To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

RENESAS

Preliminary User's Manual

μPD780852 Subseries

8-Bit Single-Chip Microcontrollers

μPD780851(A)
μPD780852(A)
μPD78F0852

Document No. U14581EJ3V0UM00 (3rd edition)
Date Published October 2000 J CP(K)

[MEMO]

SUMMARY OF CONTENTS

Major Revisions in This Edition

Page	Description
p.29	Changing **1.5 Pin Configuration (Top View)**
p.34	Changing description of supply voltage in **1.8 Outline of Function**
p.107	Changing **6.4 (4) Port mode register 4 (PM4)**
p.111	Changing **Figure 6-11 Capture Register Data Retention Timing**
p.211	Adding **Caution 3** to **Figure 16-4 LCD Display Control Register (LCDC) Format**
p.278	Adding **Note** to **Table 22-3 Transmission Method List**

The mark ★ shows major revised points.

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 : μPD78F0852GC-8BT

The customer must judge the need for license : μPD780851GC(A)-×××-8BT, 780852GC(A)-×××-8BT

- **The information in this document is current as of October, 2000. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC's data sheets or data books, etc., for the most up-to-date specifications of NEC semiconductor products. Not all products and/or types are available in every country. Please check with an NEC sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without prior written consent of NEC. NEC assumes no responsibility for any errors that may appear in this document.
- NEC does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC semiconductor products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of customer's equipment shall be done under the full responsibility of customer. NEC assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC endeavours to enhance the quality, reliability and safety of NEC semiconductor products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC semiconductor products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment, and anti-failure features.
- NEC semiconductor products are classified into the following three quality grades:
  "Standard", "Special" and "Specific". The "Specific" quality grade applies only to semiconductor products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of a semiconductor product depend on its quality grade, as indicated below. Customers must check the quality grade of each semiconductor product before using it in a particular application.
  "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots
  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support)
  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.
  The quality grade of NEC semiconductor products is "Standard" unless otherwise expressly specified in NEC's data sheets or data books, etc. If customers wish to use NEC semiconductor products in applications not intended by NEC, they must contact an NEC sales representative in advance to determine NEC's willingness to support a given application.
  (Note)
  (1) "NEC" as used in this statement means NEC Corporation and also includes its majority-owned subsidiaries.
  (2) "NEC semiconductor products" means any semiconductor product developed or manufactured by or for NEC (as defined above).

M8E 00. 4

Regional Information

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

NEC do Brasil S.A.
Electron Devices Division
Guarulhos-SP Brasil
Tel: 55-11-6462-6810
Fax: 55-11-6462-6829

J00.7

[MEMO]

INTRODUCTION

Readers

This manual has been prepared for user engineers who want to understand the functions of the μPD780852 Subseries and design and develop its application systems and programs.

μPD780852 Subseries: μPD780851(A), 780852(A), 78F0852

Purpose

This manual is designed to help users understand the following functions using the organization below.

Organization

The μPD780852 Subseries manual is separated into two parts: this manual and the instruction edition (common to the 78K/0 Series).

μPD780852 Subseries User's Manual (This Manual)	**78K/0 Series User's Manual Instructions**
• Pin functions • Internal block functions • Interrupt • Other on-chip peripheral functions	• CPU functions • Instruction set • Explanation of each instruction

How to Read This Manual

This manual assumes general knowledge of electric engineering, logic circuits, and microcontrollers.

- To understand the functions of the μPD780851(A), 780852(A), and 78F0852 in general:
  → Read this manual in the order of the **CONTENTS**.
- How to read register formats:
  → The name of a bit whose number is enclosed in square is reserved for the RA78K/0 and is defined for the CC78K/0 by the header file sfrbit.h.
- To learn the detailed functions of a register whose register name is known:
  → See **APPENDIX C REGISTER INDEX**.

The application examples in this manual are for the "standard" model for general-purpose electronic systems. If the examples in this manual are to be used for applications where a quality higher than that of the "special" model is required, study the quality grade of the respective components and circuits actually used.

