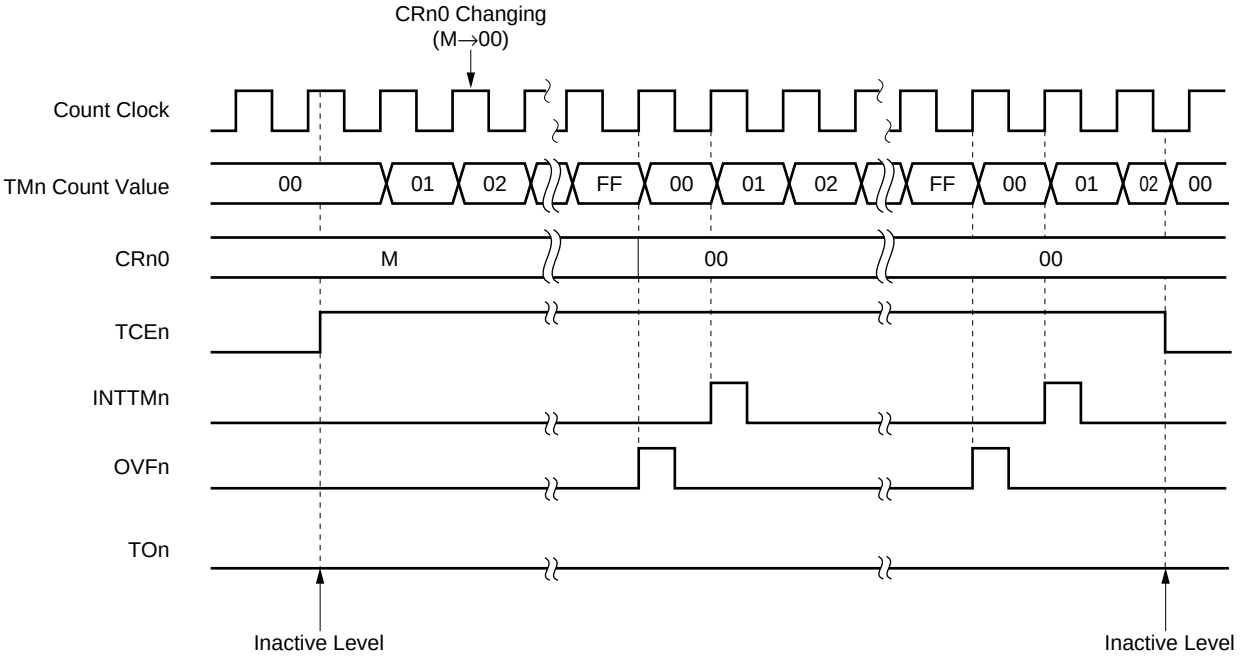
2. × : don't care

Figure 6-14. PWM Output Operation Timing (Active high setting)



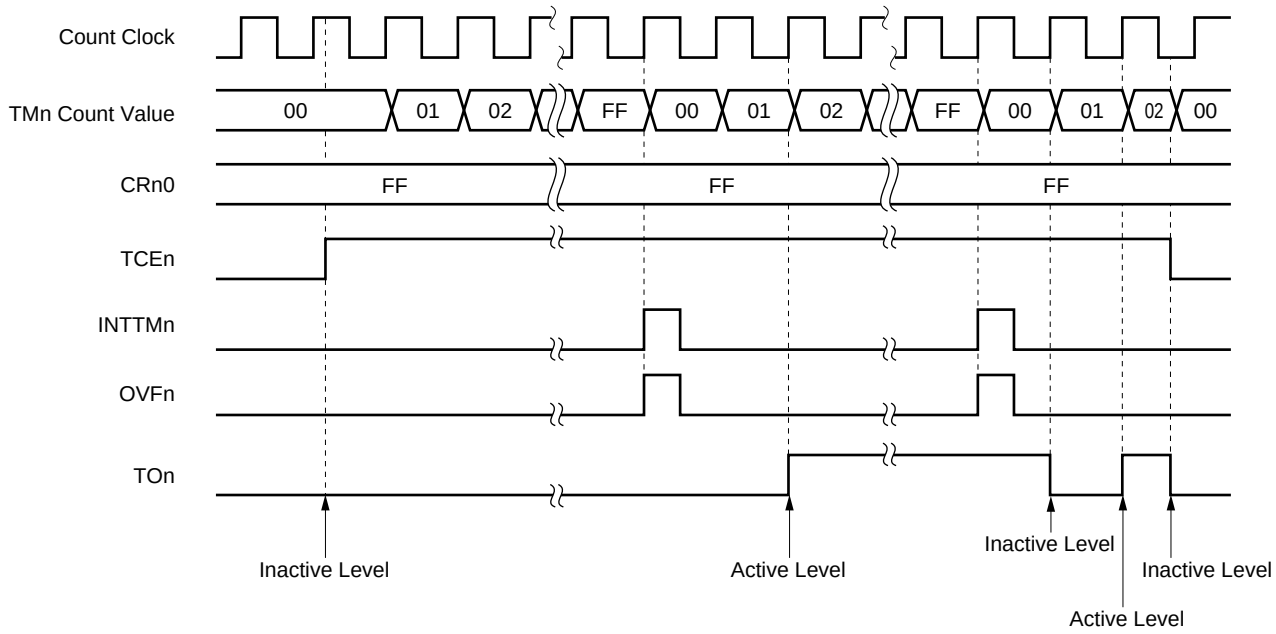
Remark n = 5, 6

Figure 6-15. PWM Output Operation Timings (CRn0 = 00H, active high setting)



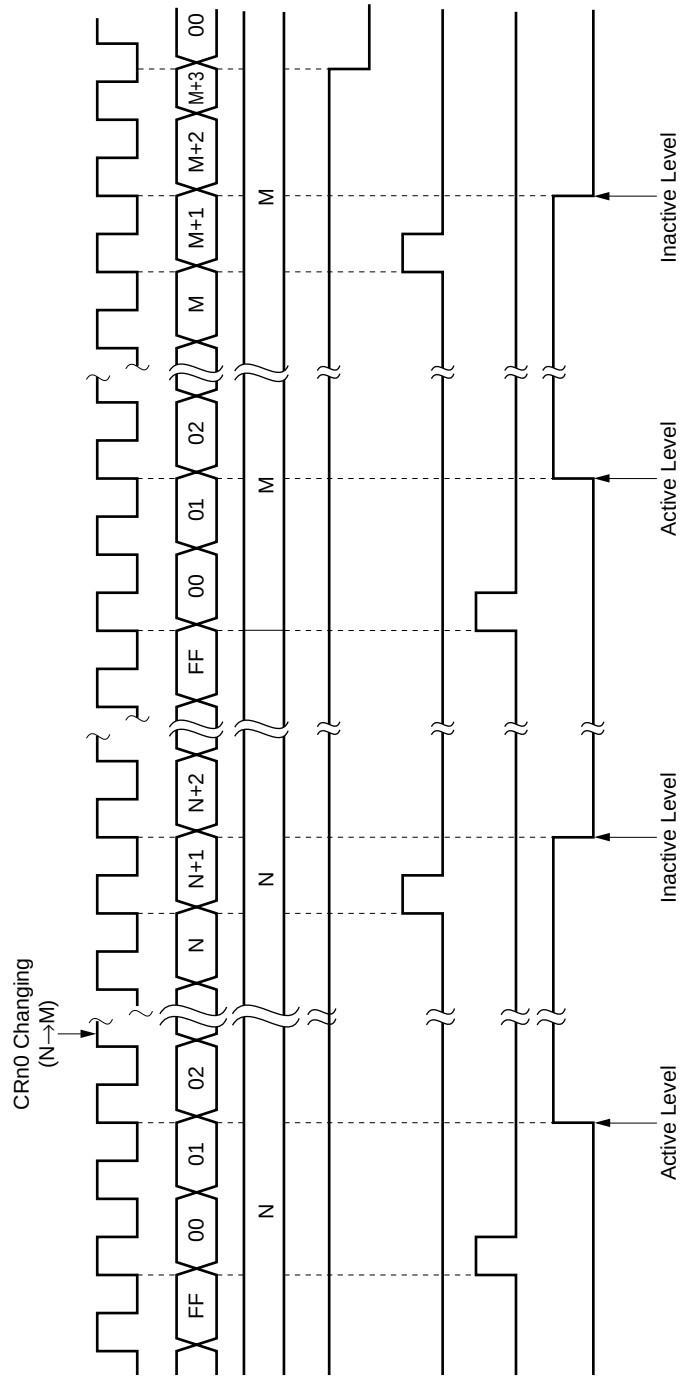
Remark n = 5, 6

Figure 6-16. PWM Output Operation Timings (CRn0 = FFH, active high setting)



Remark n = 5, 6

Figure 6-17. PWM Output Operation Timings (CRn0 changing, active high setting)



Caution If CRn0 is changed during TMn operation, the value changed is not reflected until TMn overflows.

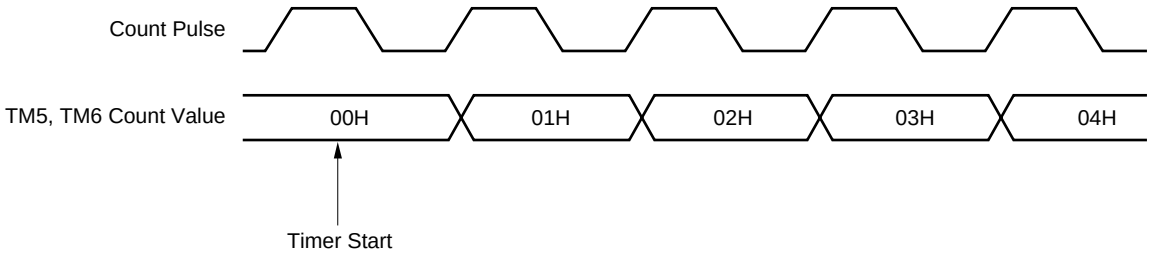
Remark n = 5, 6

6.5 Cautions on 8-Bit Timer/Event Counters 5 and 6

(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 5 and 6 (TM5 and TM6) are started asynchronously with the count pulse.

Figure 6-18. 8-Bit Timer Registers 5 and 6 Start Timing

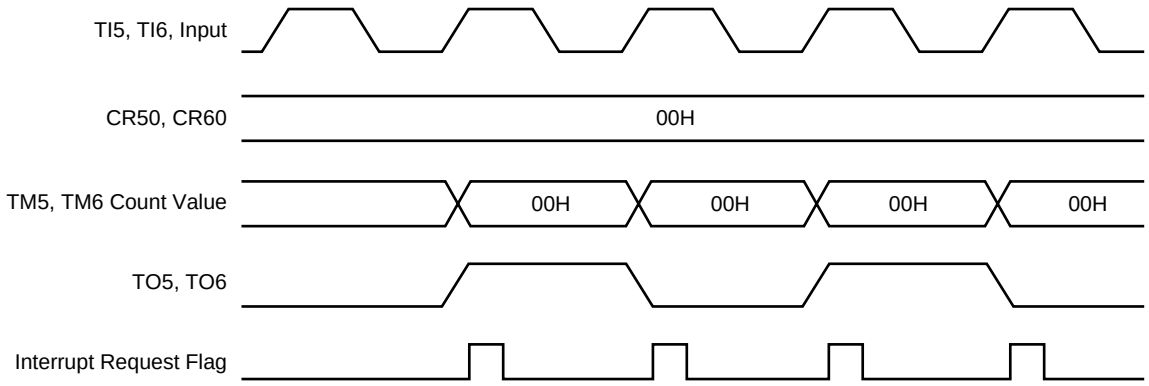


(2) 8-bit compare register 50 and 60 setting

The 8-bit compare registers 50 and 60 (CR50 and CR60) 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.

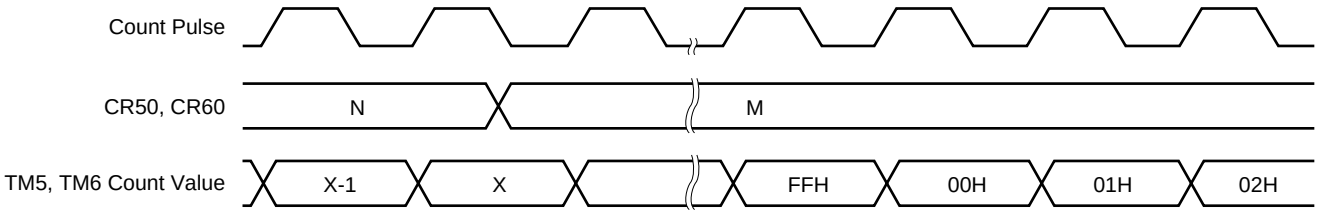
Figure 6-19. External Event Counter Operation Timing



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

If the values after the 8-bit compare registers 50 and 60 (CR50 and CR60) are changed are smaller than those of 8-bit timer registers 5 and 6 (TM5 and TM6), TM5 and TM6 continue counting, overflow and then restart counting from 0. Thus, if the value (M) after CR50 and CR60 change is smaller than value (N) before the change, it is necessary to restart the timer after changing CR50 and CR60.

Figure 6-20. Timing after Compare Register Change during Timer Count Operation



Remark N > X > M

[MEMO]

CHAPTER 7 WATCHDOG TIMER

7.1 Watchdog Timer Functions

The watchdog timer has the following functions.

- Watchdog timer
- Interval timer

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

(1) Watchdog timer mode

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

Table 7-1. Watchdog Timer Overrun Detection Times

Runaway Detection Time	MCS = 1	MCS = 0
$2^{11} \times 1/f_{XX}$	$2^{11} \times 1/f_X$ (410 μs)	$2^{12} \times 1/f_X$ (819 μs)
$2^{12} \times 1/f_{XX}$	$2^{12} \times 1/f_X$ (819 μs)	$2^{13} \times 1/f_X$ (1.64 ms)
$2^{13} \times 1/f_{XX}$	$2^{13} \times 1/f_X$ (1.64 ms)	$2^{14} \times 1/f_X$ (3.28 ms)
$2^{14} \times 1/f_{XX}$	$2^{14} \times 1/f_X$ (3.28 ms)	$2^{15} \times 1/f_X$ (6.55 ms)
$2^{15} \times 1/f_{XX}$	$2^{15} \times 1/f_X$ (6.55 ms)	$2^{16} \times 1/f_X$ (13.1 ms)
$2^{16} \times 1/f_{XX}$	$2^{16} \times 1/f_X$ (13.1 ms)	$2^{17} \times 1/f_X$ (26.2 ms)
$2^{17} \times 1/f_{XX}$	$2^{17} \times 1/f_X$ (26.2 ms)	$2^{18} \times 1/f_X$ (52.4 ms)
$2^{19} \times 1/f_{XX}$	$2^{19} \times 1/f_X$ (104.9 ms)	$2^{20} \times 1/f_X$ (209.7 ms)

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

2. f_X : Main system clock oscillation frequency

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

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

(2) Interval timer mode

Interrupt requests are generated at the preset time intervals.

Table 7-2. 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. Values in parentheses when operated at f_X = 5.0 MHz.

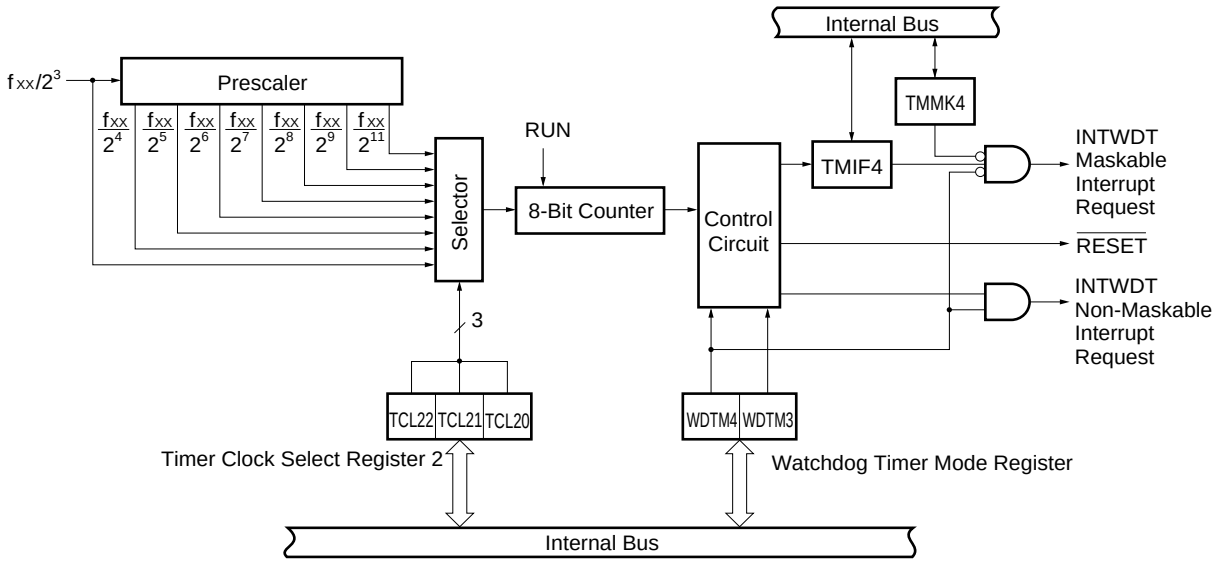
7.2 Watchdog Timer Configuration

The watchdog timer consists of the following hardware.

Table 7-3. Watchdog Timer Configuration

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

Figure 7-1. Watchdog Timer Block Diagram



7.3 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 buzzer output frequency.

Figure 7-2. Timer Clock Select Register 2 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL2	TCL27	TCL26	TCL25	0	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)

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 1. When rewriting TCL2 to other data, stop the timer operation beforehand.
2. Set 0 to the bits 3 and 4.

Remarks 1. f_{XX} : Main system clock frequency (f_X or $f_X/2$)
2. f_X : Main system clock oscillation frequency
3. × : Don't care
4. MCS : Oscillation mode selection register (OSMS) bit 0
5. Values in parentheses when operated at f_X = 5.0 MHz.

(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 7-3. 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. The watchdog timer starts operating as an interval timer as soon as RUN has been set to 1.

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

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

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

Remark ×: Don't care

7.4 Watchdog Timer Operations

7.4.1 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 overrun 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.

By setting RUN to 1, the watchdog timer can be cleared.

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

Caution The actual overrun detection time may be shorter than the set time by a maximum of 0.5 %.

Table 7-4. Watchdog Timer Overrun Detection Time

TCL22	TCL21	TCL20	Runaway Detection Time	MCS = 1	MCS = 0
0	0	0	$2^{11} \times 1/f_{XX}$	$2^{11} \times 1/f_X$ (410 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. Values in parentheses when operated at f_X = 5.0 MHz.

7.4.2 Interval timer operation

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

★ A count clock (interval time) can be selected by the bits 0 to 2 (TCL20 to TCL22) of the timer clock select register 2 (TCL2). By setting the bit 7 (RUN) of WDTM to 1, the watchdog timer starts operating as an interval timer.

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

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

Cautions 1. Once bit 4 (WDTM4) of WDTM is set to 1 (with the watchdog timer mode selected), the interval timer mode is not set unless $\overline{\text{RESET}}$ input is applied.

2. The interval time just after setting with WDTM may be shorter than the set time by a maximum of 0.5 %.

Table 7-5. 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. Values in parentheses when operated at f_X = 5.0 MHz.

CHAPTER 8 CLOCK OUTPUT CONTROL CIRCUIT

8.1 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.

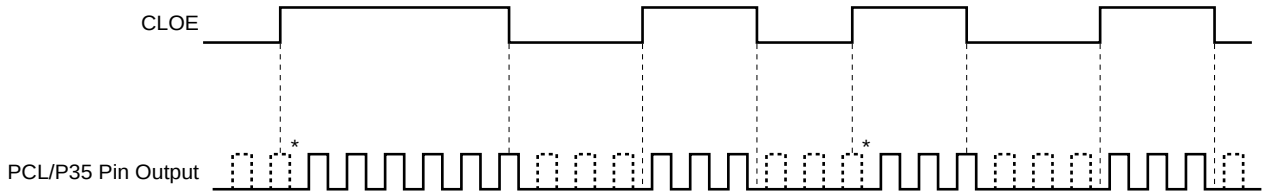
Follow the procedure below to output clock pulses.

(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 timer clock select register 0 to 1.

Caution Clock output cannot be used when setting P35 output latch to 1.

Remark When clock output enable/disable is switched, the clock output control circuit does not output pulses with small widths (See the portions marked with * in **Figure 8-1**).

Figure 8-1. Remote Controlled Output Application Example



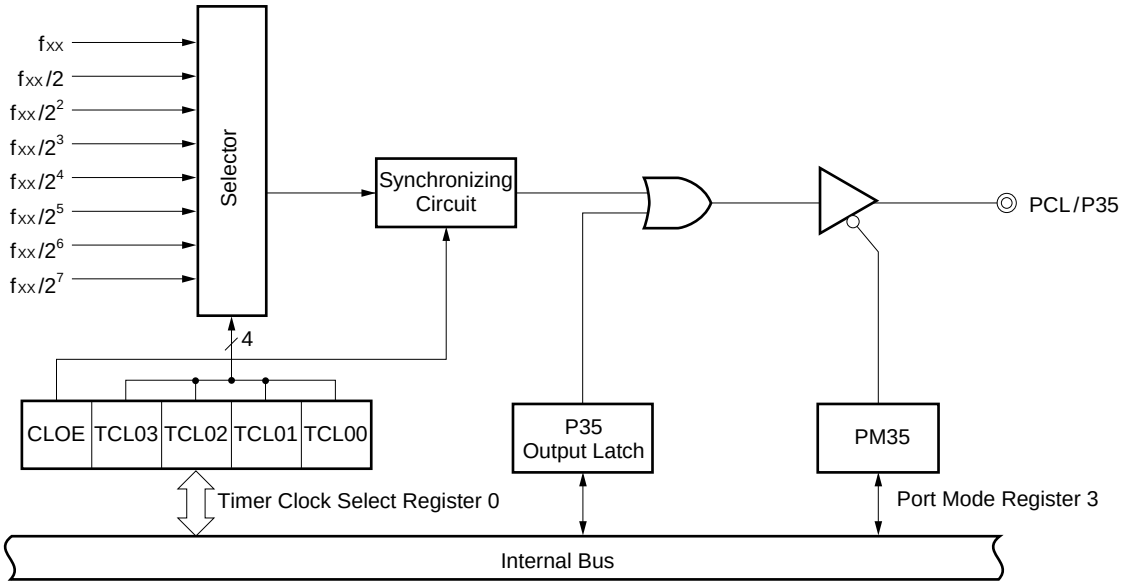
8.2 Clock Output Control Circuit Configuration

The clock output control circuit consists of the following hardware.

Table 8-1. Clock Output Control Circuit Configuration

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

Figure 8-2. Clock Output Control Circuit Block Diagram



8.3 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.

Figure 8-3. Timer Clock Select Register 0 Format

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

TCL03	TCL02	TCL01	TCL00	PCL Output Clock Selection		
					MCS=1	MCS=0
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		

CLOE	PCL Output Control
0	Output disable
1	Output enable

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

★ **2. When rewriting TCL0 to other data, stop the clock operation beforehand.**

3. Set 0 to bits 4 to 6.

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

(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.

RESET input sets PM3 to FFH.

Figure 8-4. Port Mode Register 3 Format

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

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

CHAPTER 9 BUZZER OUTPUT CONTROL CIRCUIT

9.1 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.