Conventions

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

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

- **Related documents for μPD780852 Subseries**

Document Name	Document No.	
	Japanese	English
μPD780851(A), 780852(A) Preliminary Product Information	U14577J	U14577E
μPD78F0852 Preliminary Product Information	U14576J	U14576E
μPD780852 Subseries User's Manual	U14581J	This manual
78K/0 Series User's Manual Instructions	U12326J	U12326E
78K/0 Series Instruction Table	U10903J	—
78K/0 Series Instruction Set	U10904J	—

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

Document Name		Document No.	
		Japanese	English
RA78K0 Assembler Package	Operation	U11802J	U11802E
	Language	U11801J	U11801E
	Structured Assembly Language	U11789J	U11789E
CC78K0 C Compiler	Operation	U11517J	U11517E
	Language	U11518J	U11518E
CC78K0 C Compiler Application Note	Programming Know-how	U13034J	U13034E
IE-78K0-NS		U13731J	U13731E
IE-780852-NS-EM4		To be prepared	To be prepared
SM78K0 System Simulator Windows™ Base	Reference	U10181J	U10181E
SM78K Series System Simulator	External Part User Open Interface Specifications	U10092J	U10092E
ID78K0 Integrated Debugger EWS Base	Reference	U11151J	—
ID78K0 Integrated Debugger Windows Base	Guide	U11649J	U11649E
ID78K0 Integrated Debugger PC Base	Reference	U11539J	U11539E

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

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

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

- **Other related documents**

Document Name	Document No.	
	Japanese	English
SEMICONDUCTOR SELECTION GUIDE Products & Packages (CD-ROM)	X13769X	
Semiconductor Device Mounting Technology Manual	C10535J	C10535E
Quality Grades on NEC Semiconductor Device	C11531J	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983J	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892J	C11892E
Semiconductor Device Quality Control/Reliability Handbook	U12769J	—
Guide for Products Related to Micro-Computer: Other Companies	U11416J	—

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

[MEMO]

CONTENTS

LIST OF FIGURES (1/5)

LIST OF FIGURES (2/5)

LIST OF FIGURES (3/5)

LIST OF FIGURES (4/5)

LIST OF FIGURES (5/5)

LIST OF TABLES (1/2)

LIST OF TABLES (2/2)

[MEMO]

CHAPTER 1 OUTLINE

1.1 Features

- Internal memory

Item / Part Number	Program Memory (ROM/Flash Memory)	Data Memory		
		Internal High-Speed RAM	Internal Expansion RAM	LCD Display RAM
μPD780851(A)	32 Kbytes	1,024 bytes	512 bytes	20 × 4 bits
μPD780852(A)	40 Kbytes			
μPD78F0852	40 Kbytes			

- Instruction execution time can be changed from high-speed (0.24 μs) to low-speed (3.81 μs)
- Instruction set suited to system control
  - Bit manipulation possible in all address spaces
  - Multiply and divide instructions
- Fifty-six I/O ports: (including pins that have an alternate function as segment signal output)
- 8-bit resolution A/D converter: 5 channels
- Sound generator: 1 channel
- Meter controller/driver: PWM output (8-bit resolution): 16
  Can set pulse width with a precision of 8 + 1 bits with 1-bit addition function
- LCD controller/driver
  - Segment signal output: 20 max.
  - Common signal output: 4 max.
  - Bias: 1/3 bias
  - Power supply voltage: V_{LCD} = 3.0 V to V_{DD}
- Serial interface: 3 channels
  - 3-wire serial I/O mode: 2 channels
  - UART mode: 1 channel
- Timer: Six channels
  - 16-bit timer: 1 channel
  - 8-bit timer: 1 channel
  - 8-bit timer/event counter: 2 channels
  - Watch timer: 1 channel
  - Watchdog timer: 1 channel
- Vectored interrupt sources: 21
- Power supply voltage: V_{DD} = 4.0 to 5.5 V

1.2 Applications

Automobile meter (dash board) control

1.3 Ordering Information

Part Number	Package	Internal ROM
μPD780851GC(A)-×××-8BT	80-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD780852GC(A)-×××-8BT	80-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78F0852GC-8BT	80-pin plastic QFP (14 × 14 mm)	Flash memory

Remark ××× indicates ROM code suffix.

1.4 Quality Grade

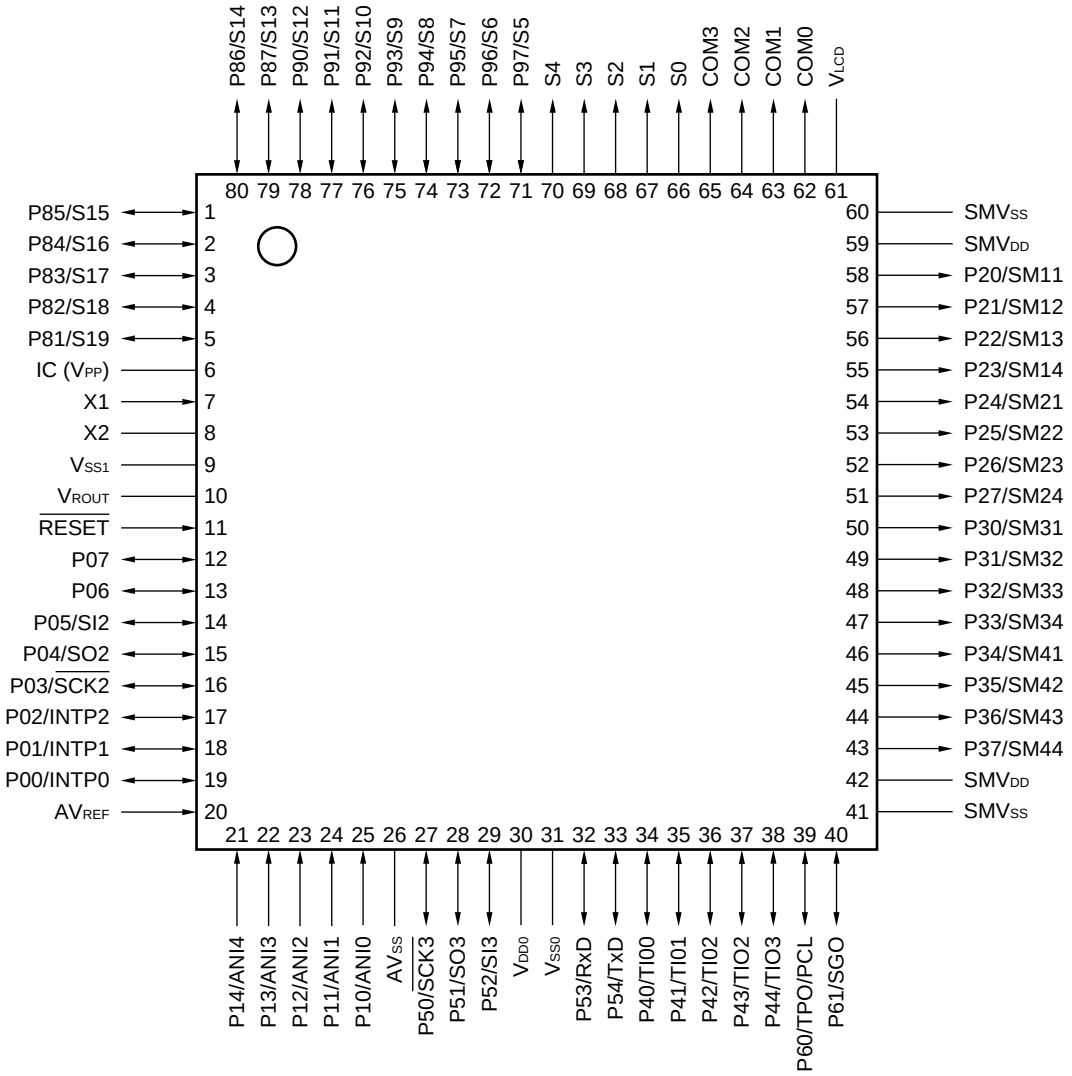
Part Number	Package	Quality Grade
μPD78F0852GC-8BT	80-pin plastic QFP (14 × 14 mm)	Standard
μPD780851GC(A)-×××-8BT	80-pin plastic QFP (14 × 14 mm)	Special
μPD780852GC(A)-×××-8BT	80-pin plastic QFP (14 × 14 mm)	Special

Remark ××× indicates ROM code suffix.

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

1.5 Pin Configuration (Top View)

★ • **80-pin plastic QFP (14 × 14 mm)**
μPD780851GC(A)-×××-8BT, 780852GC(A)-×××-8BT, 78F0852GC-8BT