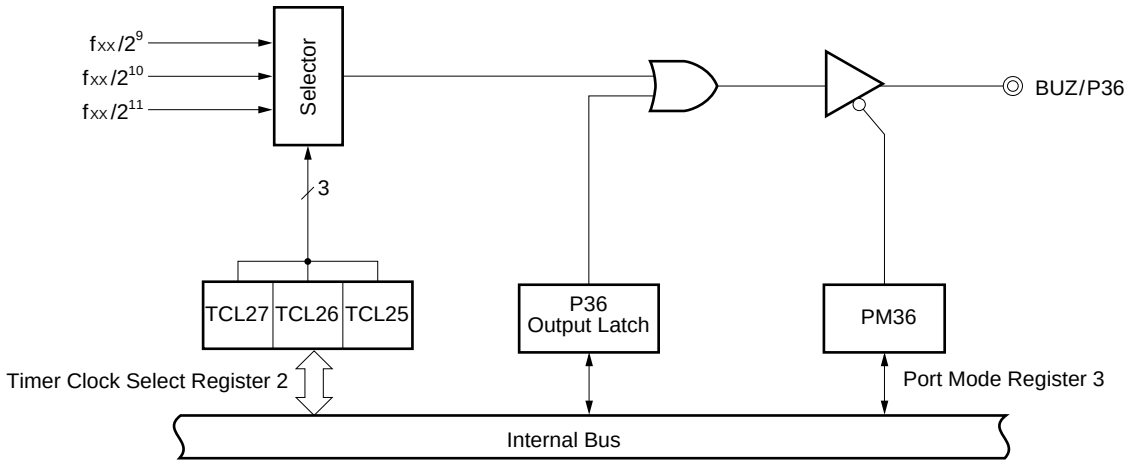
9.2 Buzzer Output Control Circuit Configuration

The buzzer output control circuit consists of the following hardware.

Table 9-1. Buzzer Output Control Circuit Configuration

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

Figure 9-1. Buzzer Output Control Circuit Block Diagram



9.3 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 watchdog timer count clock.

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	0	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)

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		

Cautions 1. When rewriting TCL2 to other data, stop the timer operation beforehand.
2. Set 0 to bits 3 and 4.

Remarks 1. f_{XX} : Main system clock frequency (f_X or $f_X/2$)
2. f_X : Main system clock oscillation frequency
3. × : don't care
4. MCS : Oscillation mode selection register (OSMS) bit 0
5. Values in parentheses when operated at f_X = 5.0 MHz

(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 9-3. 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 10 A/D CONVERTER

10.1 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.

Select 1 channel of analog input from ANI0 to ANI7 and perform A/D conversion. As for A/D conversion operations, when the hardware is started up, the A/D conversion operation stops when A/D conversion is completed, 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.

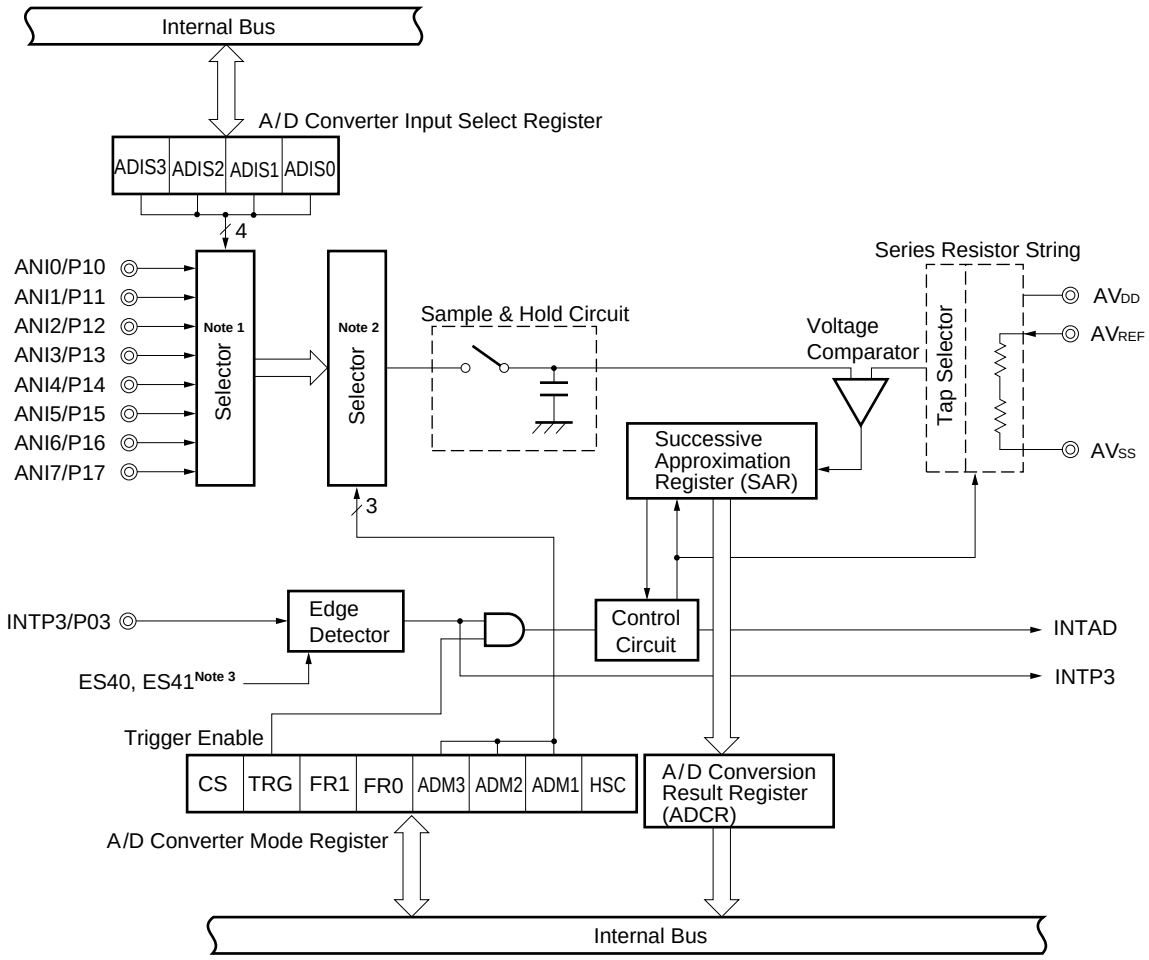
10.2 A/D Converter Configuration

The A/D converter consists of the following hardware.

Table 10-1. 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 10-1. 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. External interrupt mode register 1 (INTM1) bits 0 and 1.

(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 held to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR are transferred to the A/D conversion results register.

(2) A/D conversion result register (ADCR)

This register holds the A/D conversion result. Each time A/D conversion terminates, the conversion result is loaded from the successive approximation register.

ADCR is read with an 8-bit memory manipulation instruction.

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

(3) Sample & hold circuit

The sample & hold circuit samples each analog input signal sequentially applied from the input circuit and sends it to the voltage comparator. This circuit holds the sampled analog input voltage value during A/D conversion.

(4) Voltage comparator

The voltage comparator compares the analog input to the series resistor string output voltage.

(5) Series resistor string

The series resistor string is connected within AV_{REF} to AV_{SS} and generates a voltage for comparison with 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.

Caution 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.

(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.

(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.

10.3 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 10-2. A/D Converter Mode Register Format

Symbol	⑦	⑥	5	4	3	2	1	0	Address	After Reset	R/W
ADM	CS	TRG	FR1	FR0	ADM3	ADM2	ADM1	HSC	FF80H	01H	R/W

ADM3	ADM2	ADM1	Analog Input Channel Selection
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

★

FR1	FR0	HSC	A/D Conversion Time Selection[Note 1]			
			f_X=5.0 MHz Operation		f_X=4.19 MHz Operation	
			MCS=1	MCS=0	MCS=1	MCS=0
0	0	1	80/f_X (Setting prohibited[Note 2])	160/f_X (32.0 μs)	80/f_X (19.1 μs)	160/f_X (38.1 μs)
0	1	1	40/f_X (Setting prohibited[Note 2])	80/f_X (Setting prohibited[Note 2])	40/f_X (Setting prohibited[Note 2])	80/f_X (19.1 μs)
1	0	0	50/f_X (Setting prohibited[Note 2])	100/f_X (20.0 μs)	50/f_X (Setting prohibited[Note 2])	100/f_X (23.8 μs)
1	0	1	100/f_X (20.0 μs)	200/f_X (40.0 μs)	100/f_X (23.8 μs)	200/f_X (47.7 μs)
Other than above			Setting prohibited			

TRG	External Trigger Selection
0	No external trigger (software starts)
1	Conversion started by external trigger (hardware starts)

CS	A/D Conversion Operation Control
0	Operation stop
1	Operation start

Notes 1. Set so that the A/D conversion time is 19.1 μs or more.

2. Setting prohibited because A/D conversion time is less than 19.1 μs.

★ **Cautions 1. The following sequence is recommended for power consumption reduction of A/D converter when the standby function is used: Clear bit 7 (CS) to 0 first to stop the A/D conversion operation, and then execute the HALT or STOP instruction.**

2. When restarting the stopped A/D conversion operation, start the A/D conversion operation after clearing the interrupt request flag (ADIF) to 0.

Remarks 1. f_X : Main system clock oscillation frequency

2. MCS : 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 (PUOL).

Figure 10-3. 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.

INTM1 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets INTM1 to 00H.

Figure 10-4. External Interrupt Mode Register 1 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
INTM1	0	0	0	0	0	0	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

Caution Set 0 to the bits 2 to 7.

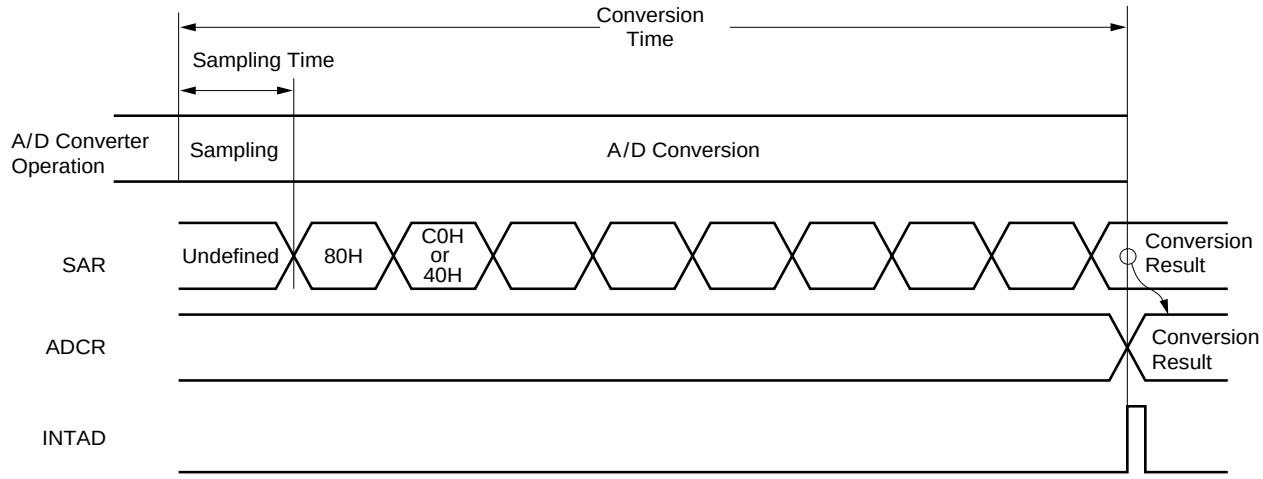
10.4 A/D Converter Operations

10.4.1 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. The series resistor string voltage tap is set at (1/2) AV_{REF} by the tap selector.
(6) The voltage difference between the series resistor string voltage tap and the analog input is compared by the voltage comparater. If the analog input is larger than (1/2) AV_{REF}, the MSB of the SAR remains set. If it 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 10-5. A/D Converter Basic Operation



A/D conversion operations are performed continuously until 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.

10.4.2 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.

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

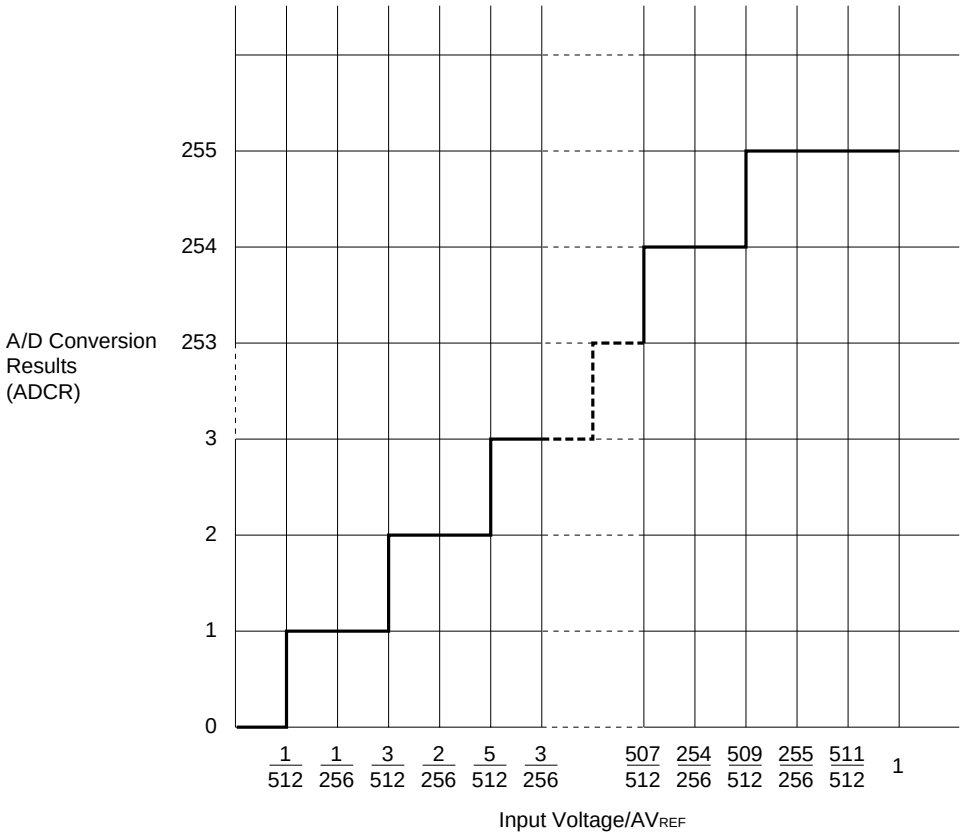
or

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

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

Figure 10-6 shows the relation between the analog input voltage and the A/D conversion result.

Figure 10-6. Relations between Analog Input Voltage and A/D Conversion Result



10.4.3 A/D converter operating mode

Using the A/D converter input select register (ADIS) and the A/D converter mode register (ADM), select one channel for the analog input from ANI0 to ANI7 and start 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

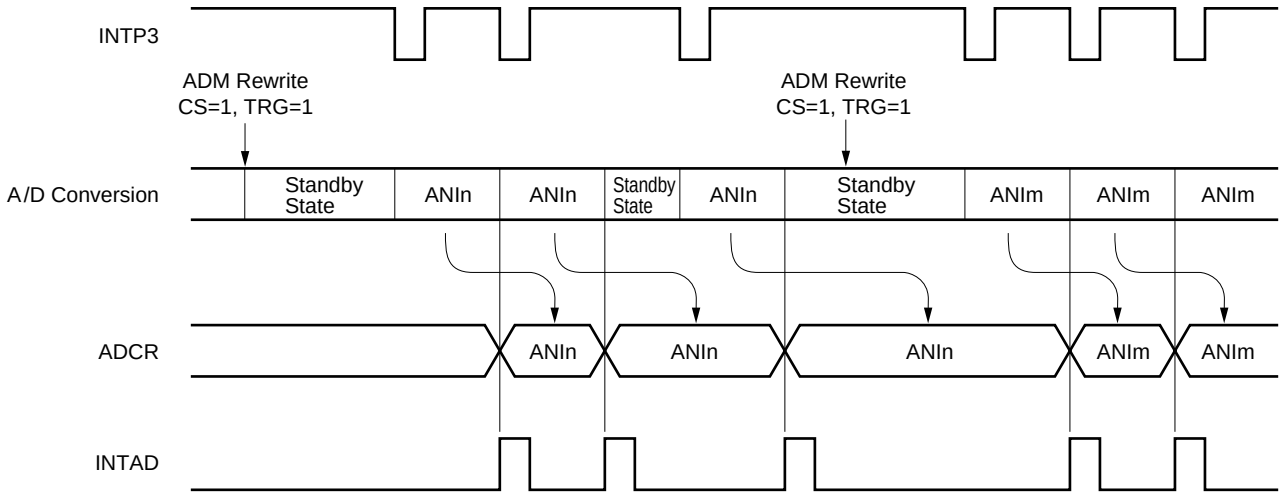
When bit 6 (TRG) and bit 7 (CS) of A/D converter mode register (ADM) are set to 1, the A/D conversion standby state is set. When the external trigger signal (INTP3) is input, the A/D conversion starts on the voltage applied to the analog input pins specified with bits 1 to 3 (ADM1 to ADM3) of ADM.

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

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

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

Figure 10-7. A/D Conversion by Hardware Start



Remarks 1. n = 0, 1, ... , 7

2. m = 0, 1, ... , 7

(2) A/D conversion operation in software start

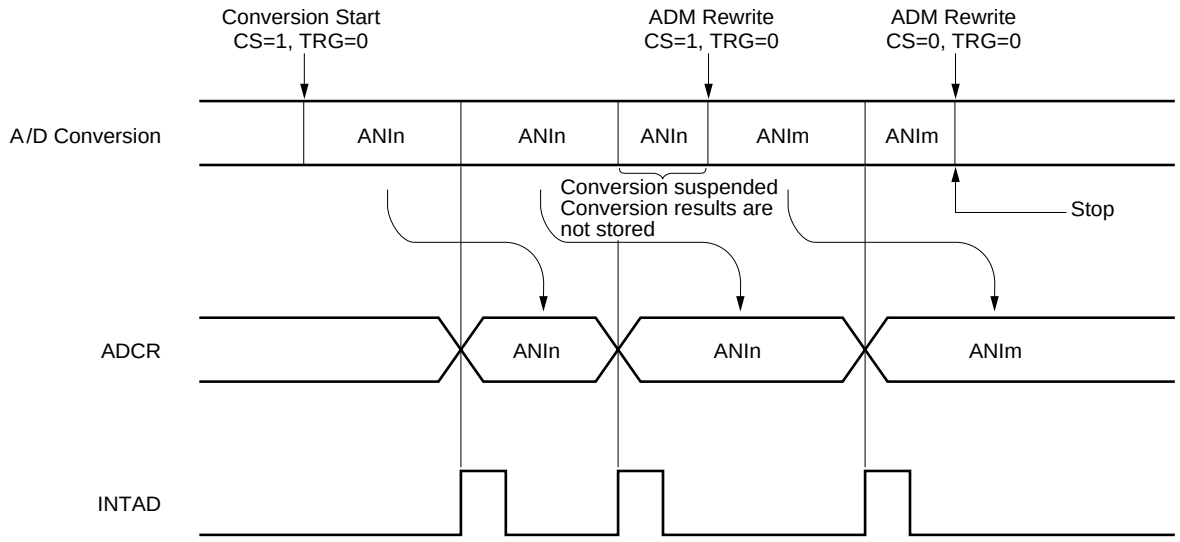
When bit 6 (TRG) and bit 7 (CS) of A/D converter mode register (ADM) are set to 0 and 1, respectively, the A/D conversion starts on the voltage applied to the analog input pins specified with bits 1 to 3 (ADM1 to ADM3) of ADM.

Upon termination of the A/D conversion, the conversion result is stored in the A/D conversion result register (ADCR) and the interrupt request signal (INTAD) is generated. After one A/D conversion operation is started and terminated, the next A/D conversion operation starts immediately. The A/D conversion operation continues repeatedly until new data is written to ADM.

If data with CS set to 1 is written to ADM again during A/D conversion, the converter suspends its A/D conversion operation and starts A/D conversion on the newly written data.

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

Figure 10-8. A/D Conversion by Software Start



Remarks 1. n = 0, 1, ... , 7

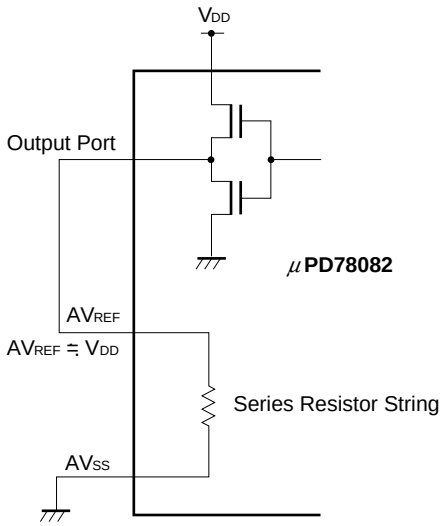
2. m = 0, 1, ... , 7

10.5 A/D Converter Cautions

(1) Power consumption in standby mode

The A/D converter operates on the main system clock. Therefore, its operation stops in STOP mode. 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 10-9, the power dissipation can be reduced by outputting a low-level signal to the output port in standby mode. However, there is no precision to the actual AV_{REF} voltage, and therefore the conversion values themselves lack precision and can only be used for relative comparison.

Figure 10-9. Example of Method of Reducing Current Dissipation in Standby Mode



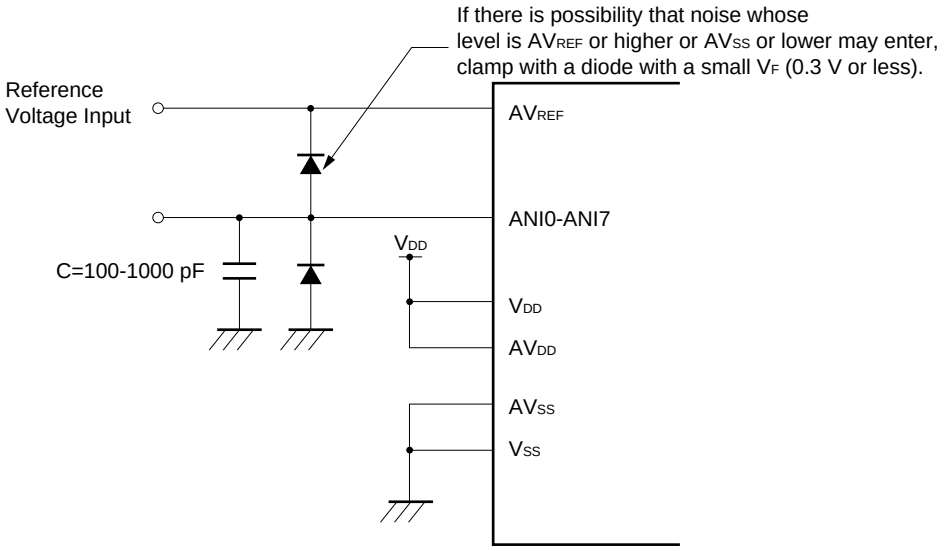
(2) Input range of ANI0 to ANI7

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 10-10 in order to reduce noise.

Figure 10-10. Analog Input Pin Disposition



(4) Pins ANI0/P10 to ANI7/P17

The analog input pins ANI0 to ANI7 also function as input/output port (PORT1) pins. When A/D conversion is performed with any of pins ANI0 to ANI7 selected, be sure not to execute 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.

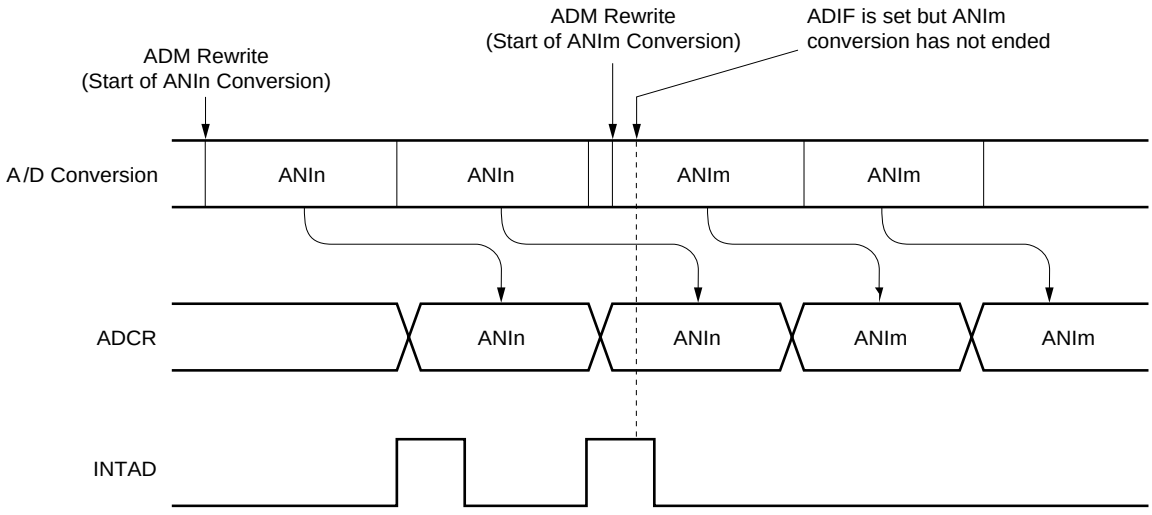
Therefore, if the output impedance of the reference voltage source is high, this will result in parallel connection to the series resistor string between the AV_{REF} pin and the AV_{SS} pin, and there will be a large reference voltage error.

(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 10-11. A/D Conversion End Interrupt Request Generation Timing

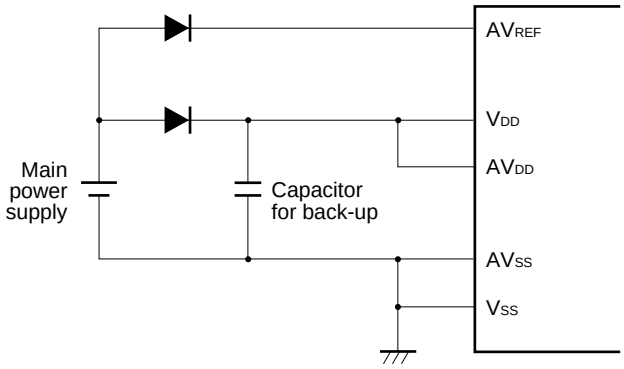


(7) AV_{DD} pin

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 shown in the following figure even when the application circuit is designed so as to switch to a backup battery.

Figure 10-12. Handling of AV_{DD} Pin



[MEMO]

CHAPTER 11 SERIAL INTERFACE CHANNEL 2

11.1 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.

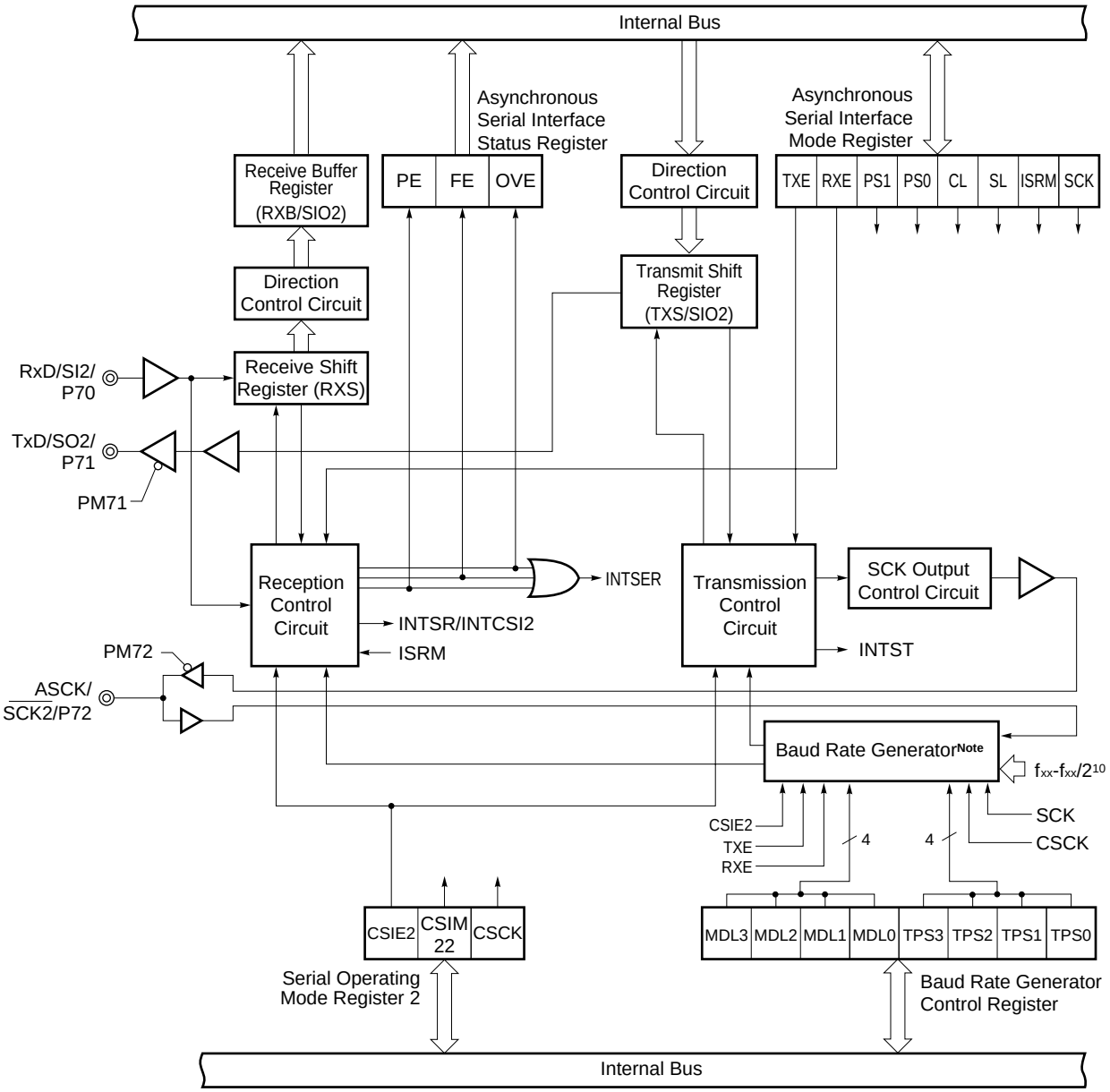
11.2 Serial Interface Channel 2 Configuration

Serial interface channel 2 consists of the following hardware.

Table 11-1. Serial Interface Channel 2 Configuration

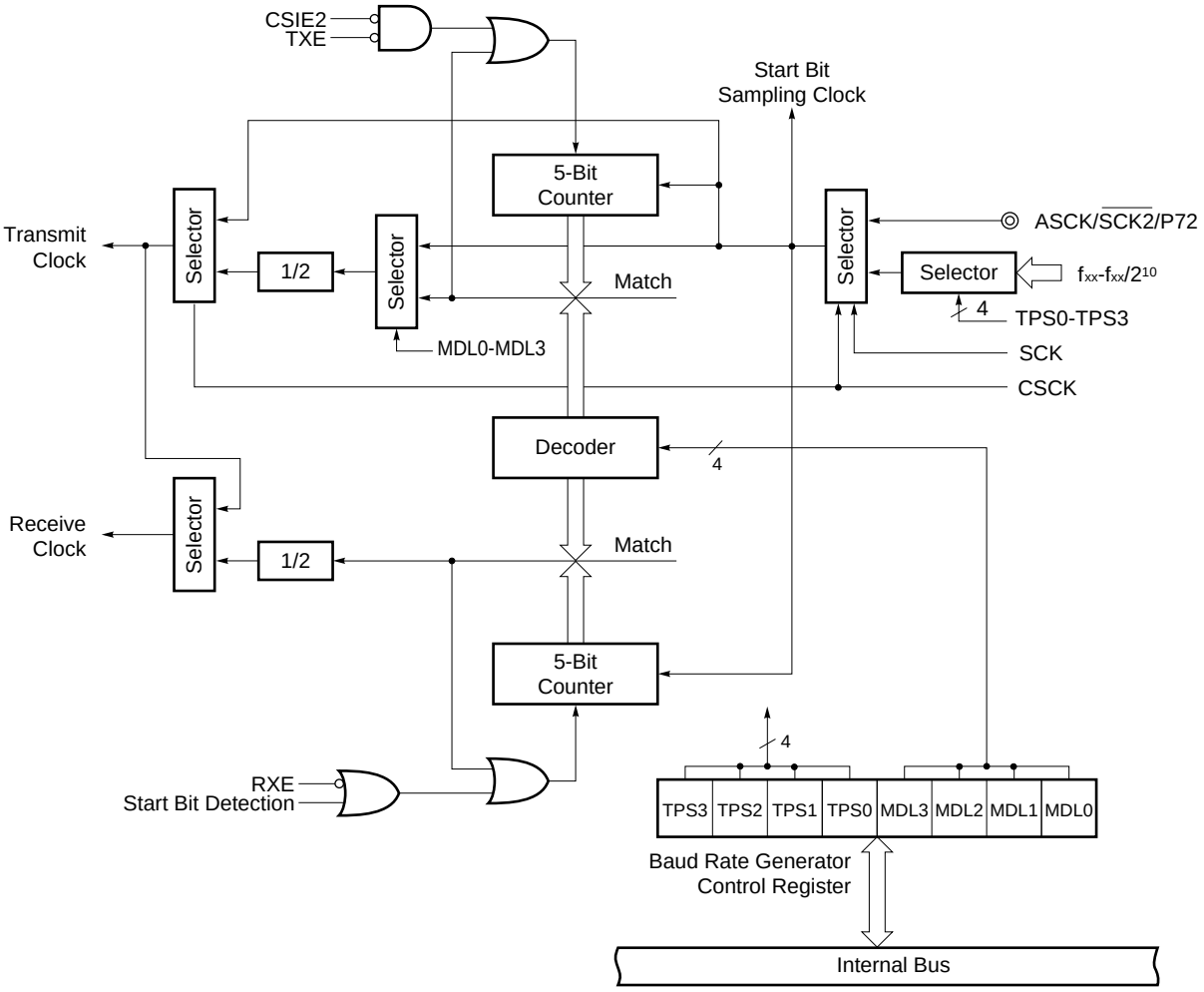
Item	Configuration
Register	Transmit shift register (TXS) Receive shift register (RXS) Receive buffer register (RXB)
Control register	Serial operating mode register 2 (CSIM2) Asynchronous serial interface mode register (ASIM) Asynchronous serial interface status register (ASIS) Baud rate generator control register (BRGC)

★ **Figure 11-1. Serial Interface Channel 2 Block Diagram**



Note See Figure 11-2 for the baud rate generator configuration.

Figure 11-2. 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.

11.3 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 11-3. Serial Operating Mode Register 2 Format

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
CSIM2	CSIE2	0	0	0	0	CSIM 22	CSCK	0	FF72H	00H	R/W

CSCK	Clock Selection in 3-wire Serial I/O Mode
0	Input clock from off-chip to $\overline{\text{SCK2}}$ pin
1	Dedicated baud rate generator output

CSIM22	First Bit Specification
0	MSB
1	LSB

CSIE2	Operation Control in 3-wire Serial I/O Mode
0	Operation stopped
1	Operation enabled

Cautions 1. Set 0 to the bits 0 and 3 to 6.

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 11-4. 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 in Case of Error Generation
0	Reception completion interrupt generated in case of error generation
1	Reception completion interrupt 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 11-2. 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	x	0	x	x	x[Note1]	x[Note1]	x[Note1]	x[Note1]	x[Note1]	x[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	1[Note2]	x[Note2]	0	1	1	x	MSB	External clock	SI2[Note2]	SO2 (CMOS output)	$\overline{SCK2}$ input
					1					0	1		Internal clock			$\overline{SCK2}$ output
			1	1	0					1	x	LSB	External clock			$\overline{SCK2}$ 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	x[Note1]	x[Note1]	0	1	1	x	LSB	External clock	P70	TxD (CMOS output)	ASCK input
		1								x[Note1]	x[Note1]		Internal clock			P72
0	1	0	0	0	0	1	x	x[Note1]	x[Note1]	1	x		External clock	RxD	P71	ASCK input
		1								x[Note1]	x[Note1]		Internal clock			P72
1	1	0	0	0	0	1	x	0	1	1	x		External clock		TxD (CMOS output)	ASCK input
		1								x[Note1]	x[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×× : Port output latch

(3) Asynchronous serial interface status register (ASIS)

This is a register which displays the type of error when a reception error is generated in the asynchronous serial interface mode.

ASIS is read with a 1-bit or 8-bit memory manipulation instruction.

In 3-wire serial I/O mode, the contents of the ASIS are undefined.

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

Figure 11-5. 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 11-6. 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 11-6. 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. Values in parentheses when operated at f_X=5.0 MHz

The baud rate transmit/receive clock generated is either a signal scaled from the main system clock, or a signal scaled from the clock input from the ASCK pin.

(a) Generation of baud rate transmit/receive clock by means of main system clock

The transmit/receive clocks generated by scaling the main system clock. The baud rate generated from the main system clock is found from the following expression.

$$[\text{Baud rate}] = \frac{f_{XX}}{2^n \times (k+16)} \text{ [Hz]}$$

where, f_X : Main system clock oscillation frequency
f_{XX} : Main system clock frequency (fx or fx/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 11-3. 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.

$$[\text{Baud rate}] = \frac{f_{ASCK}}{2 \times (k+16)} [\text{Hz}]$$

where, f_{ASCK} : Frequency of clock input to ASCK pin

k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

Table 11-4. 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

11.4 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

11.4.1 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 Set 0 to the bits 0 and 3 to 6.

(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

11.4.2 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.

When the UART mode is selected, 00H should be set in CSIM2.

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
CSIM2	CSIE2	0	0	0	0	CSIM22	CSCK	0	FF72H	00H	R/W

CSCK	Clock Selection in 3-wire Serial I/O Mode
0	Input clock from off-chip to $\overline{\text{SCK2}}$ 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 Set 0 to the bits 0 and 3 to 6.

(b) Asynchronous serial interface mode register (ASIM)

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

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

Symbol	⑦	⑥	5	4	3	2	1	0	Address	After Reset	R/W
ASIM	TXE	RXE	PS1	PS0	CL	SL	ISRM	SCK	FF70H	00H	R/W

SCK	Clock Selection in Asynchronous Serial Interface Mode
0	Input clock from off-chip to ASCK pin
1	Dedicated baud rate generator output[Note]

ISRM	Control of Reception Completion Interrupt in Case of Error Generation
0	Reception completion interrupt request generated in case of error generation
1	Reception completion interrupt request not generated in case of error generation

SL	Transmit Data Stop Bit Length Specification
0	1 bit
1	2 bits

CL	Character Length Specification
0	7 bits
1	8 bits

PS1	PS0	Parity Bit Specification
0	0	No Parity
0	1	0 parity always added in transmission No parity test in reception (parity error not generated)
1	0	Odd parity
1	1	Even parity

RXE	Receive Operation Control
0	Receive operation stopped
1	Receive operation enabled

TXE	Transmit Operation Control
0	Transmit operation stopped
1	Transmit operation enabled

Note When SCK is set to 1 and the baud rate generator output is selected, the ASCK pin can be used as an input/output port.

Caution The serial transmit/receive operation must be stopped before changing the operating mode.

(c) Asynchronous serial interface status register (ASIS)

ASIS is set with 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. Values in parentheses when operated at f_X = 5.0 MHz.

The baud rate transmit/receive clock generated is either a signal scaled from the main system clock, or a signal scaled from the clock input from the ASCK pin.

(i) Generation of baud rate transmit/receive clock by means of main system clock

The transmit/receive clock is generated by scaling the main system clock. The baud rate generated from the main system clock is obtained with the following expression.

$$[\text{Baud rate}] = \frac{f_{XX}}{2^n \times (k+16)} \text{ [Hz]}$$

where, f_X : Main system clock oscillation frequency
f_{XX} : Main system clock frequency (fx or fx/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 11-5. 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.

$$[\text{Baud rate}] = \frac{f_{ASCK}}{2 \times (k+16)} \text{ [Hz]}$$

where, f_{ASCK} : Frequency of clock input to ASCK pin

k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

Table 11-6. Relation between ASCK Pin Input Frequency and Baud Rate (When BRGC is set to 00H)

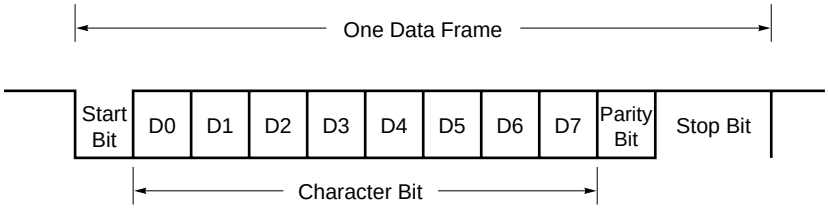
Baud Rate (bps)	ASCK Pin Input Frequency
75	2.4 kHz
110	3.52 kHz
150	4.8 kHz
300	9.6 kHz
600	19.2 kHz
1200	38.4 kHz
2400	76.8 kHz
4800	153.6 kHz
9600	307.2 kHz
19200	614.4 kHz
31250	1000.0 kHz
38400	1228.8 kHz

(2) Communication operation

(a) Data format

The transmit/receive data format is as shown in Figure 11-7.

Figure 11-7. Asynchronous Serial Interface Transmit/Receive Data Format



1 Data frame is configured from 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

- **Transmission**

  The number of bits with a value of “1”, including the parity bit, in the transmit data is controlled to be even.

  The value of the parity bit is as follows:

  Number of bits with a value of “1” in transmit data is odd: 1
  Number of bits with a value of “1” in transmit data is even: 0

- **Reception**

  The number of bits with a value of “1”, including the parity bit, in the receive data is counted. If it is odd, a parity error occurs.

(ii) Odd parity

- **Transmission**

  Conversely to the situation with even parity, the number of bits with a value of “1”, including the parity bit, in the transmit data is controlled to be odd. The value of the parity bit is as follows:

  Number of bits with a value of “1” in transmit data is odd: 0
  Number of bits with a value of “1” in transmit data is even: 1

- **Reception**

  The number of bits with a value of “1”, including the parity bit, in the receive data is counted. If it is even, a parity error occurs.

(iii) 0 Parity

When transmitting, the parity bit is set to "0" irrespective of the transmit data.
At reception, a parity bit check is not performed. Therefore, a parity error is not generated, irrespective of whether the parity bit is set to "0" or "1".

(iv) No parity

A parity bit is not added to the transmit data. At reception, data is received assuming that there is no parity bit. Since there is no parity bit, a parity error is not generated.

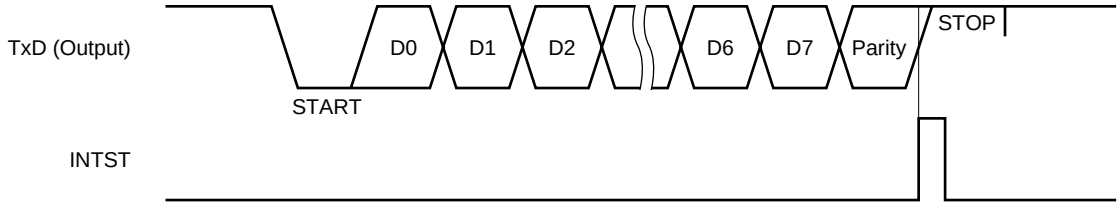
(c) Transmission

A transmit operation is started by writing transmit data to the transmit shift register (TXS). The start bit, parity bit and stop bit(s) are added automatically.

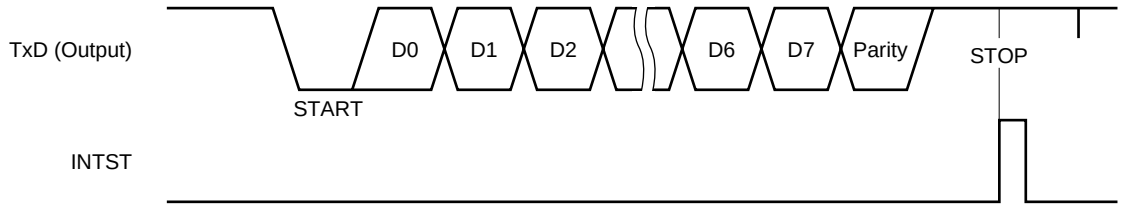
When the transmit operation starts, the data in the transmit shift register (TXS) is shifted out, and when the transmit shift register (TXS) is empty, a transmission completion interrupt request (INTST) is generated.

Figure 11-8. 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.

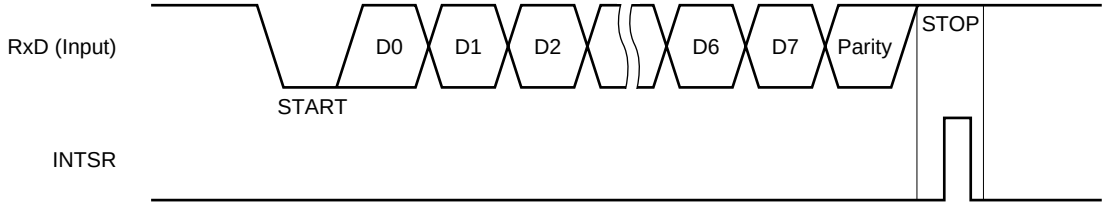
When the RxD pin input becomes low, the 5-bit counter of the baud rate generator (see Figure 11-2) starts counting, and at the time when the half time determined by specified baud rate has passed, the data sampling start timing signal is output. If the RxD pin input sampled again as a result of this 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 11-9. 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.

★ When a data reception results error flag is set in the asynchronous serial interface register (ASIS), a reception error interrupt request (INTSER) is generated.

The reception error interrupt request is generated first before the reception completed interrupt request (INTSR). Receive error causes are shown in Table 11-7.

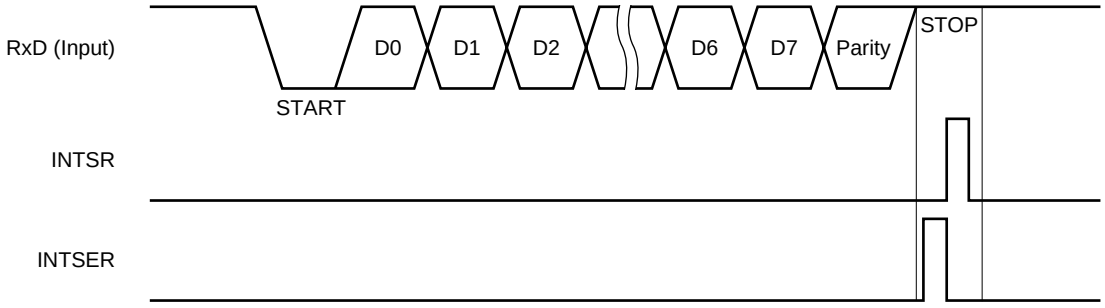
It is possible to determine what kind of error was generated during reception by reading the contents of the asynchronous serial interface status register (ASIS) in the reception error interrupt servicing (INTSER) (see **Figures 11-9** and **11-10**).

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 11-7. 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 11-10. 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.

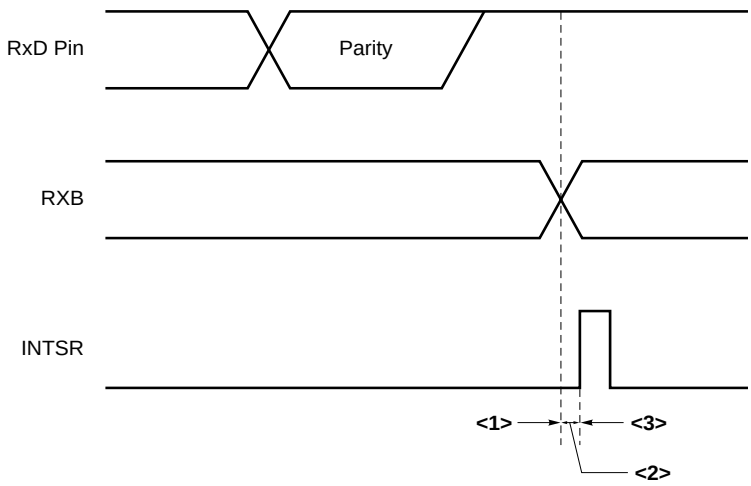
2. The receive buffer register (RXB) must be read even if a receive error is generated. If RXB is not read, an overrun error will be generated when the next data is received, and the receive error state will continue indefinitely.

(3) UART mode cautions

(a) In cases where bit 7 (TXE) of the asynchronous serial interface mode register (ASIM) has been cleared and a transmit operation has been terminated during transmission, be sure to set 1 in TXE after setting FFH in the transmit shift register (TXS) before executing the next transmission.

(b) In cases where bit 6 (RXE) of the asynchronous serial interface mode register (ASIM) has been cleared (0) and a receive operation terminated during reception, enable/disable will differ depending on the timing, the condition of the receive buffer register (RXB), and generation of the reception completed interrupt request (INTSR) . The timing is displayed in Figure 11-11.

Figure 11-11 State of the Receive Buffer Register (RXB) when Reception is Interrupted, and Generation/ Non Generation of an 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.

11.4.3 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	CSIM 22	CSCK	0	FF72H	00H	R/W

CSCK	Clock Selection in 3-wire Serial I/O Mode
0	Input clock from off-chip to $\overline{SCK2}$ 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 Set 0 to the bits 0 and 3 to 6.

(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 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}	—

(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. Values in parentheses when operated at 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 the same as the source clock frequency for the 5-bit counter.

(ii) When the baud rate generator is used:

Select a serial clock frequency with TPS0-TPS3. Be sure then to set MDL0 to MDL3 to 1,1,1,1.

The serial clock frequency is calculated by the following formula:

$$\text{Serial clock frequency} = \frac{f_{XX}}{2^n \times (k + 16)} \text{ [Hz]}$$

Remarks 1. f_X : Main system clock oscillation frequency

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

3. n : Value set in TPS0 to TPS3 ($1 \leq n \leq 11$)

4. k : Value set in MDL0 to MDL3 ($0 \leq k \leq 14$)

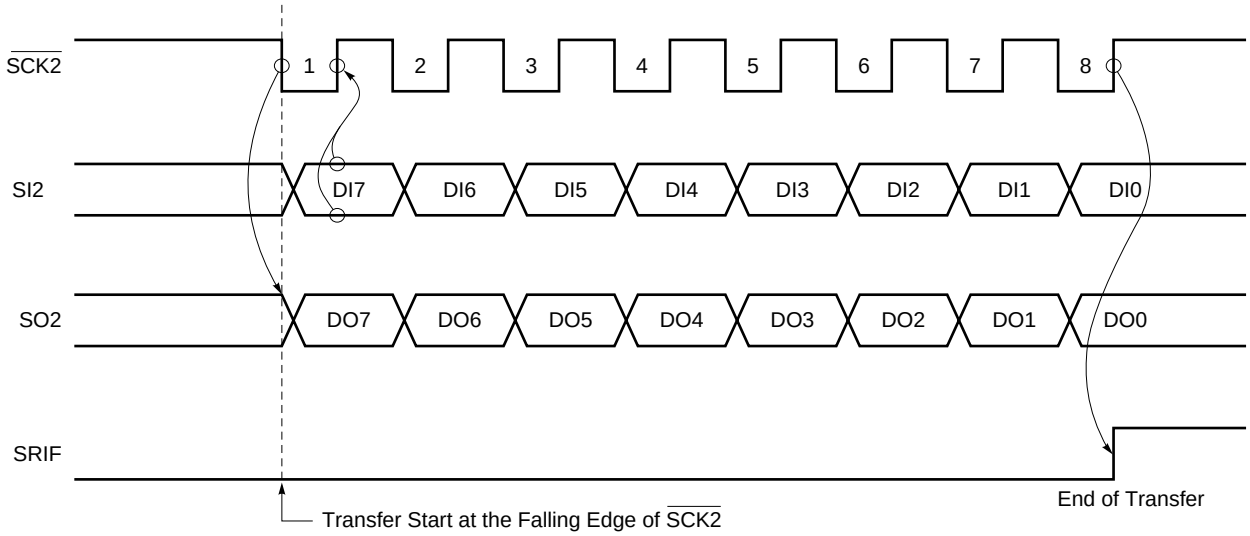
(2) Communication operation

In the 3-wire serial I/O mode, data transmission/reception is performed in 8-bit units. Data is transmitted/received bit by bit in synchronization with the serial clock.

Transmit shift register (TXS/SIO2) and receive shift register (RXS) shift operations are performed in synchronization with the fall of the serial clock $\overline{SCK2}$. 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 TXS/SIO2 or RXS stops automatically, and the interrupt request flag (SRIF) is set.

Figure 11-12. 3-Wire Serial I/O Mode Timing



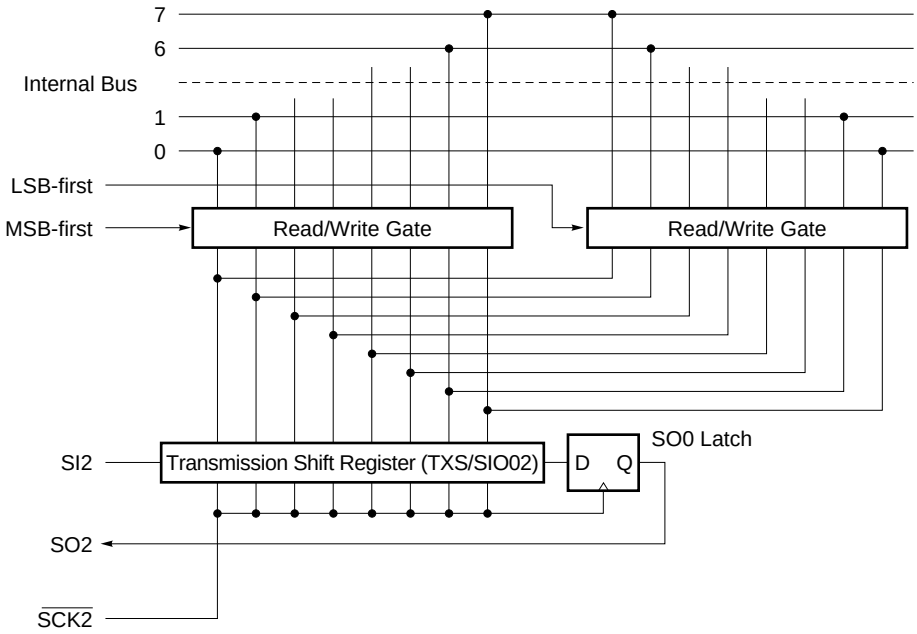
★ **(3) MSB/LSB switching as the start bit**

The 3-wire serial I/O mode enables to select transfer to start from MSB or LSB.

Figure 11-13 shows the configuration of the transmission shift register (TXS/SIO2) and internal bus. As shown in the figure, MSB/LSB can be read/written in reverse form.

MSB/LSB switching as the start bit can be specified with bit 2 (CSIM22) of the serial operating mode register 2 (CSIM2).

★ **Figure 11-13. Circuit of Switching in Transfer Bit Order**



Start bit switching is realized by switching the bit order for data write to SIO2. The SIO2 shift order remains unchanged.

Thus, switching between MSB-first and LSB-first must be performed before writing data to the shift register.

(4) Transfer start

Serial transfer is started by setting transfer data to the transmission shift register (TXS/SIO2) when the following two conditions are satisfied.

- Serial interface channel 2 operation control bit (CSIE2) =1
- Internal serial clock is stopped or $\overline{SCK2}$ is a high level after 8-bit serial transfer.

Caution If CSIE2 is set to "1" after data write to TXS/SIO2, transfer does not start.

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (SRIF) is set.

[MEMO]

CHAPTER 12 INTERRUPT FUNCTION

12.1 Interrupt Function Types

The following three types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally even in the interrupt disabled status. It does not undergo interrupt priority control and is given top priority over all other interrupt requests.
It generates a standby release signal.
One of the non-maskable interrupts is the interrupt request from the Watchdog Timer.

(2) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into a high interrupt priority group and a low interrupt priority group by setting the priority specify flag register (PR0L, PR0H, PR1L).
Multiple high priority interrupts can be applied to low priority interrupts. If two or more interrupts with the same priority are simultaneously generated, each interrupts has a predetermined priority (see **Table 12-1**).
A standby release signal is generated.
Maskable interrupts include three external interrupt requests and eight internal interrupt requests.

(3) Software interrupt

This is a vectored interrupt that occurs when the BRK instruction is executed. It is acknowledged even in a disabled state. The software interrupt does not undergo interrupt priority control.

12.2 Interrupt Sources and Configuration

There are a total of 13 interrupts, combining non-maskable interrupts, maskable interrupts and software interrupts (see **Table 12-1**).

Table 12-1. Interrupt Source List

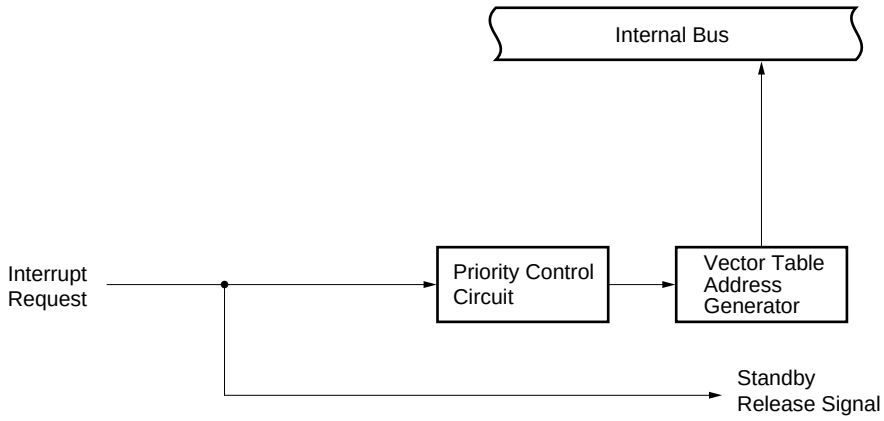
Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2
		Name	Trigger			
Non-maskable	—	INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (with interval timer mode selected)			(B)
	1	INTP1	Pin input edge detection	External	0008H	(C)
	2	INTP2			000AH	
	3	INTP3			000CH	
	4	INTSER	Serial interface channel 2 UART reception error generation	Internal	0018H	(B)
	5	INTSR	End of serial interface channel 2 UART reception		001AH	
		INTCSI2	End of serial interface channel 2 3-wire transfer			
	6	INTST	End of serial interface channel 2 UART transfer		001CH	
	7	INTAD	End of A/D converter conversion		0028H	
	8	INTTM5	Generation of 8-bit timer/event counter 5 match signal		002AH	
	9	INTTM6	Generation of 8 bit timer/event counter 6 match signal		002CH	
Software	—	BRK	BRK instruction execution	—	003EH	(D)

Notes 1. Default priorities are intended for two or more simultaneously generated maskable interrupts. 0 is the highest priority and 9 is the lowest priority.

2. Basic configuration types (A) to (D) correspond to (A) to (D) of Figure 12-1.

Figure 12-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt

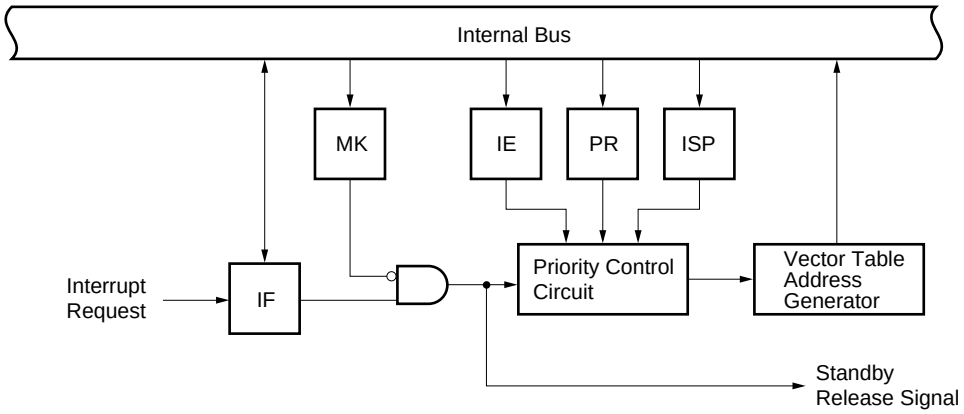
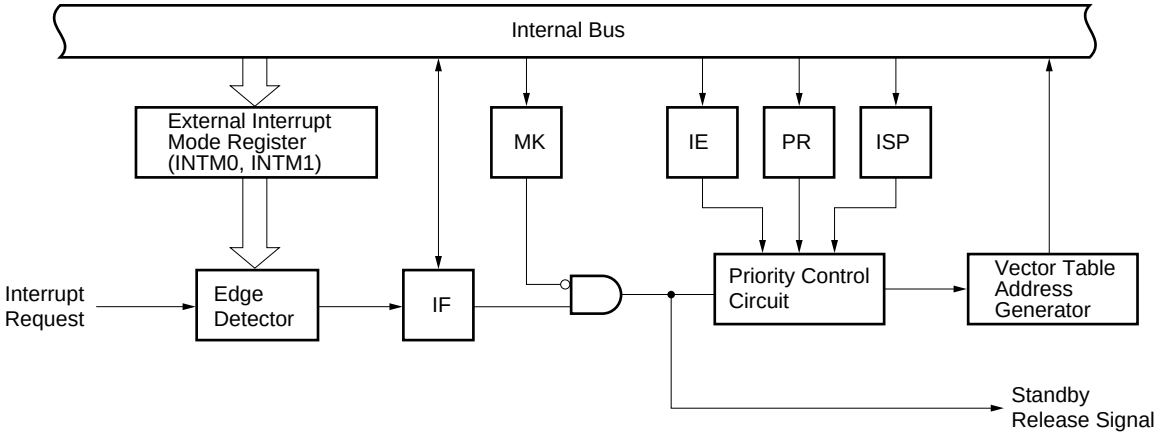
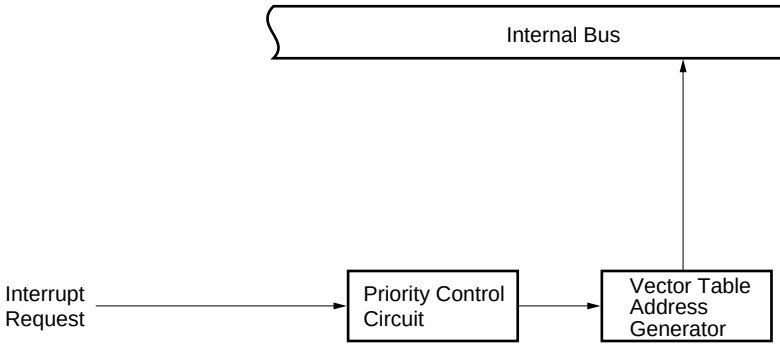


Figure 12-1. Basic Configuration of Interrupt Function (2/2)

(C) External maskable interrupt



(D) Software interrupt



Remark IF : Interrupt request flag
IE : Interrupt enable flag
ISP : Inservice priority flag
MK : Interrupt mask flag
PR : Priority specify flag

12.3 Interrupt Function Control Registers

The following five 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)
- Program status word (PSW)

Table 12-2 gives a listing of interrupt request flags, interrupt mask flags, and priority specify flags corresponding to interrupt request sources.

Table 12-2. 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
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTSER	SERIF	IF0H	SERMK	MK0H	SERPR	PR0H
INTSR/INTCSI2	SRIF		SRMK		SRPR	
INTST	STIF		STMK		STPR	
INTAD	ADIF	IF1L	ADMK	MK1L	ADPR	PR1L
INTTM5	TMIF5		TMMK5		TMPR5	
INTTM6	TMIF6		TMMK6		TMPR6	

(1) Interrupt request flag registers (IF0L, IF0H, IF1L)

The interrupt request flag is set to 1 when the corresponding interrupt request is generated or an instruction is executed. It is cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon application of $\overline{\text{RESET}}$ input.

IF0L, IF0H, and IF1L are set with a 1-bit or 8-bit memory manipulation instruction. If IF0L and IF0H are used as a 16-bit register IF0 use a 16-bit memory manipulation instruction for the setting.

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

Figure 12-2. Interrupt Request Flag Register Format

Symbol	7	6	5	④	③	②	1	⓪	Address	After Reset	R/W
IF0L	0	0	0	PIF3	PIF2	PIF1	0	TMIF4	FFE0H	00H	R/W
	7	6	5	④	③	②	1	0			
IF0H	0	0	0	STIF	SRIF	SERIF	0	0	FFE1H	00H	R/W
	7	6	5	④	③	②	1	0			
IF1L	0	0	0	TMIF6	TMIF5	ADIF	0	0	FFE2H	00H	R/W

××IF×	Interrupt Request Flag
0	No interrupt request signal
1	Interrupt request signal is generated; Interrupt request state

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 0 to the bits 1, 5 to 7 of IF0L and bits 0, 1, 5 to 7 of IF0H and IF1L.

(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 12-3. Interrupt Mask Flag Register Format

Symbol	7	6	5	④	③	②	1	⓪	Address	After Reset	R/W
MK0L	1	1	1	PMK3	PMK2	PMK	1	TMMK4	FFE4H	FFH	R/W
	7	6	5	④	③	②	1	0			
MK0H	1	1	1	STMK	SRMK	SERMK	1	1	FFE5H	FFH	R/W
	7	6	5	④	③	②	1	0			
MK1L	1	1	1	TMMK6	TMMK5	ADMK	1	1	FFE6H	FFH	R/W

×× MK ×	Interrupt Servicing Control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

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 1 to the bits 1, 5 to 7 of MK0L and bits 0, 1, 5 to 7 of MK0H and MK1L.

(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 12-4. Priority Specify Flag Register Format

Symbol	7	6	5	④	③	②	1	⓪	Address	After Reset	R/W
PR0L	1	1	1	PPR3	PPR2	PPR1	1	TMPR4	FFE8H	FFH	R/W
	7	6	5	④	③	②	1	0			
PR0H	1	1	1	STPR	SRPR	SERPR	1	1	FFE9H	FFH	R/W
	7	6	5	④	③	②	1	0			
PR1L	1	1	1	TMPR6	TMPR5	ADPR	1	1	FFEAH	FFH	R/W

××PR×	Priority Level Selection
0	High priority level
1	Low priority level

Cautions 1. If a watchdog timer is used in watchdog timer mode 1, set TMPR4 flag to 1.

2. Set 1 to the bits 1, 5 to 7 of PR0L and bits 0, 1, 5 to 7 of PR0H and PR1L.

(4) External interrupt mode register (INTM0, INTM1)

These registers set the valid edge for INTP1 to INTP3.

INTM0 and INTM1 are set by 8-bit memory manipulation instructions.

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

Figure 12-5. 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	0	0	0	0	FFECH	00H	R/W

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 0 to the bits 0 to 3.

Figure 12-6. External Interrupt Mode Register 1 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
INTM1	0	0	0	0	0	0	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

Caution Set 0 to the bits 2 to 7.

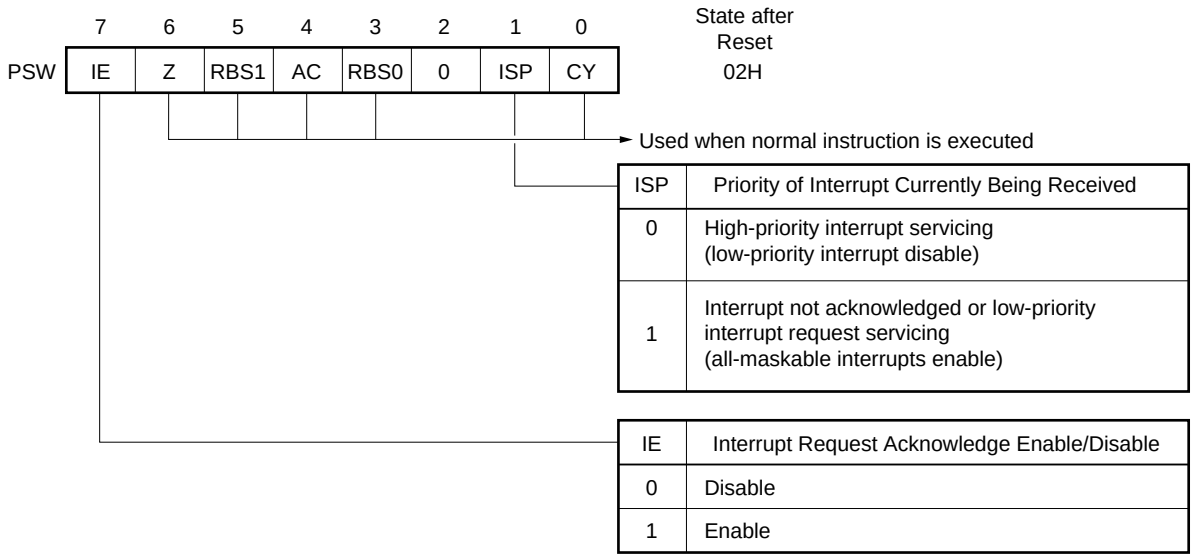
(5) Program status word (PSW)

The program status word is a register to hold the instruction execution result and the current status for interrupt request. The IE flag to set maskable interrupt enable/disable and the ISP flag to control multiple interrupt processing are mapped.

Besides 8-bit unit read/write, this register can carry out operations with a bit manipulation instruction and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged or when the BRK instruction is executed, the contents of PSW is automatically saved to the stack and the IE flag is reset to 0. If a maskable interrupt request is acknowledged the contents of the priority specify flag of the acknowledged interrupt are transferred to the ISP flag. The contents of PSW are also saved to the stack by the PUSH PSW instruction. It is reset from the stack with the RETI, RETB, and POP PSW instructions.

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

Figure 12-7. Program Status Word Configuration



ISP	Priority of Interrupt Currently Being Received
0	High-priority interrupt servicing (low-priority interrupt disable)
1	Interrupt not acknowledged or low-priority interrupt request servicing (all-maskable interrupts enable)

IE	Interrupt Request Acknowledge Enable/Disable
0	Disable
1	Enable

12.4 Interrupt Servicing Operations

12.4.1 Non-maskable interrupt request acknowledge operation

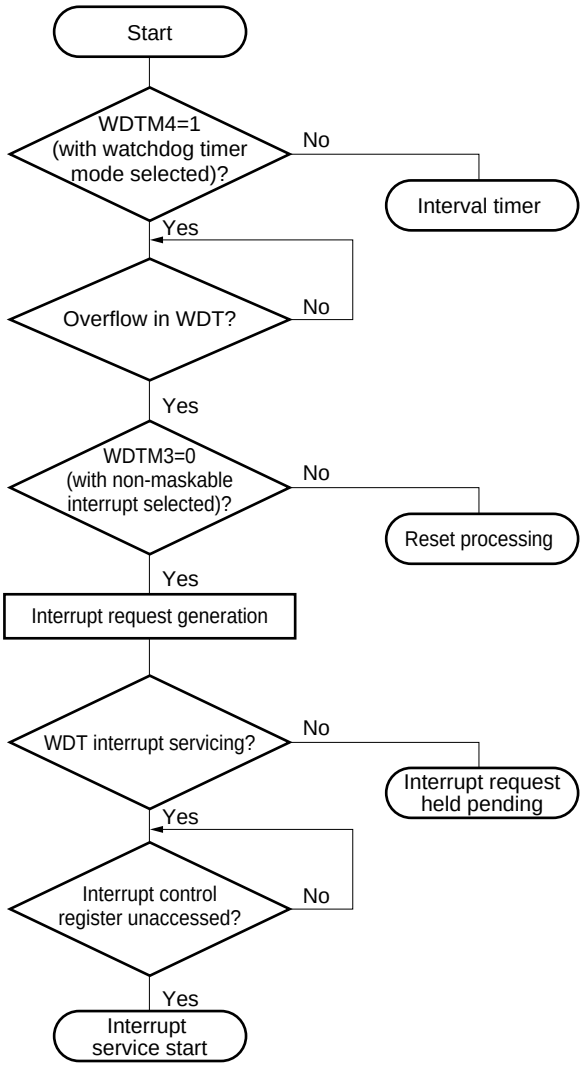
A non-maskable interrupt request is unconditionally acknowledged even if in an interrupt request acknowledge disable state. It does not undergo interrupt priority control and has highest priority over all other interrupts.

If a non-maskable interrupt request is acknowledged, the contents of program status word (PSW) and program counter (PC), in that order, are saved to the stack, the IE flag and ISP flag are reset (0), and the contents of the vector table are loaded in the PC and branched.

A new non-maskable interrupt request generated during execution of a non-maskable interrupt servicing program is acknowledged after the current execution of the non-maskable interrupt servicing program is terminated (following RETI instruction execution) and one main routine instruction is executed. If a new non-maskable interrupt request is generated twice or more during non-maskable interrupt service program execution, only one non-maskable interrupt request is acknowledged after termination of the non-maskable interrupt service program execution.

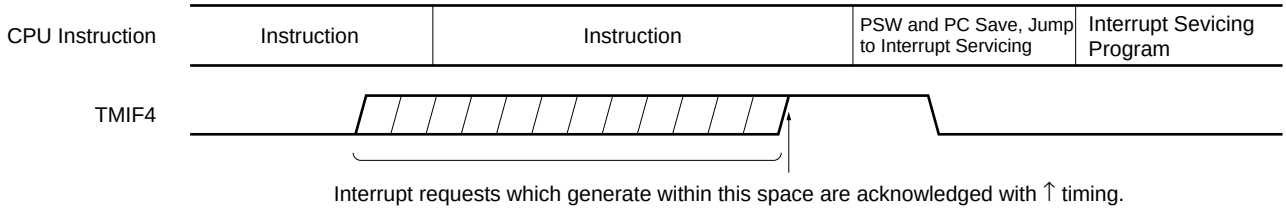
The flowchart showing the flow from non-maskable interrupt request generation to acknowledgment is shown in Figure 12-8, the non-maskable interrupt request acknowledge timing is shown in Figure 12-9, and acknowledge operation in the case where multiple non-maskable interrupt requests are generated, is shown in Figure 12-10.

Figure 12-8. Flowchart from Non-Maskable Interrupt Request Generation to Acknowledgment



WDTM : Watchdog timer mode register
WDT : Watchdog timer

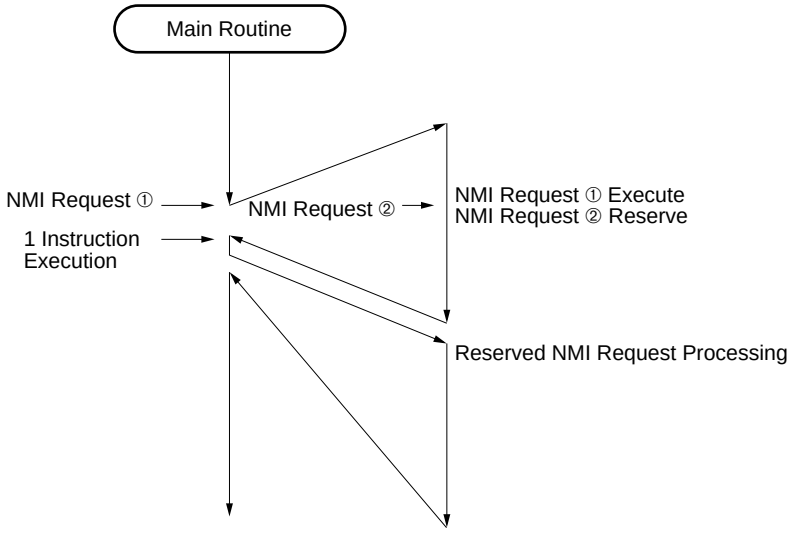
Figure 12-9. Non-Maskable Interrupt Request Acknowledge Timing



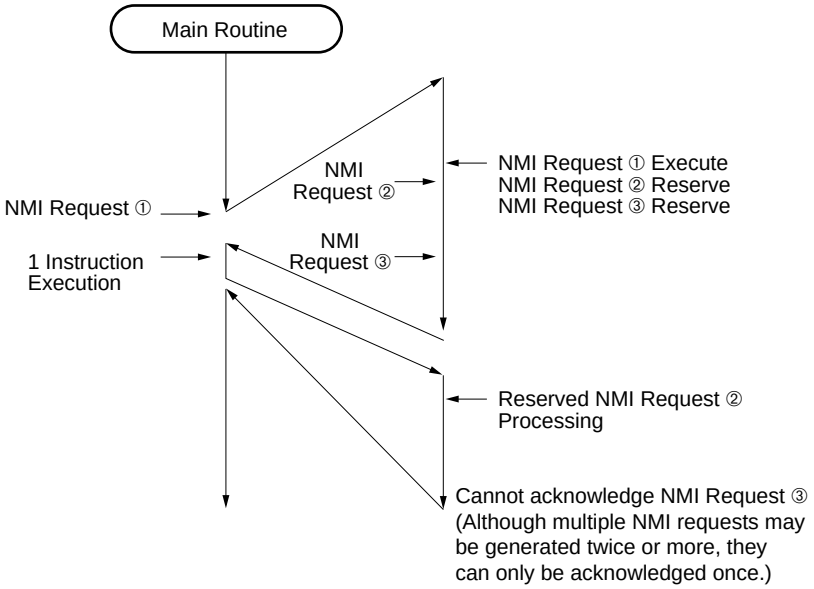
TMIF4 : Watchdog timer interrupt request flag

Figure 12-10. 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



12.4.2 Maskable interrupt request acknowledge operation

A maskable interrupt request becomes acknowledgeable when an interrupt request flag is set to 1 and the interrupt mask (MK) flag is cleared to 0. A vectored interrupt request is acknowledged in an interrupt enable state (with IE flag set to 1). However, a low-priority interrupt request is not acknowledged during high-priority interrupt service (with ISP flag reset to 0).

The waiting time from the point when a maskable interrupt request is generated until interrupt processing is executed is as shown in Table 12-3.

Please refer to Figures 12-12 and 12-13 concerning interrupt request acknowledgement timing.

Table 12-3. Times from Maskable Interrupt Request Generation to Interrupt Service

	Minimum Time	Maximum Time Note
When ××PR=0	7 clocks	32 clocks
When ××PR=1	8 clocks	33 clocks

Note If an interrupt request is generated just before a divide instruction, the wait time is maximized.

Remark 1 clock : $\frac{1}{f_{CPU}}$ (f_{CPU}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request specified for higher priority with the priority specify flag is acknowledged first. Also, when the same priority is specified with the priority specify flag, the interrupt request with the higher default priority is acknowledged first.

Any reserved interrupt requests are acknowledged when they become acknowledgeable.

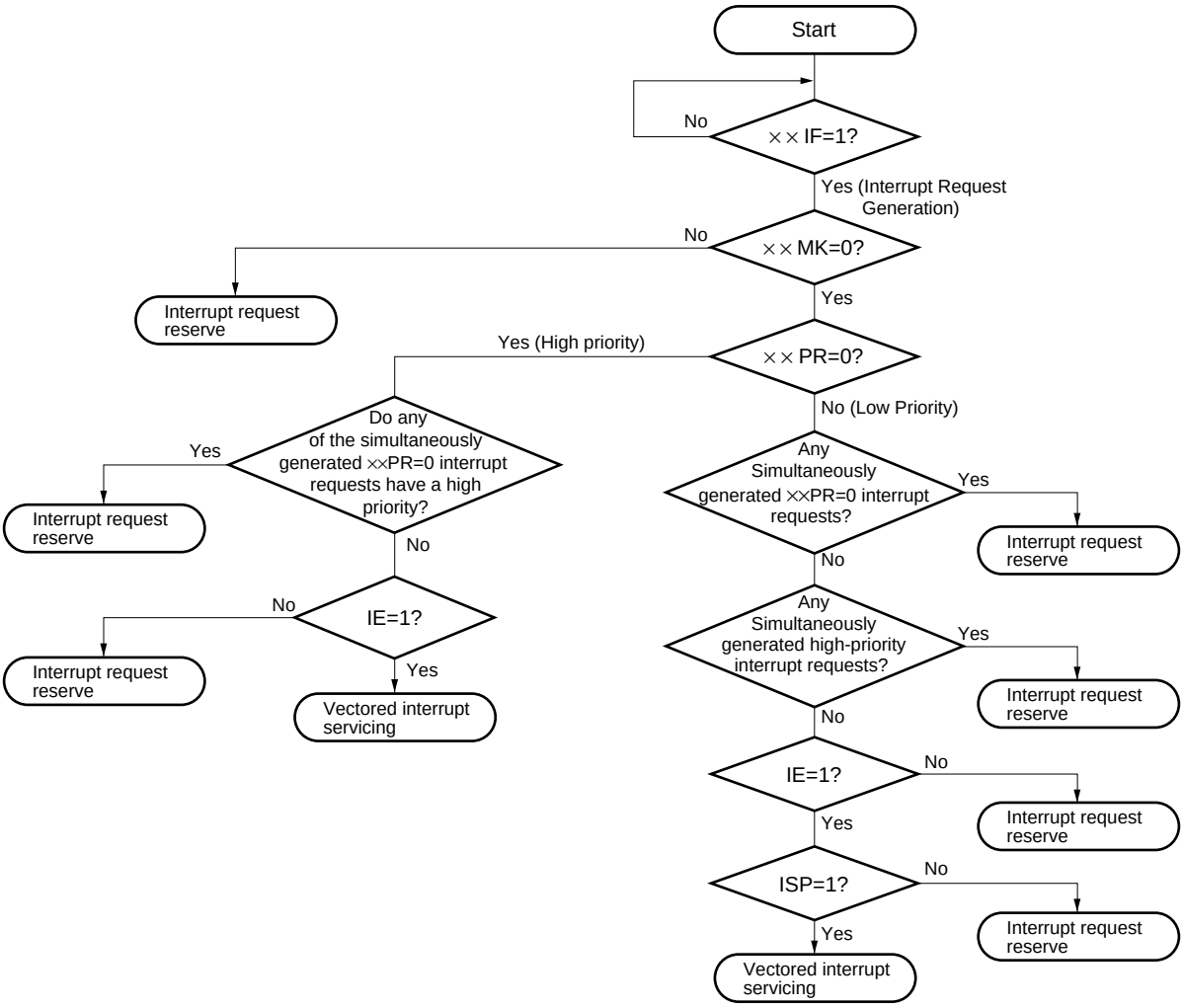
Figure 12-11 shows interrupt request acknowledge algorithms.

If a maskable interrupt request is acknowledged, the contents are saved to the stack in the order of first, program status word (PSW), then, program counter (PC), then the IE flag is reset (0) and the contents of the acknowledged interrupt request priority specification flag are transferred to the ISP flag.

Further, the data in the vector table which has been determined with each interrupt request, are loaded into the PC and branched.

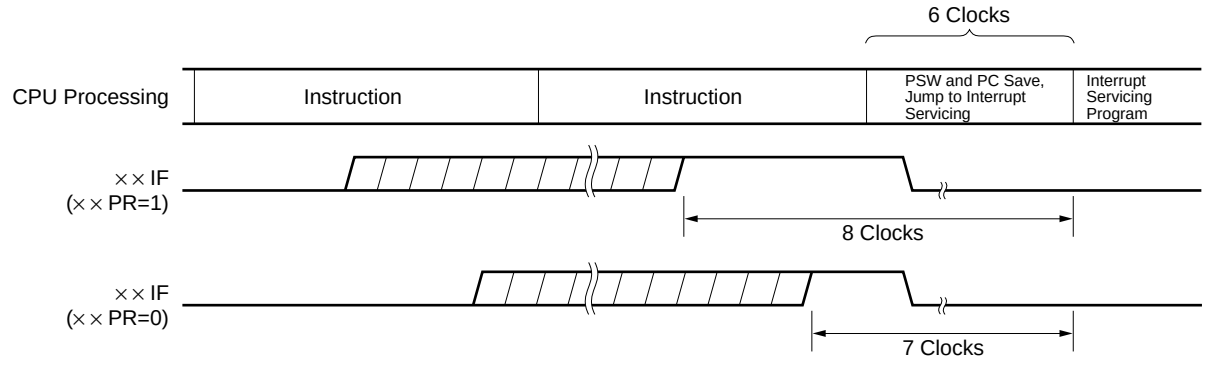
Return from the interrupt is possible with the RETI instruction.

Figure 12-11. Interrupt Request Acknowledge Processing Algorithm



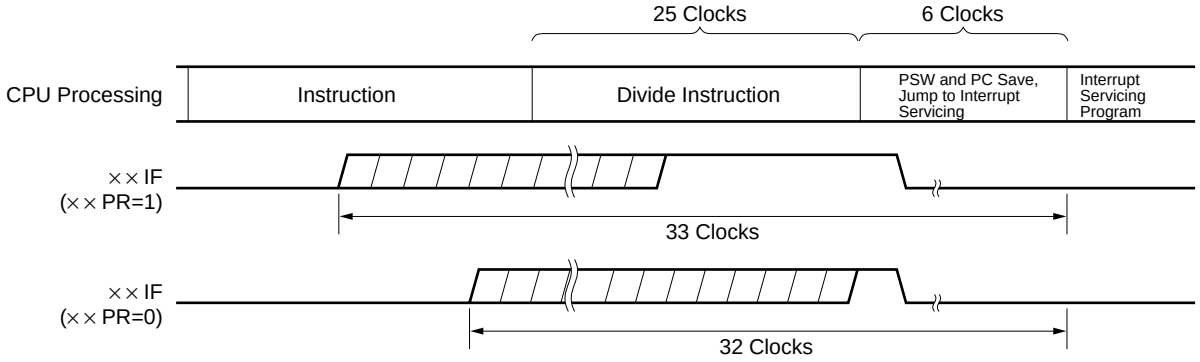
××IF : Interrupt request flag
××MK : Interrupt mask flag
××PR : Priority specify flag
IE : Flag which controls maskable interrupt request acknowledgment (1 = enable, 0 = disable)
ISP : Flag which indicates the priority of the interrupt currently being processed. (0 = high priority interrupt being processed, 1= interrupt request not acknowledged, or low priority interrupt being processed.)

Figure 12-12. Interrupt Request Acknowledge Timing (Minimum Time)



Remark 1 clock : $\frac{1}{f_{CPU}}$ (f_{CPU}: CPU clock)

Figure 12-13. Interrupt Request Acknowledge Timing (Maximum Time)



Remark 1 clock : $\frac{1}{f_{CPU}}$ (f_{CPU}: CPU clock)

12.4.3 Software interrupt request acknowledge operation

A software interrupt request is acknowledged by BRK instruction execution. Software interrupt cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved to the stack in the order of first, program status word (PSW), then the program counter (PC), then the IE flag is reset (0) and the contents of the vector tables (003EH, 003FH) are loaded into the PC and branched.

Return from the software interrupt is possible with the RETB instruction.

Caution Do not use the RETI instruction for returning from the software interrupt.

12.4.4 Multiple interrupt servicing

When another interrupt is acknowledged while an interrupt is being processed, this is called multiple interrupt.

Multiple interrupts are not generated unless interrupt request acknowledge is enabled (IE = 1) (non-maskable interrupts excepted). Also, at the point when an interrupt request is acknowledged, acknowledgment of other interrupt requests is disabled (IE = 0).Therefore, in order to enable multiple interrupts, it is necessary during interrupt processing to set the IE flag (1) using the IE instruction, and enable interrupts.

There are cases where multiple requests are not enabled even though interrupts are enabled. However, this is controlled by the priority of the interrupt. There are two interrupt priorities, the default priority and the programmable priority. Multiple interrupt control is handled by programmable priority control.

In interrupt enabled condition, when an interrupt request is generated which is the same level, or which has a higher priority than the interrupt currently being processed, it is acknowledged as a multiple interrupt. When an interrupt request is generated which has a lower priority than the interrupt currently being processed, it is not acknowledged as a multiple interrupt.

Interrupt requests which have not been permitted as multiple interrupts because of low priority or because of the interrupt being disabled, are reserved, and after the current interrupt processing has been completed, they are acknowledged after executing one of the main processing instructions.

Multiple interrupts are not permitted during non-maskable interrupt processing.

Multiple interrupt enabled interrupt requests are shown in Table 12-4, and an example of multiple interrupts is given in Figure 12-14 .

Table 12-4. Interrupt Request Enabled for Multiple Interrupt during Interrupt Servicing

Multiple Interrupt Request / Interrupt during processing		Non-maskable Interrupt Request	Maskable Interrupt Request			
			PR=0		PR=1	
			IE=1	IE=0	IE=1	IE=0
Non-maskable interrupt		D	D	D	D	D
Maskable interrupt	ISP=0	E	E	D	D	D
	ISP=1	E	E	D	E	D
Software interrupt		E	E	D	E	D

Remarks 1. E : Multiple interrupt enable

2. D : Multiple interrupt disable

3. ISP and IE are the flags contained in PSW

ISP=0 : An interrupt with higher priority is being serviced

ISP=1 : An interrupt request is not accepted or an interrupt with lower priority is being serviced

IE=0 : Interrupt request acknowledge is disabled

IE=1 : Interrupt request acknowledge is enabled

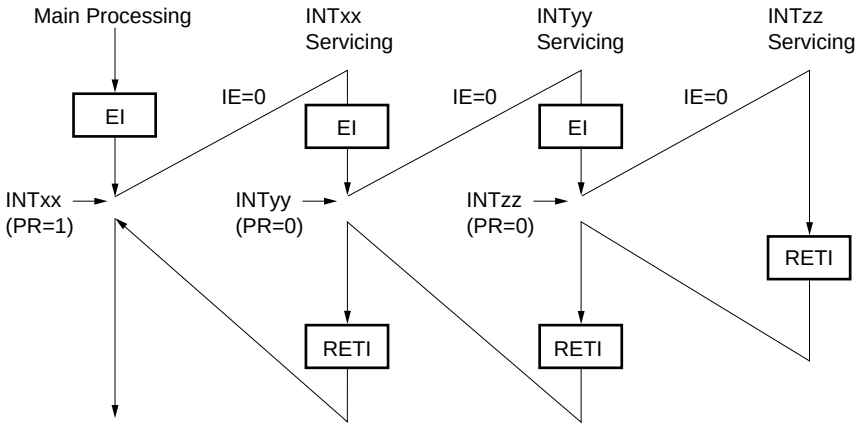
4. PR is a flag contained in PR0L, PR0H, and PR1L

PR=0 : Higher priority level

PR=1 : Lower priority level

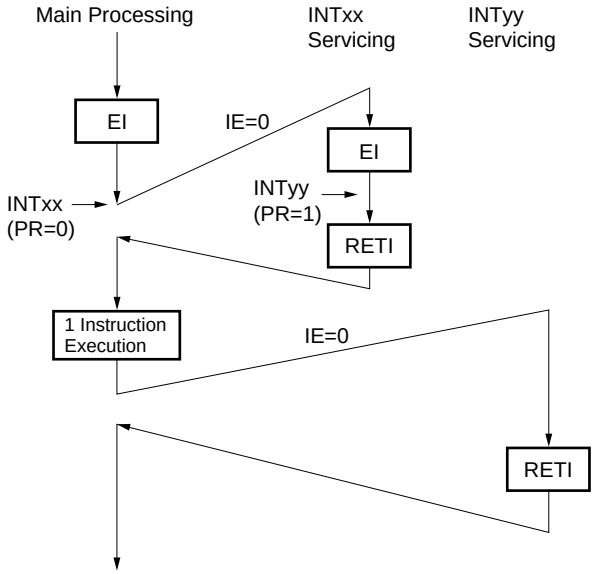
Figure 12-14. Multiple Interrupt Example (1/2)

Example 1. Example of when a multiple interrupt is generated twice.



Two interrupt requests, INTyy and INTzz, are acknowledged during processing of interrupt INTxx, and a multiple interrupt is generated. Before each interrupt request is acknowledged, the EI instruction is always executed and interrupt request acknowledgment enabled.

Example 2. Example of when a multiple interrupt is not generated because of priority control.

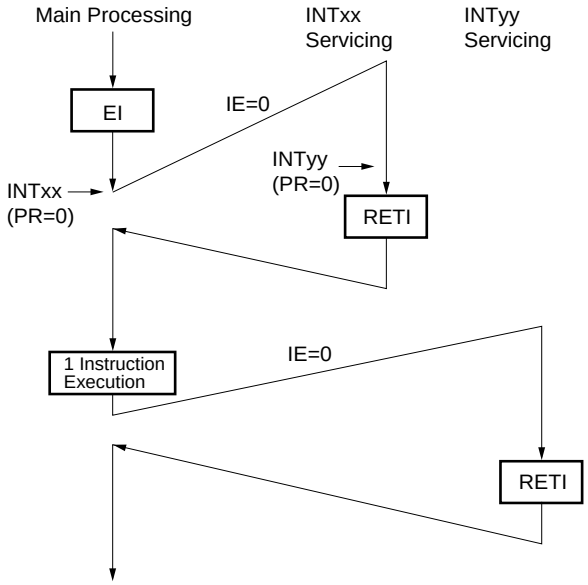


Interrupt request INTyy, which has been generated during processing of interrupt INTxx, and which has an interrupt priority that is lower than that of INTxx, is not acknowledged, and a multiple interrupt is not generated. Interrupt request INTyy is reserved and is acknowledged after execution of one main processing instructions.

PR = 0 : High priority level
PR = 1 : Low priority level
IE = 0 : Interrupt request acknowledge disabled

Figure 12-14 Multiple Interrupt Example (2/2)

Example 3. Example of when a multiple interrupt is not generated because interrupts are not enabled.



Because interrupts are not enabled (the EI instruction is not executed) during processing of interrupt INTxx , interrupt request INTyy is not acknowledged and a multiple interrupt is not generated. Interrupt request INTyy is reserved and acknowledged after one main processing instruction is implemented.

PR = 0 : High priority level
IE = 0 : Interrupt request acknowledge disabled.

12.4.5 Interrupt request reserve

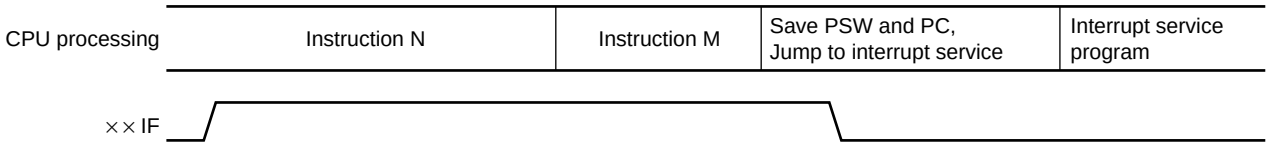
There are some instructions which, though an interrupt request may be generated while they are being executed, will reserve the acknowledgment of the request until after execution of the next instruction. These instructions (interrupt request reserve instructions) are shown below.

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW.bit, CY
- MOV1 CY, PSW.bit
- AND1 CY, PSW.bit
- OR1 CY, PSW.bit
- XOR1 CY, PSW.bit
- SET1 PSW.bit
- CLR1 PSW.bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW.bit, $addr16
- BF PSW.bit, $addr16
- BTCLR PSW.bit, $addr16
- EI
- DI
- Manipulate instructions for IF0L, IF0H, IF1L, MK0L, MK0H, MK1L, PR0L, PR0H, PR1L, INTM0, INTM1 registers

Caution The BRK instruction is not one of the above interrupt request reserve instructions. However in the case of software interrupts, which are activated by execution of the BRK instruction, the IE flag is cleared to 0. Therefore, even if a maskable interrupt request is generated during BRK instruction execution, the interrupt request will not be acknowledged. However, non-maskable interrupt requests will be acknowledged.

The interrupt request reserve timing is shown in Figure 12-15.

Figure 12-15. Interrupt Request Hold



Remarks 1. Instruction N: Instruction that holds interrupts requests

2. Instruction M: Instructions other than instruction N

3. The operation of ××IF (interrupt request) is not affected by ××PR (priority level) values.

CHAPTER 13 STANDBY FUNCTION

13.1 Standby Function and Configuration

13.1.1 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.

(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. When proceeding to the STOP mode, be sure to stop the peripheral hardware operation and execute the STOP instruction.

★ **2. 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.**

13.1.2 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.

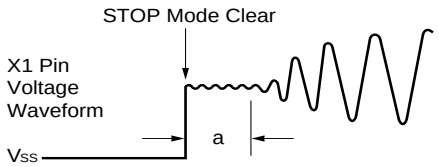
$\overline{\text{RESET}}$ input sets OSTS to 04H. However, it takes $2^{17}/f_X$, not $2^{18}/f_X$, until the STOP mode is cleared by $\overline{\text{RESET}}$ input.

Figure 13-1. 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 after STOP mode clear does not include the time (see "a" in the illustration below) from STOP mode clear to clock oscillation start, regardless of clearance by $\overline{\text{RESET}}$ input or by interrupt generation.



Remarks 1. f_{XX} : Main system clock frequency (f_X or $f_X/2$)
2. f_X : Main system clock oscillation frequency
3. MCS : Oscillation mode select register (OSMS) bit 0
4. Values in parentheses when operated at f_X = 5.0 MHz

13.2 Standby Function Operations

13.2.1 HALT mode

(1) HALT mode set and operating status

The HALT mode is set by executing the HALT instruction.

The operating status in the HALT mode is described below.

Table 13-1. HALT Mode Operating Status

Item	HALT Mode Operating Status
Clock generator	Can be oscillated. Supply to the CPU clock is stopped.
CPU	Operation stops.
Port	Status before HALT mode setting is held.
8-bit timer/event counter 5, 6	Operable.
Watchdog timer	
A/D converter	
Serial interface	
External interrupt request	

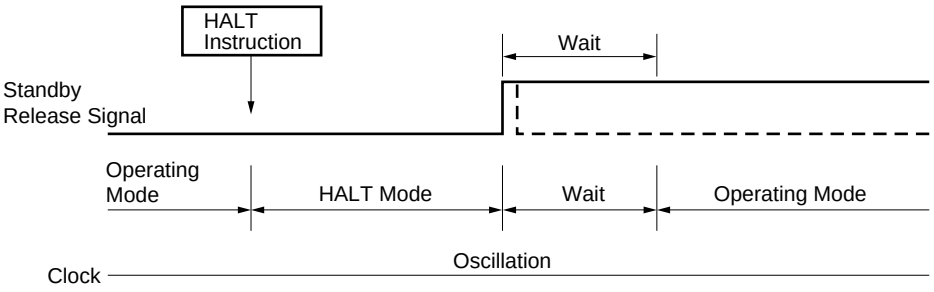
(2) HALT mode clear

The HALT mode can be cleared with the following three types of sources.

(a) Clear upon unmasked interrupt request

An unmasked interrupt request is used to clear the HALT mode. If interrupt acknowledge is enabled, vectored interrupt service is carried out. If disabled, the next address instruction is executed.

Figure 13-2. HALT Mode Clear upon Interrupt 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

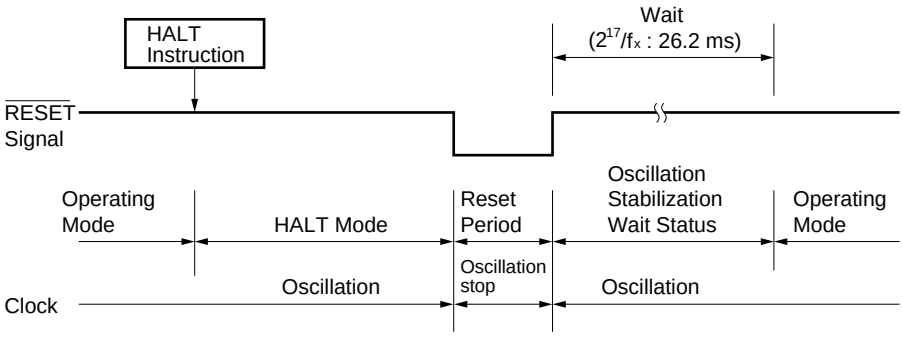
(b) Clear upon non-maskable interrupt request

The HALT mode is cleared and vectored interrupt service is carried out whether interrupt acknowledge is enabled or disabled.

(c) Clear upon $\overline{\text{RESET}}$ input

As is the case with normal reset operation, a program is executed after branch to the reset vector address.

Figure 13-3. HALT Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fx: main system clock oscillation frequency

2. Values in parentheses when operated at fx = 5.0 MHz

Table 13-2. Operation after HALT Mode Release

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt service execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt service execution
	1	×	×	×	HALT mode hold
Non-maskable interrupt request	–	–	×	×	Interrupt service execution
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

Remark ×: Don't care

13.2.2 STOP mode

(1) STOP mode set and operating status

The STOP mode is set by executing the STOP instruction.

★ **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 13-3. STOP Mode Operating Status

Item	STOP Mode Operating Status
Clock generator	Oscillation stops.
CPU	Operation stops.
Port	Status before STOP mode setting is held.
8-bit timer/event counter 5, 6	Operable when TI5 and TI6 are selected for the count clock.
Watchdog timer	Operation stops.
A/D converter	
Serial interface	Three-wire serial I/O is operable when the externally input clock is selected as the serial clock. UART operation stops.
External interrupt request	Operable.

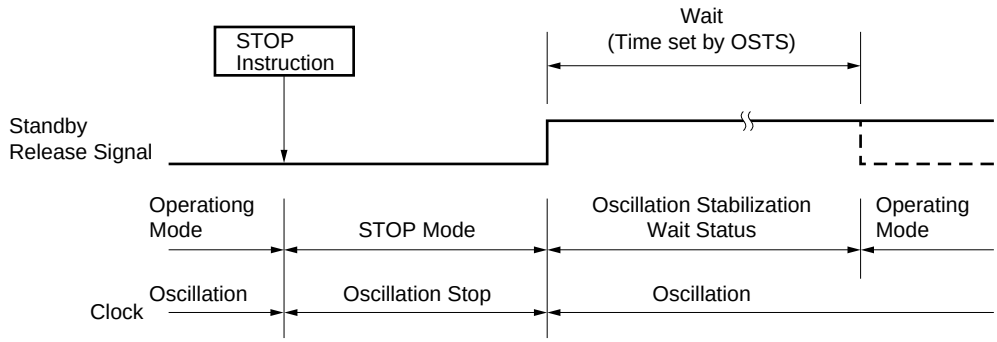
(2) STOP mode release

The STOP mode can be cleared with the following two types of sources.

(a) Release by unmasked interrupt request

An unmasked interrupt request is used to release the STOP mode. 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 13-4. STOP Mode Release by Interrupt Generation

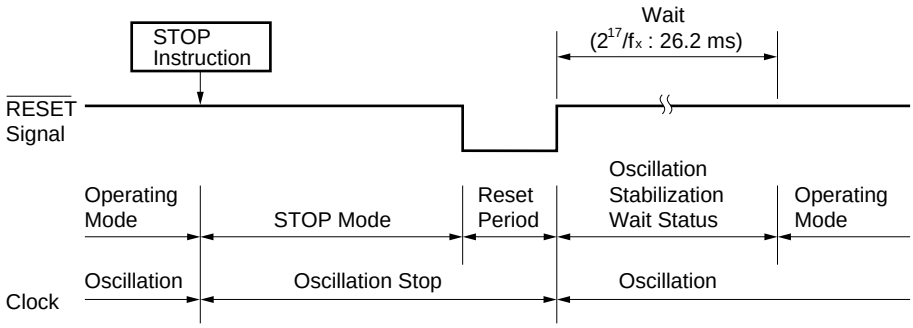


Remark The broken line indicates the case when the interrupt request which has cleared the standby status is acknowledged.

(b) Release by $\overline{\text{RESET}}$ input

The STOP mode is cleared and after the lapse of oscillation stabilization time, reset operation is carried out.

Figure 13-5. Release by STOP Mode $\overline{\text{RESET}}$ Input



Remarks 1. fx: main system clock oscillation frequency

2. Values in parentheses when operated at fx = 5.0 MHz

Table 13-4. Operation after STOP Mode Release

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt service execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt service execution
	1	×	×	×	STOP mode hold
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

Remark ×: Don't care

CHAPTER 14 RESET FUNCTION

14.1 Reset Function

The following two operations are available to generate the reset signal.

(1) External reset input with $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer overrun time detection

External reset and internal reset have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H by $\overline{\text{RESET}}$ input.

When a low level is input to the $\overline{\text{RESET}}$ pin or the watchdog timer overflows, a reset is applied and each hardware is set to the status as shown in Table 14-1. Each pin has high impedance during reset input or during oscillation stabilization time just after reset clear.

When a high level is input to the $\overline{\text{RESET}}$ input, the reset is cleared and program execution starts after the lapse of oscillation stabilization time ($2^{17}/f_X$). The reset applied by watchdog timer overflow is automatically cleared after a reset and program execution starts after the lapse of oscillation stabilization time ($2^{17}/f_X$) (see **Figure 14-2** to **14-4**).

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
2. Main system clock oscillation stops during reset input.
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 14-1. Block Diagram of Reset Function

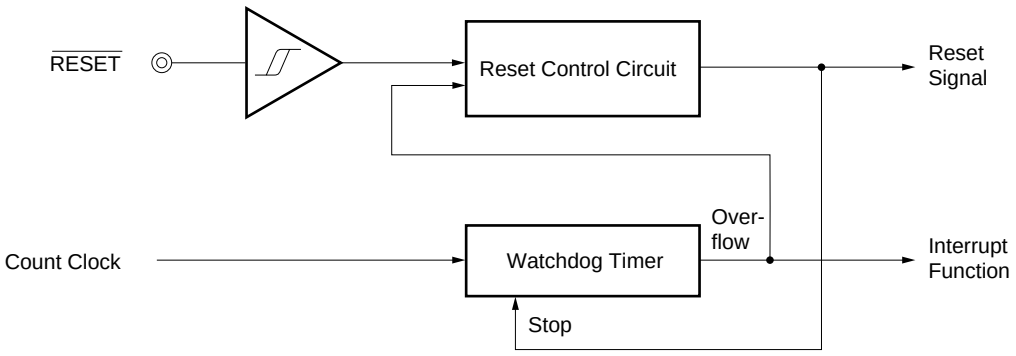


Figure 14-2. Timing of Reset Input by $\overline{\text{RESET}}$ Input

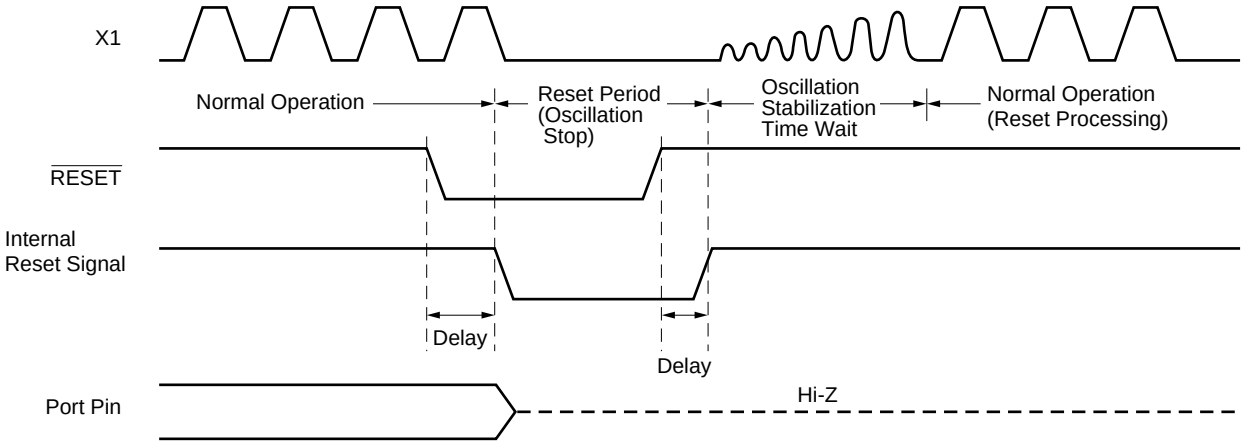


Figure 14-3. Timing of Reset due to Watchdog Timer Overflow

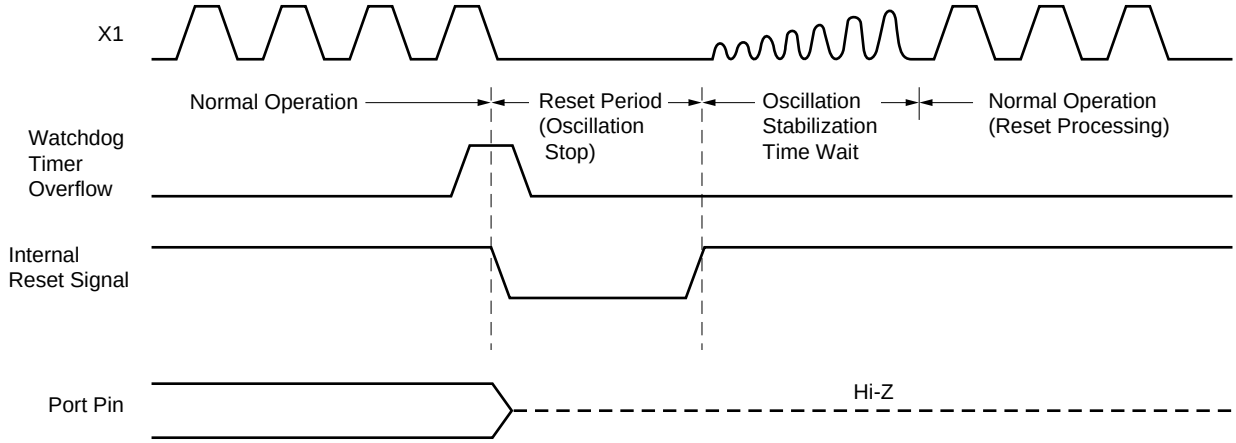


Figure 14-4. Timing of Reset Input in STOP Mode by $\overline{\text{RESET}}$ Input

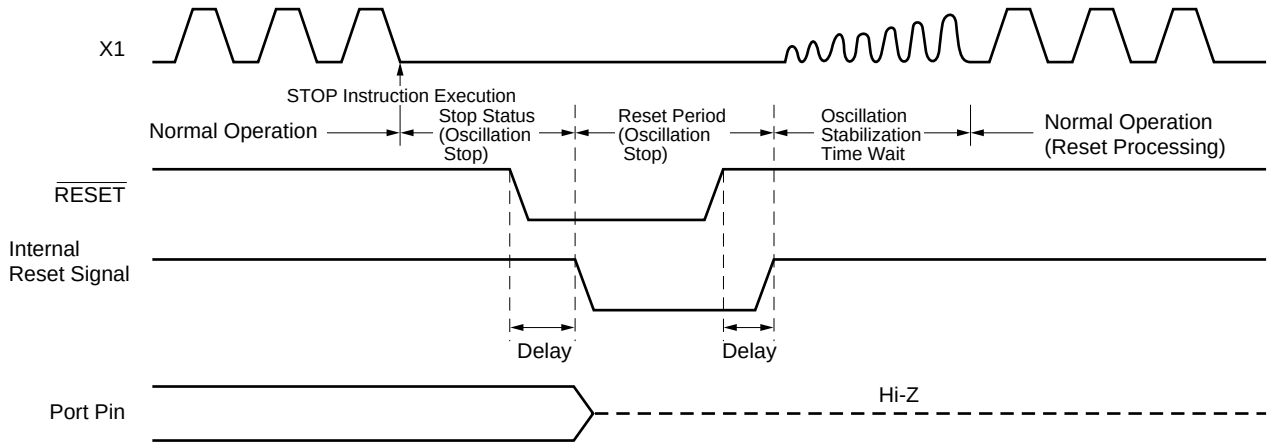


Table 14-1. Hardware Status after Reset (1/2)

Hardware		Status after Reset
Program counter (PC) Note1		The contents of reset vector tables (0000H and 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined Note2
	General register	Undefined Note2
Port (Output latch)	Port 0, Port 1, Port 3, Port 5, Port 7, Port 10 (P0, P1, P3, P5, P7, P10)	00H
Port mode register (PM0, PM1, PM3, PM5, PM7, PM10)		FFH
Pull-up resistor option register (PUOH, PUOL)		00H
Processor clock control register (PCC)		04H
Oscillation mode selection register (OSMS)		00H
Memory size switching register (IMS)		Note3
Oscillation stabilization time select register (OSTS)		04H
Timer clock selection register 0 (TCL0)		00H
8-bit timer/event counter 5 and 6	Timer register (TM5, TM6)	00H
	Compare registers (CR50, CR60)	00H
	Clock select register (TCL5, TCL6)	00H
	Mode control registers (TMC5, TMC6)	00H
Watchdog timer	Clock select register (TCL2)	00H
	Mode register (WDTM)	00H
Serial Interface	Mode register (CSIM2)	00H
	Asynchronous serial interface mode register (ASIM)	00H
	Asynchronous serial interface status register (ASIS)	00H
	Baud rate generator control register (BRGC)	00H
	Transmit shift register (TXS)	FFH
	Receive buffer register (RXB)	
A/D converter	Mode register (ADM)	01H
	Conversion result register (ADCR)	Undefined
	Input select register (ADIS)	00H

Table 14-1. Hardware Status after Reset (2/2)

Hardware		Status after Reset
Interrupt	Request flag register (IF0L, IF0H, IF1L)	00H
	Mask flag register (MK0L, MK0H, MK1L)	FFH
	Priority specify flag register (PR0L, PR0H, PR1L)	FFH
	External interrupt mode register (INTM0, INTM1)	00H

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. The post-reset status is held in the standby mode.

3. The values after reset depend on the product.
μPD78081 : 82H, μPD78082 : 64H, μPD78P083 : 46H

CHAPTER 15 μPD78P083

The μPD78P083 is a single-chip microcontroller with an on-chip one-time PROM or with an on-chip EPROM which has program write, erasure and rewrite capability.

Differences between the μPD78P083 and mask ROM versions are shown in Table 15-1.

Table 15-1. Differences between the μPD78P083 and Mask ROM Versions

Parameter	μPD78P083	Mask ROM Versions
Internal ROM type	One-time PROM/EPROM	Mask ROM
Internal ROM capacity	24 Kbytes	μPD78081 : 8 Kbytes μPD78082 : 16 Kbytes
Internal high-speed RAM capacity	512 bytes	μPD78081 : 256 bytes μPD78082 : 384 bytes
Internal ROM and internal high-speed RAM capacity change by internal memory size switching register (IMS)	Enable **Note**	Disable
IC pin	Not available	Available
V_{PP} pin	Available	Not available
Electrical specifications	Refer to a data sheet of each product	

Note The internal PROM becomes 24 Kbytes and the internal expansion RAM becomes 512 bytes by the $\overline{\text{RESET}}$ input.

★ **Caution Noise resistance and noise radiation differs between PROM versions and Mask ROM versions. If studying the replacement of PROM versions with mask ROM versions in the process of prototype to volume production, do a thorough evaluation with mask ROM versions with CS versions (not ES versions).**

15.1 Memory Size Switching Register

It is possible to specify the internal memory of the μPD78P083 by means of the memory size switching register (IMS). By setting the IMS, memory mapping can be made to match the memory mapping of the μPD78081 and 78082, which have different internal memory.

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

$\overline{\text{RESET}}$ input sets IMS to 46H.

★ **Figure 15-1. 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	46H	R/W

ROM3	ROM2	ROM1	ROM0	Internal ROM Capacity Selection
0	0	1	0	8 Kbytes
0	1	0	0	16 Kbytes
0	1	1	0	24 Kbytes
Other than above				Setting prohibited

RAM2	RAM1	RAM0	Internal High-Speed RAM Capacity Selection
0	1	0	512 bytes
0	1	1	384 bytes
1	0	0	256 bytes
Other than above			Setting prohibited

Caution If using mask ROM versions, do not specify any values in the IMS other than when resetting.

The IMS settings to give the same memory map as mask ROM versions are shown in Table 15-2.

Table 15-2. Examples of Memory Size Switching Register Settings

Relevant Mask ROM Version	IMS Setting
μPD78081	82H
μPD78082	64H

15.2 PROM Programming

The μPD78P083 incorporate a 24-Kbyte PROM as program memory, respectively. To write a program into the μPD78P083 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, see paragraph **(2) "PROM programming mode"** in section **1.5 Pin Configuration (Top View)**.

Caution Perform program writing only in the 0000H to 5FFFH address range (specify the last address as 5FFFH.)
The program cannot be correctly written by a PROM programmer which does not have a write address specification function.

15.2.1 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 μPD78P083 are set to the PROM programming mode. This is one of the operating modes shown in Table 15-3 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 15-3. PROM Programming Operating Modes

Pin / Operating mode	$\overline{RESET}$	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

Remark ×: 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 μPD78P083s 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 ($X \leq 10$).

(6) Byte write mode

A byte write is executed by applying a 0.1-ms program pulse (active-low) to the $\overline{PGM}$ pin while $\overline{CE}$=L and $\overline{OE}$=H. After this, program verification can be performed by setting $\overline{OE}$ to L.

If programming is not performed by one program pulse, repeated write and verify operations are executed X times ($X \leq 10$).

(7) Program verify mode

Setting $\overline{CE}$ to L, $\overline{PGM}$ to H, and $\overline{OE}$ to L sets the program verify mode.

After writing is performed, this mode should be used to check whether the data was written correctly.

(8) Program inhibit mode

The program inhibit mode is used when the $\overline{OE}$ pins, V_{PP} pins and pins D0 to D7 of multiple μPD78P083s 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.

15.2.2 PROM write procedure

Figure 15-2. Page Program Mode Flowchart

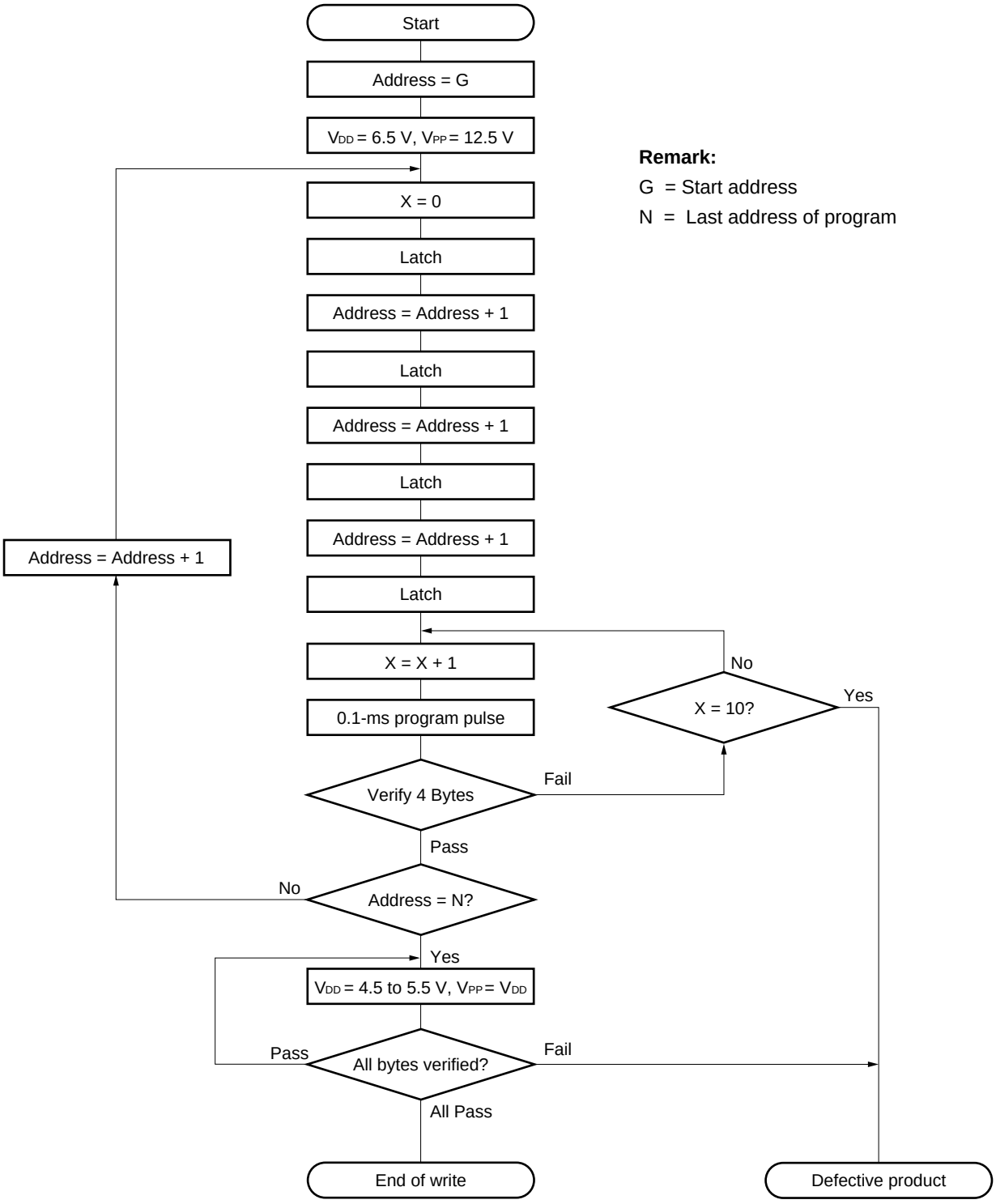


Figure 15-3. Page Program Mode Timing

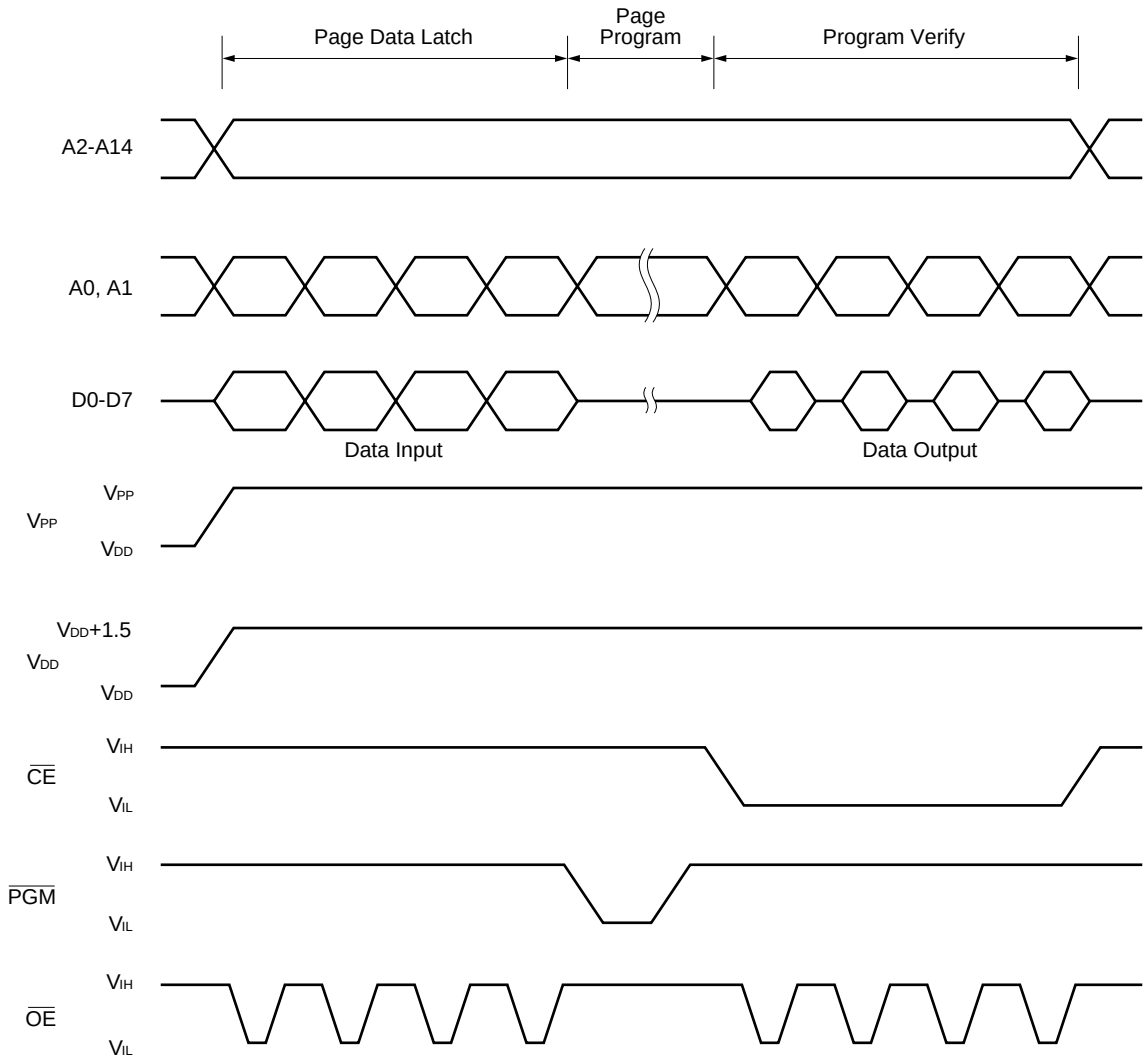


Figure 15-4. Byte Program Mode Flowchart

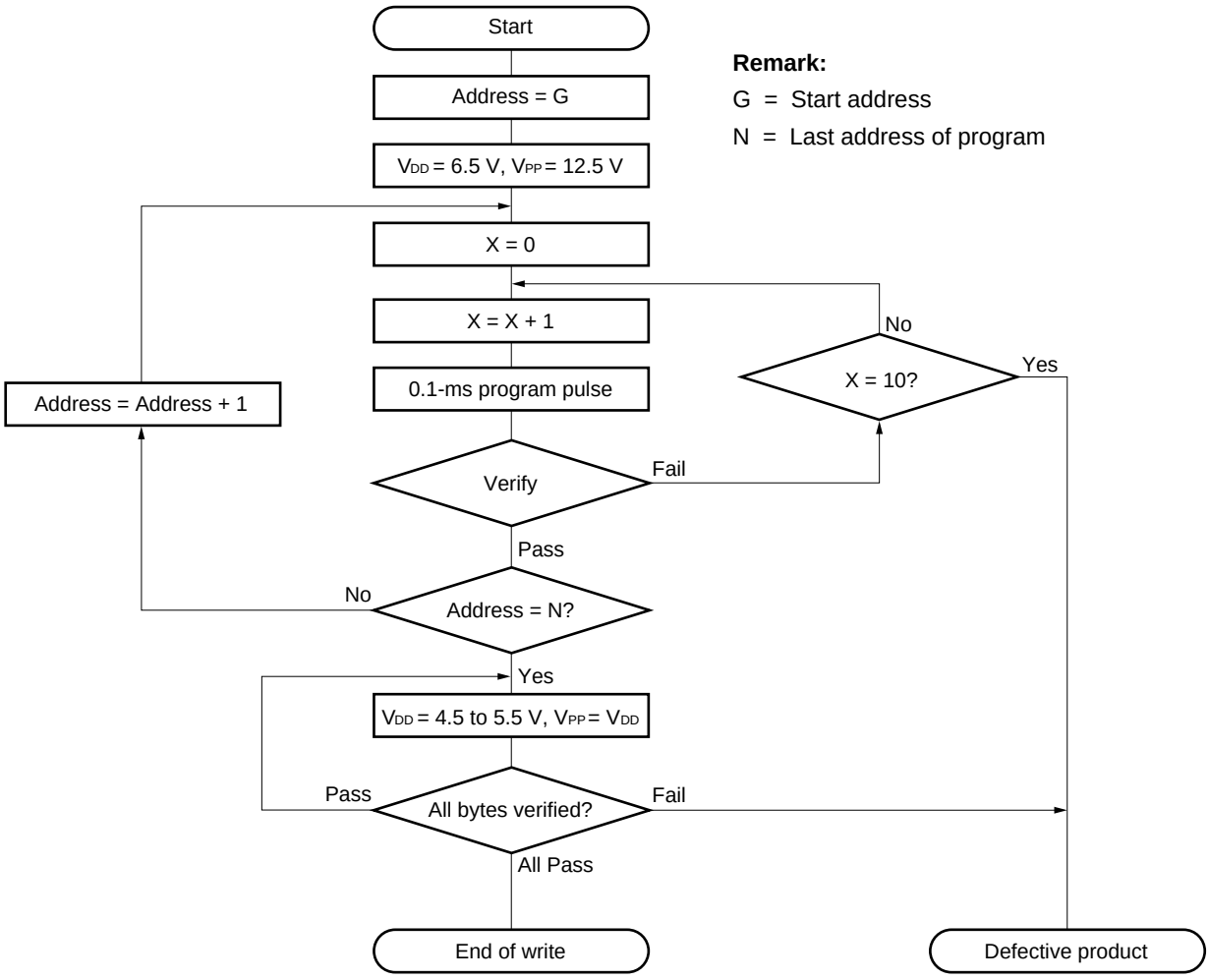
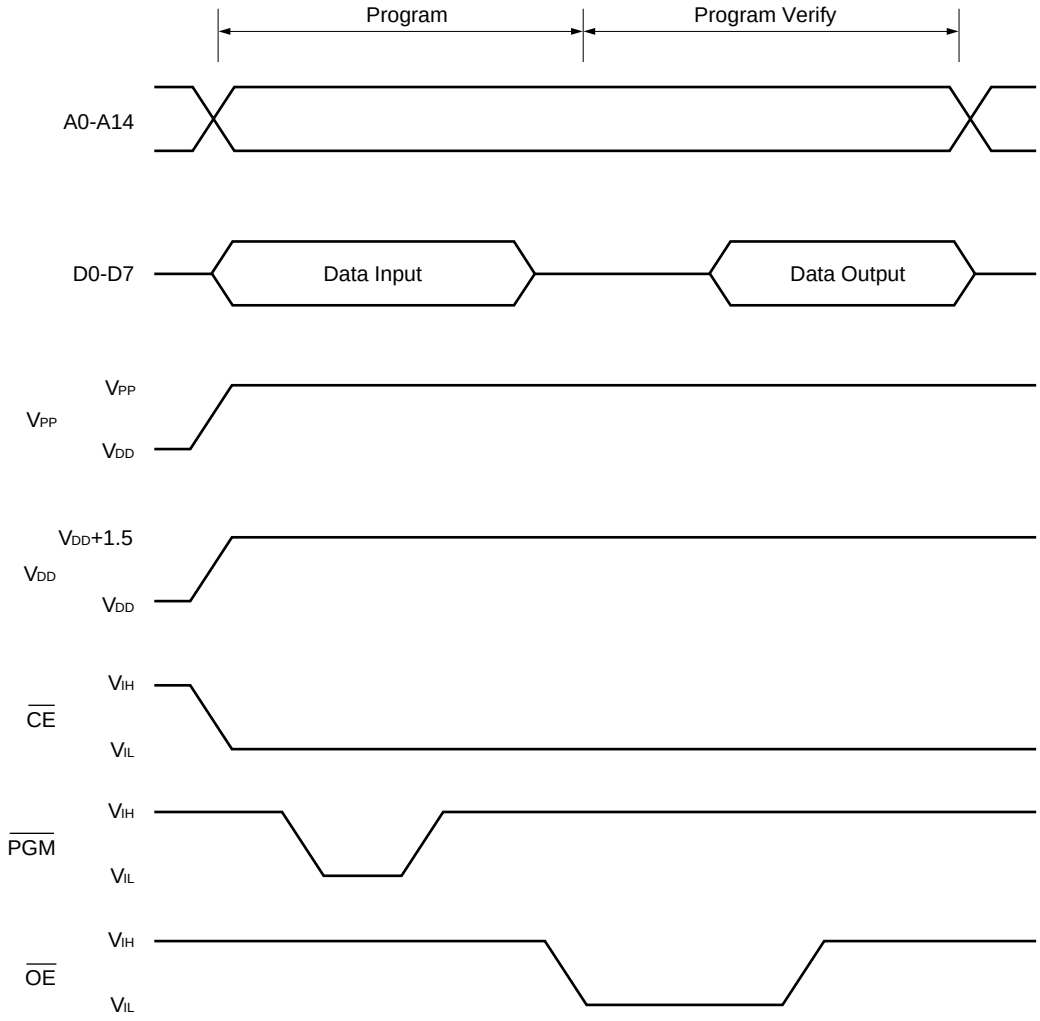


Figure 15-5. Byte Program Mode Timing



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.

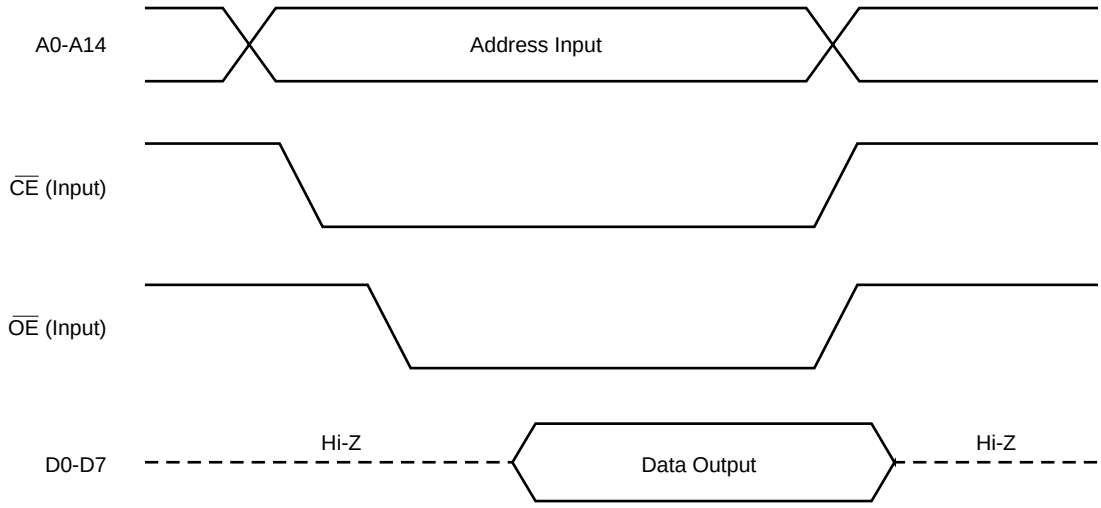
15.2.3 PROM reading procedure

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, **(2) "PROM programming mode"** in section **1.5 Pin Configuration (Top View)**.
(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 A14.
(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 15-6.

Figure 15-6. PROM Read Timing



15.3 Erasure Procedure (μPD78P083DU Only)

With the μPD78P083DU, it is possible to erase (or set all contents to FFH) the data contents written in the program memory, and rewrite the memory.

The data can be erased by exposing the window to light with a wavelength of approximately 400 nm or shorter. Typically, data is erased by 254-nm ultraviolet light rays. The minimum lighting level to completely erase the written data is shown below.

★ • UV intensity × exposure time: 30 W·s/cm^2 or more

★ • Exposure time: 40 minutes or more (using a 12 mW/cm^2 ultraviolet lamp. A longer exposure time may be required in case of deterioration of the ultraviolet lamp or dirt on the package window).

When erasing written data, remove any filter on the window and place the device within 2.5 cm of the lamp tube.

15.4 Opaque Film Masking the Window (μPD78P083DU Only)

To prevent unintentional erasure of the EPROM contents by light and to prevent internal circuits from mulfunction due to light coming in through the erasure window, mask the window with opaque film after writing the EPROM.

15.5 Screening of One-Time PROM Versions

One-time PROM versions (μPD78P083CU, 78P083GB-3B4, 78P083GB-3-MTX) 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. As regards the μPD78P083, preparations are underway. For details, please consult an NEC sales representative.

CHAPTER 16 INSTRUCTION SET

This chapter describes each instruction set of the μPD78083 subseries as list table. For details of its operation and operation code, refer to the separate document **"78K/0 series USER'S MANUAL—Instruction (IEU-1372)**."

16.1 Legends Used in Operation List

16.1.1 Operand identifiers and description methods

Operands are described in “Operand” column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for detail). When there are two or more description methods, select one of them. Alphabetic letters in capitals and symbols, #, !, $ and [ ] are key words and must be described as they are. Each symbol has the following meaning.

- # : Immediate data specification
- ! : Absolute address specification
- $: Relative address specification
- [ ] : Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, $, and [ ] symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for description.

Table 16-1. Operand Identifiers and Description Methods

Identifier	Description Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7),
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special-function register symbol**Note**
sfrp	Special-function register symbol (16-bit manipulatable register even addresses only)**Note**
saddr	FE20H-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 3-2 Special-Function Register List**.

16.1.2 Description of “operation” column

A	:	A register; 8-bit accumulator
X	:	X register
B	:	B register
C	:	C register
D	:	D register
E	:	E register
H	:	H register
L	:	L register
AX	:	AX register pair; 16-bit accumulator
BC	:	BC register pair
DE	:	DE register pair
HL	:	HL register pair
PC	:	Program counter
SP	:	Stack pointer
PSW	:	Program status word
CY	:	Carry flag
AC	:	Auxiliary carry flag
Z	:	Zero flag
RBS	:	Register bank select flag
IE	:	Interrupt request enable flag
NMIS	:	Non-maskable interrupt servicing flag
( )	:	Memory contents indicated by address or register contents in parentheses
$\times_H$, $\times_L$	:	Higher 8 bits and lower 8 bits of 16-bit register
$\wedge$	:	Logical product (AND)
$\vee$	:	Logical sum (OR)
$\veebar$	:	Exclusive logical sum (exclusive OR)
$\overline{\quad}$	:	Inverted data
addr16	:	16-bit immediate data or label
jdisp8	:	Signed 8-bit data (displacement value)

16.1.3 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

16.2 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 (f_{CPU}) selected by the 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 (f_{CPU}) selected by the 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 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 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, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$) × 1 time			×
	ROL	A, 1	1	2	–	(CY, $A_0 \leftarrow A_7$, $A_{m+1} \leftarrow A_m$) × 1 time			×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, $A_{m-1} \leftarrow A_m$) × 1 time			×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, $A_{m+1} \leftarrow A_m$) × 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 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 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) ← (PC + 3)H, (SP – 2) ← (PC + 3)L, PC ← addr16, SP ← SP – 2			
	CALLF	!addr11	2	5	–	(SP – 1) ← (PC + 2)H, (SP – 2) ← (PC + 2)L, PC15 – 11 ← 00001, PC10 – 0 ← addr11, SP ← SP – 2			
	CALLT	[addr5]	1	6	–	(SP – 1) ← (PC + 1)H, (SP – 2) ← (PC + 1)L, PCH ← (00000000, addr5 + 1), PCL ← (00000000, addr5), SP ← SP – 2			
	BRK		1	6	–	(SP – 1) ← PSW, (SP – 2) ← (PC + 1)H, (SP – 3) ← (PC + 1)L, PCH ← (003FH), PCL ← (003EH), SP ← SP – 3, IE ← 0			
	RET		1	6	–	PCH ← (SP + 1), PCL ← (SP), SP ← SP + 2			
	RETI		1	6	–	PCH ← (SP + 1), PCL ← (SP), PSW ← (SP + 2), SP ← SP + 3, NMIS ← 0	R	R	R
	RETB		1	6	–	PCH ← (SP + 1), PCL ← (SP), PSW ← (SP + 2), SP ← SP + 3	R	R	R
Stack manipulate	**PUSH**	PSW	1	2	–	(SP – 1) ← PSW, SP ← SP – 1			
		rp	1	4	–	(SP – 1) ← rpH, (SP – 2) ← rpL, SP ← SP – 2			
	POP	PSW	1	2	–	PSW ← (SP), SP ← SP + 1	R	R	R
		rp	1	4	–	rpH ← (SP + 1), rpL ← (SP), SP ← SP + 2			
	MOVW	SP, #word	4	–	10	SP ← word			
		SP, AX	2	–	8	SP ← AX			
		AX, SP	2	–	8	AX ← SP			
Unconditional branch	**BR**	!addr16	3	6	–	PC ← addr16			
		$addr16	2	6	–	PC ← PC + 2 + jdisp8			
		AX	2	8	–	PCH ← A, PCL ← X			
Conditional branch	**BC**	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 0			

Notes 1. When the internal high-speed RAM area is accessed or 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 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 PCC register.

16.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

First Operand \ Second Operand	#byte	A	r[Note]	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
★ B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

Note Except r = A

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

First Operand \ Second Operand	#word	AX	rp[Note]	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW[Note]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

First Operand \ Second Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call/instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

First Operand \ Second Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

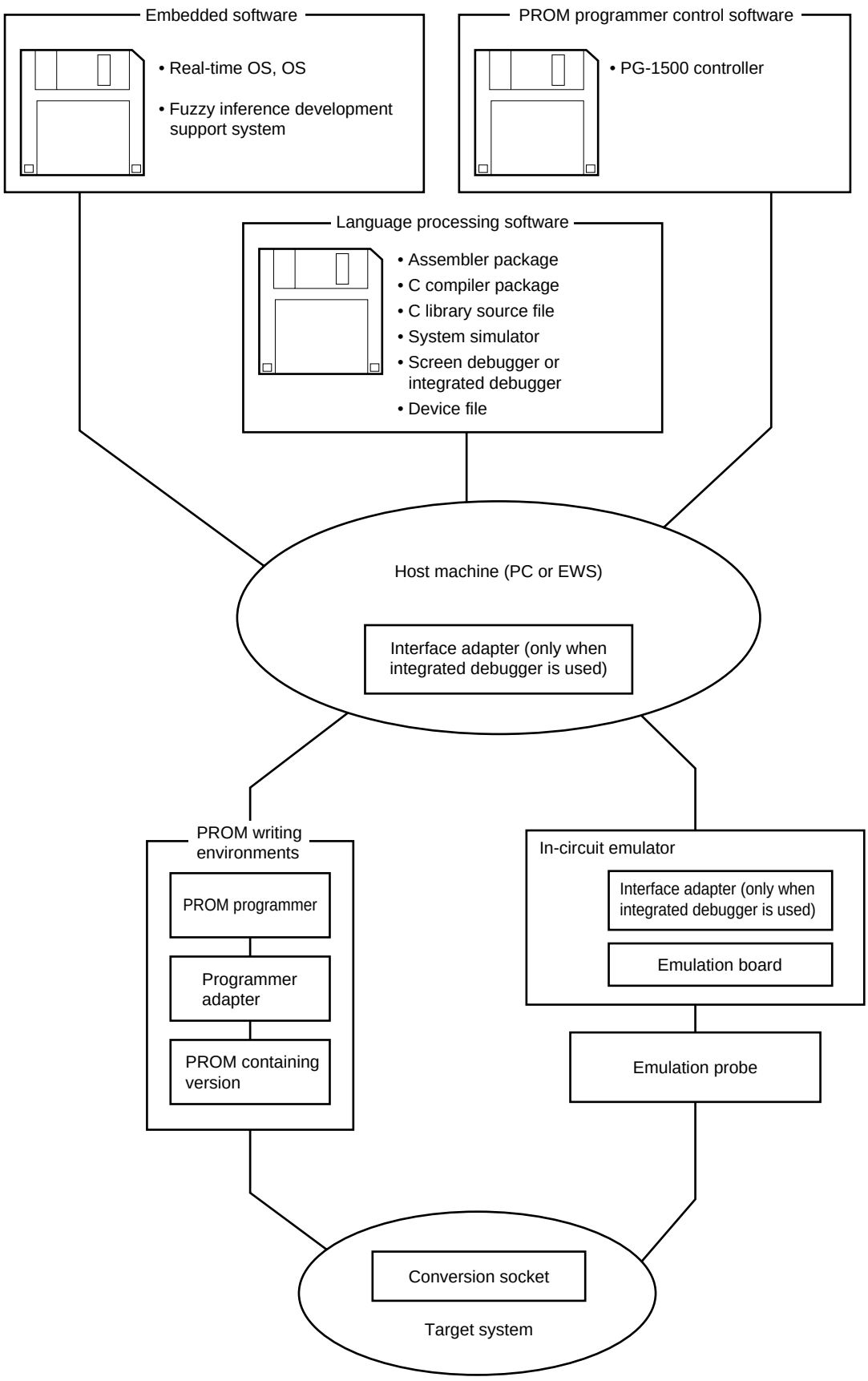
[MEMO]

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems which employ the μPD78083 subseries.

Figure A-1 shows the configuration of the development tools.

★ **Figure A-1. Development Tool Configuration**



A.1 Language Processing Software

RA78K/0 Assembler Package	This assembler converts a program written in mnemonics into an object code executable with a microprocontroller. Further, this assembler is provided with functions capable of automatically creating symbol tables and branch instruction optimization. Used in combination with optional device file (DF78083).
	Part Number: μS××××RA78K/0
CC78K/0 C Compiler Package	This compiler converts a program written in C Language into an object code executable with an microcontroller. Used in combination with optional assembler package (RA78K/0) and device file (DF78083).
	Part Number: μS××××CC78K/0
DF78083[Note] Device File	This is a file containing information inherent to the device. Used in combination with optional RA78K/0, CC78K/0, SM78K0, ID78K0, or SD78K/0.
	Part Number: μS××××DF78083
CC78K/0-L C Library Source File	This is a function source program configurating object library included in CC78K/0 C compiler. Necessary for changing object library included in CC78K/0 in according to customer's specifications.
	Part Number: μS××××CC78K0-L

Note The DF78083 can be used commonly with all the 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××××DF78083
μS××××CC78K0-L

××××	Host Machine	OS	Medium
5A13	PC-9800 Series	MS-DOS	3.5-inch 2HD
5A10		(Ver. 3.30 to 6.2[Note])	5-inch 2HD
7B13	IBM PC/AT and their	Refer to **A.4**	3.5-inch 2HC
7B10	compatible machines		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 audio tape (DAT)
3K15	SPARCstation™	SunOS™ (rel. 4.1.1)	Cartridge tape (QIC-24)
3M15	EWS-4800 series (RISC)	EWS-UX/V (rel. 4.0)	

Note The task swap function is not available with this software through the function is provided in MS-DOS version 5.0 or later.

A.2 PROM Programming Tools

A.2.1 Hardware

PG-1500 PROM programmer	This is a PROM programmer capable of programming the single-chip microcontroller with on-chip PROM by manipulating from the stand-alone or host machine through connection of the separately available programmer adapter and the attached board. It can also program separate PROM ICs with a capacity from 256 Kbits to 4 Mbits.
PA-78P083CU PA-78P083GB PROM programmer adapter	This is a PROM programmer adapter for the μPD78P083, and is used connected to the PG-1500. PA-78P083CU: 42-pin plastic shrink DIP (600 mil) 42-pin ceramic shrink DIP (with window) (600 mil) PA-78P083GB: 44-pin plastic QFP (10 × 10 mm)

A.2.2 Software

PG-1500 controller	This program controls the PG-1500 from the host machine through serial and/or parallel interface cable(s).
	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	Medium
5A13	PC-9800 Series	MS-DOS	3.5-inch 2HD
5A10		(Ver. 3.30 to 6.2[Note])	5-inch 2HD
7B13	IBM PC/AT and their	Refer to **A.4**	3.5-inch 2HC
7B10	compatible machines		5-inch 2HC

Note The task swap function is not available with this software through the function is provided in MS-DOS version 5.0 or later.

A.3 Debugging Tools

A.3.1 Hardware

★ IE-78000-R-A In-circuit emulator (supporting integrated debugger)	This in-circuit emulator helps users in debugging hardware and software of an application system that includes a 78K/0 series device. This in-circuit emulator supports integrated debugger (ID78K0). It is used with emulation probe and interface adapter that connects host machine.
★ IE-70000-98-IF-B Interface adapter	This is an adapter necessary when using the PC-9800 series (except notebook type) as a host machine for the IE-78000-R-A.
★ IE-70000-98N-IF Interface adapter	This is an adapter and cable necessary when using notebook type PC-9800 series as a host machine for the IE-78000-R-A.
★ IE-70000-PC-IF-B Interface adapter	This is an adapter necessary when using IBM PC/AT as a host machine for the IE-78000-R-A.
★ IE-78000-R-SV3 Interface adapter	This is an adapter and cable necessary when using EWS as a host machine for the IE-78000-R-A. As Ethernet™, 10Base-5 is supported. With other mode, commercially available conversion adapter is necessary.
IE-78000-R In-circuit emulator (supporting screen debugger)	This is in-circuit emulator that debugs hardware and software when application system using 78K/0 series is developed. It supports screen debugger (SD78K/0), and is used with emulation probe. This emulator is connected to host machine or PROM programmer for efficient debugging.
IE-78078-R-EM Emulation board	This board is used to emulate device-specific peripheral hardware (3.0 to 5.5 V), and is used with in-circuit emulator.
IEP-78083CU-R Emulation probe	This is a probe to connect an in-circuit emulator to target sytem. This probe is designed for 42-pin plastic shrink DIP (CU type) and 42-pin ceramic shrink DIP (DU type).
EP-78083GB-R Emulation probe	This is a probe to connect an in-circuit emulator to target system. This probe is designed for 44-pin plastic QFP (GB-3B4, GB-3BS-MTX types). This probe set includes a 44-pin conversion socket EV-9200G-44 for easier development of target systems
EV-9200G-44 Conversion socket	This adapter connects the EP-78083GB-R to the target system board designed for 44-pin plastic QFP (GB-3B4, GB-3BS-MTX types).

Remark EV-9200G-44s are sold in sets of five units.

A.3.2 Software (1/3)

★ SM78K0 System simulator	This simulator can debug target system at C source level or assembler level while simulating operation of target system on host machine. SM78K0 runs on Windows. By using SM78K0, logic and performance of application can be verified without in-circuit emulator independently of hardware development, so that development efficiency and software quality can be improved. This simulator is used with optional device file (DF78083).
	Part Number: μS××××SM78K0

Remark ×××× of the part number differs depending on the host machine and OS used. Refer to the table below.

μS××××SM78K0

××××	Host Machine	OS	Medium
AA13	PC-9800 Series	MS-DOS (Ver. 3.30 to 6.2[Note]) + Windows (Ver. 3.0 to 3.1)	3.5-inch 2HD
AB13	IBM PC/AT and their compatible machines (Windows in Japanese)	Refer to **A.4**	3.5-inch 2HC
BB13	IBM PC/AT and their compatible machines (Windows in English)		

Note The task swap function is not available with this software through the function is provided in MS-DOS version 5.0 or later.

A.3.2 Software (2/3)

★

ID78K0 Integrated debugger	This is control program that debugs 78K/0 series. This program employs Windows on personal computer and OSF/Motif™ on EWS as graphical user interface, and provides appearance and operability conforming to interface. In addition debugging functions supporting C language are reinforced. Trace result can be displayed at C level by using window integration function that associates source program, disassemble display, and memory display with trace result. Moreover, debugging efficiency of program using real-time OS can be enhanced by using function expansion modules such as task debugger and system performance analyzer. This program is used in combination with optional device file (DF78083).
	Part Number: μS××××ID78K0

Remark ×××× of the part number differs depending on the host machine and OS used. Refer to the table below.

μS××××ID78K0

××××	Host Machine	OS	Medium
AA13	PC-9800 Series	MS-DOS (Ver. 3.30 to 6.2[Note]) + Windows (Ver. 3.1)	3.5-inch 2HD
AB13	IBM PC/AT and their compatible machines (Windows in Japanese)	Refer to **A.4**	3.5-inch 2HC
BB13	IBM PC/AT and their compatible machines (Windows in English)		
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 The task swap function is not available with this software through the function is provided in MS-DOS version 5.0 or later.

A.3.2 Software (3/3)

SD78K/0 Screen debugger	This program controls IE-78000-R on host machine with IE-78000-R and host machine connected with serial interface (RS-232-C). It is used with optional device file (DF78083).
	Part Number: μS××××SD78K0
DF78083[Note] Device file	File containing device-specific information. It is used with optional RA78K/0, CC78K/0, SM78K0, ID78K0, or SD78K/0.
	Part Number: μS××××DF78083

Note This device file can be used for any of the RA78K/0, CC78K/0, SM78K0, ID78K0, SD78K/0 devices.

Remark ×××× of the part number differs depending on the host machine and OS used. Refer to the table below.

μS××××SD78K0
μS××××DF78083

××××	Host Machine	OS	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 and their compatible machines (Windows in Japanese)	Refer to **A.4**	3.5-inch 2HC
7B10			5-inch 2HC

Note The task swap function is not available with this software through the function is provided in MS-DOS version 5.0 or later.

★ A.4 OS for IBM PC

As the OS for IBM PC, the following is supported.

To run SM78K0, ID78K0, or FE9200 (refer to **B.2 Fuzzy Inference Development Support System**), Windows (Ver. 3.0 to Ver. 3.1) is necessary.

OS	Version
PC DOS	Ver. 5.02 to 6.3
	J6.1/V[Note] to J6.3/V[Note]
IBM DOS™	J5.02/V[Note]
MS-DOS	Ver. 5.0 to 6.22
	5.0/V[Note] to 6.2/V[Note]

Note Only English mode is supported.

Caution The task swap function is not available with this software through the function is provided in MS-DOS version 5.0 or later.

A.5 System-Upgrade Method from Other In-Circuit Emulators to 78K/0 Series In-Circuit Emulator

If 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 the 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 A-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 seires	IE-78320-R**Note**, IE-78327-R, IE-78330-R, IE-78350-R	

Note Maintenance product

★ **Table A-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 seires	IE-78320-R**Note 1**, IE-78327-R, IE-78330-R, IE-78350-R	
78K/0 series	IE-78000-R	—**Note 2**

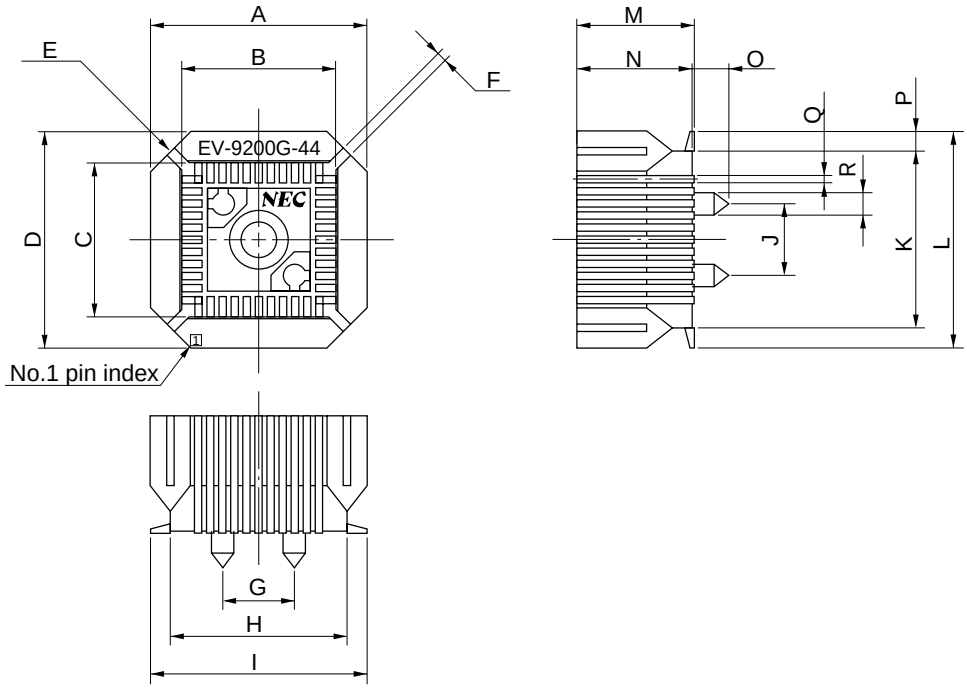
Notes 1. Maintenance product

2. Partial remodeling of the frame of the in-circuit emulator and replacement of the control/trace board with a supervisor board must be done by NEC.

Drawing and Footprint for Conversion Socket (EV-9200G-44)

Figure A-2. EV-9200G-44 Drawing (For Reference Only)

Based on EV-9200G-44
(1) Package drawing (in mm)

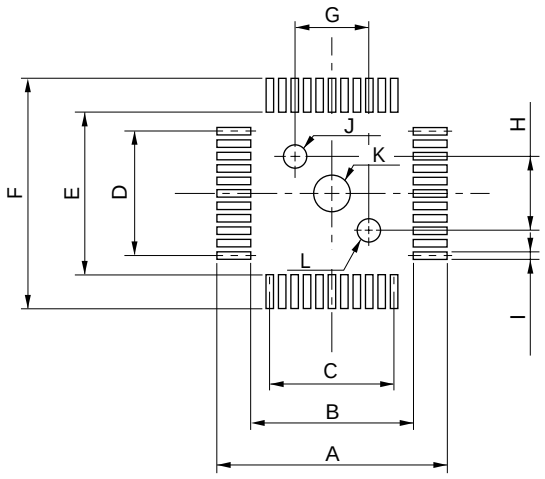


EV-9200G-44-G0E

ITEM	MILLIMETERS	INCHES
A	15.0	0.591
B	10.3	0.406
C	10.3	0.406
D	15.0	0.591
E	4-C 3.0	4-C 0.118
F	0.8	0.031
G	5.0	0.197
H	12.0	0.472
I	14.7	0.579
J	5.0	0.197
K	12.0	0.472
L	14.7	0.579
M	8.0	0.315
N	7.8	0.307
O	2.0	0.079
P	1.35	0.053
Q	0.35±0.1	$0.014^{+0.004}_{-0.005}$
R	ϕ1.5	ϕ0.059

Figure A-3. EV-9200G-44 Footprint (For Reference Only)

Based on EV-9200G-44
(2) Pad drawing (in mm)



EV-9200G-44-P1E

ITEM	MILLIMETERS	INCHES
A	15.7	0.618
B	11.0	0.433
C	$0.8\pm0.02\times10=8.0\pm0.05$	$0.031^{+0.002}_{-0.001}\times0.394=0.315^{+0.002}_{-0.002}$
D	$0.8\pm0.02\times10=8.0\pm0.05$	$0.031^{+0.002}_{-0.001}\times0.394=0.315^{+0.002}_{-0.002}$
E	11.0	0.433
F	15.7	0.618
G	5.00 ± 0.08	$0.197^{+0.003}_{-0.004}$
H	5.00 ± 0.08	$0.197^{+0.003}_{-0.004}$
I	0.5 ± 0.02	$0.02^{+0.001}_{-0.002}$
J	$\phi1.57\pm0.03$	$\phi0.062^{+0.001}_{-0.002}$
K	$\phi2.2\pm0.1$	$\phi0.087^{+0.004}_{-0.005}$
L	$\phi1.57\pm0.03$	$\phi0.062^{+0.001}_{-0.002}$

Caution Dimensions of mount pad for EV-9200 and that for target device (QFP) may be different in some parts. For the recommended mount pad dimensions for QFP, refer to "SEMICONDUCTOR DEVICE MOUNTING TECHNOLOGY MANUAL" (C10535E).

APPENDIX B EMBEDDED SOFTWARE

This section describes the embedded software which are provided for the μPD78083 subseries to allow users to develop and maintain the application program for these subseries.

★ B.1 Real-time OS

MX78K0 OS	μITRON-specification subset OS. Nucleus of MX78K0 is supplied. This OS performs task management, event management, and time management. It controls the task execution sequence for task management and selects the task to be executed next.
	Part Number: μS××××MX78K0-ΔΔΔ

Remark ×××× and ΔΔΔ of the part number differs depending on the host machine and operating system used. Refer to the table below.

μS××××MX78K0-ΔΔΔ

ΔΔΔ	Product outline	Note
001	Evaluation object	Use for experimental production.
××	Mass-production object	Use for mass production.
S01	Source program	Can be purchased only when object for mass production has been purchased.

××××	Host Machine	OS	Medium
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 and their compatible machines	Refer to **A.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 audio 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 through the function is provided in MS-DOS version 5.0 or later.

B.2 Fuzzy Inference Development Support System

FE9000/FE9200 Fuzzy Knowledge Data Creation Tool	This program supports input of fuzzy knowledge data (fuzzy rule and membership function), editing (edit), and evaluation (simulation) FE9200 operations on Windows.
	Part Number: μS××××FE9000 (PC-9800 series) μS××××FE9200 (IBM PC/AT and their compatible machines)
FT9080/FT9085 Translator	This program converts 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 and their compatible machines)
FI78K0 Fuzzy Inference Module	This program executes fuzzy inference by linking fuzzy knowledge data converted by translator.
	Part Number: μS××××FI78K0 (PC-9800 series, IBM PC/AT and their compatible machines)
FD78K0 Fuzzy Inference Debugger	This software supports evaluating and adjusting fuzzy knowledge data at hardware level by using in-circuit emulator.
	Part Number: μS××××FD78K0 (PC-9800 series, IBM PC/AT and their compatible machines)

Remark ×××× of the part number differs depending on the host machine and operating system used. Refer to the table below.

μS××××FE9000
μS××××FT9080
μS××××FI78K0
μS××××FD78K0

××××	Host Machine	OS	Medium
5A13	PC-9800 Series	MS-DOS	3.5-inch 2HD
5A10		(Ver. 3.30 to 6.2[Note2])	5-inch 2HD

Note The task swap function is not available with this software through the function is provided in MS-DOS version 5.0 or later.

μS××××FE9200
μS××××FT9085
μS××××FI78K0
μS××××FD78K0

××××	Host Machine	OS	Medium
7B13	IBM PC/AT and their compatible machines	Refer to **A.4**	3.5-inch 2HC
7B10			5-inch 2HC

[MEMO]

APPENDIX C REGISTER INDEX

C.1 Register Index

APPENDIX D REVISION HISTORY

Major revisions by edition and revised chapters are shown below.

Edition	Major revisions from previous version	Revised Chapter
2nd	The following products have been already developed μPD78081CU-×××, 78081GB-×××-3B4, 78082CU-×××, 78082GB-×××-3B4, 78P083CU, 78P083DU, 78P083GB-3B4	Throughout
	The following products have been added μPD78081GB-×××-3BS-MTX, 78082GB-×××-3B4-MTX, 780P083GB-3BS-MTX, 78081GB(A)-×××-3B4,78082GB(A)-×××-3B4, 78P083CU(A), 78P083GB(A)-3B4, 78P083GB(A)-3BS-MTX, 78081GB(A2)-×××-3B4	
	Power voltage changed to V_{DD} = 1.8 to 5.5V.	
	1.6 78K/0 Series Development has been changed.	**CHAPTER 1 OUTLINE**
	1.9 Differences between the μPD78081, 78082, and 78P083, the μPD78081(A), 78082(A), and 78P083(A), and the μPD78081(A2) has been added.	
	Cautions regarding the use of functions in common with **2.2.5 (2) (d) ASCK** has been added.	**CHAPTER 2 PIN FUNCTION**
	Cautions concerning the Write to OSMS Command has been added to **5.3 (2) Oscillation mode select register (OSMS).**	**CHAPTER 5 CLOCK GENERATOR**
	Cautions concerning external clock input in **5.4.1 Main system clock oscillation** has been changed.	
	Figure 7-3. Watchdog Timer Mode Register Format, notes and cautions have been added.	**CHAPTER 7 WATCHDOG TIMER**
	Description of **7.4.2 Interval timer operation** has been changed.	
	Cautions with regard to rewriting TCL0 to other than same data has been added to **8.3 (1) Timer clock select register 0 (TCL0)**.	**CHAPTER 8 CLOCK OUTPUT CONTROL CIRCUIT**
	The HSC bit has been added to the A/D Converter Mode Register in **Figure10-1. A/D Converter Block Diagram.**	**CHAPTER 10 A/D CONVERTER**
	10.3 (1) A/D converter mode register (ADM), 13.1.1 Standby function, and **Cautions** have been added.	
	Figure 11-1. Serial Interface Channel 2 Block Diagram has been corrected.	**CHAPTER 11 SERIAL INTERFACE CHANNEL 2**
	11.3 (4) (a), 11.4.2 (1) (d) (i) Generation of baud rate transmit/ receive clock by means of main system clock have been added. 76800 bps has been added to baud rate generated from the main system clock.	
	Figure 11-10. Receive Error Timing has been corrected.	
	11.4.3 (c) Baud rate generator control register (BRGC) has been added.	
	15.1 Memory Size Switching Register has been changed from W to R/W.	**CHAPTER 15 μPD78P083**
	Items and cautions have been added to **Table 15-1. Differences between the μPD78P083 and Mask ROM Versions.**	
	A description of the QTOP icon has been added to **15.5 Screening of One-Time PROM Versions.**	

Edition	Major revisions from previous version	Revised Chapter
2nd	**Figure A-1. Development Tool Configuration** has been changed.	**APPENDIX A DEVELOPMENT TOOLS**
	APPENDIX A DEVELOPMENT TOOLS The following Development Tools have been added: IE-78000-R-A, IE-70000-98-IF-B, IE-70000-98-N-IF, IE-70000-PC-IF-B, IE-78000-R-SV3, SM78K0, ID78K0	
	A.4 OS for IBM PC has been added.	
	Table A-2. System-Up Method from Other In-Circuit Emulator to IE-78000-R-A has been added.	
	B.1 Real-time OS has been added.	**APPENDIX B EMBEDDED SOFTWARE**

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	❑	❑	❑	❑