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

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

3. Connect AV_{REF} pin to V_{DD0}.

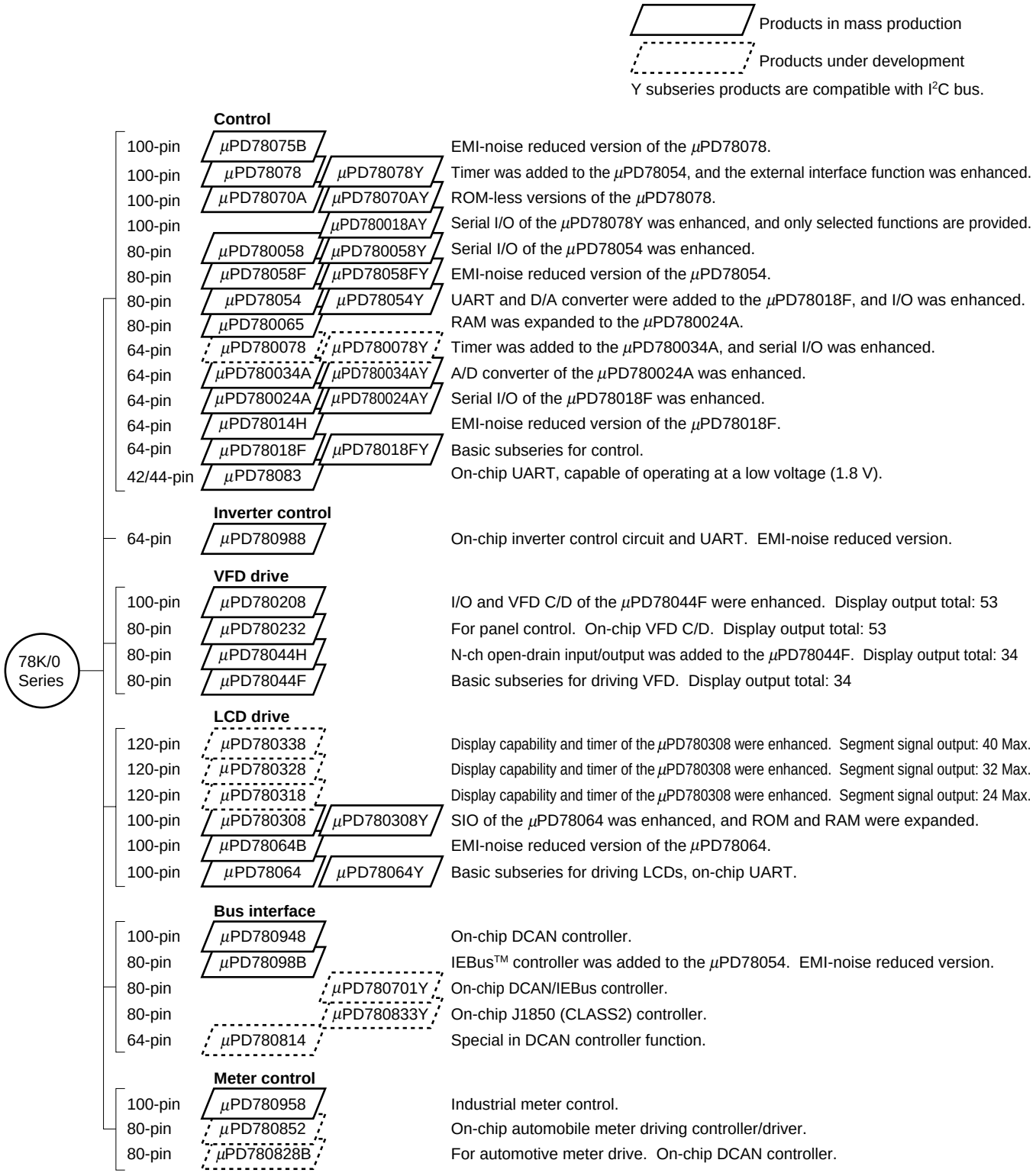
Remarks 1. When these devices are used in applications that require reduction of the noise generated from inside the microcontroller, the implementation of noise reduction measures, such as connecting the V_{SS0} and V_{SS1} to different ground lines, is recommended.

2. Pin connection in parentheses is intended for the μPD78F0852.

ANI0 to ANI4:	Analog Input
AV_{REF}:	Analog Reference Voltage
AV_{SS}:	Analog Ground
COM0 to COM3:	Common Output
IC:	Internally Connected
INTP0 to INTP2:	External Interrupt Input
P00 to P07:	Port0
P10 to P14:	Port1
P20 to P27:	Port2
P30 to P37:	Port3
P40 to P44:	Port4
P50 to P54:	Port5
P60, P61:	Port6
P81 to P87:	Port8
P90 to P97:	Port9
PCL:	Programmable Clock Output
$\overline{\text{RESET}}$:	Reset
RxD:	Receive Data
S0 to S19:	Segment Output
$\overline{\text{SCK2}}$, $\overline{\text{SCK3}}$:	Serial Clock
SGO:	Sound Generator Output
SI2, SI3:	Serial Input
SM11 to SM14, SM21 to SM24, SM31 to SM34, SM41 to SM44:	Meter Output
SMV_{DD}:	Meter Controller Power Supply
SMV_{SS}:	Meter Controller Ground
SO2, SO3:	Serial Output
TI00 to TI02:	Timer Input
TIO2, TIO3:	Timer Output/Event Counter Input
TPO:	Prescaler Output
TxD:	Transmit Data
V_{DD0}:	Power Supply
V_{LCD}:	LCD Power Supply
V_{PP}:	Programming Power Supply
V_{ROUT}:	Power Supply Regulator Output
V_{SS0}, V_{SS1}:	Ground
X1, X2:	Crystal (Main System Clock)

1.6 78K/0 Series Product Development

These products are a further development in the 78K/0 Series. The designations appearing inside the boxes are subseries names.



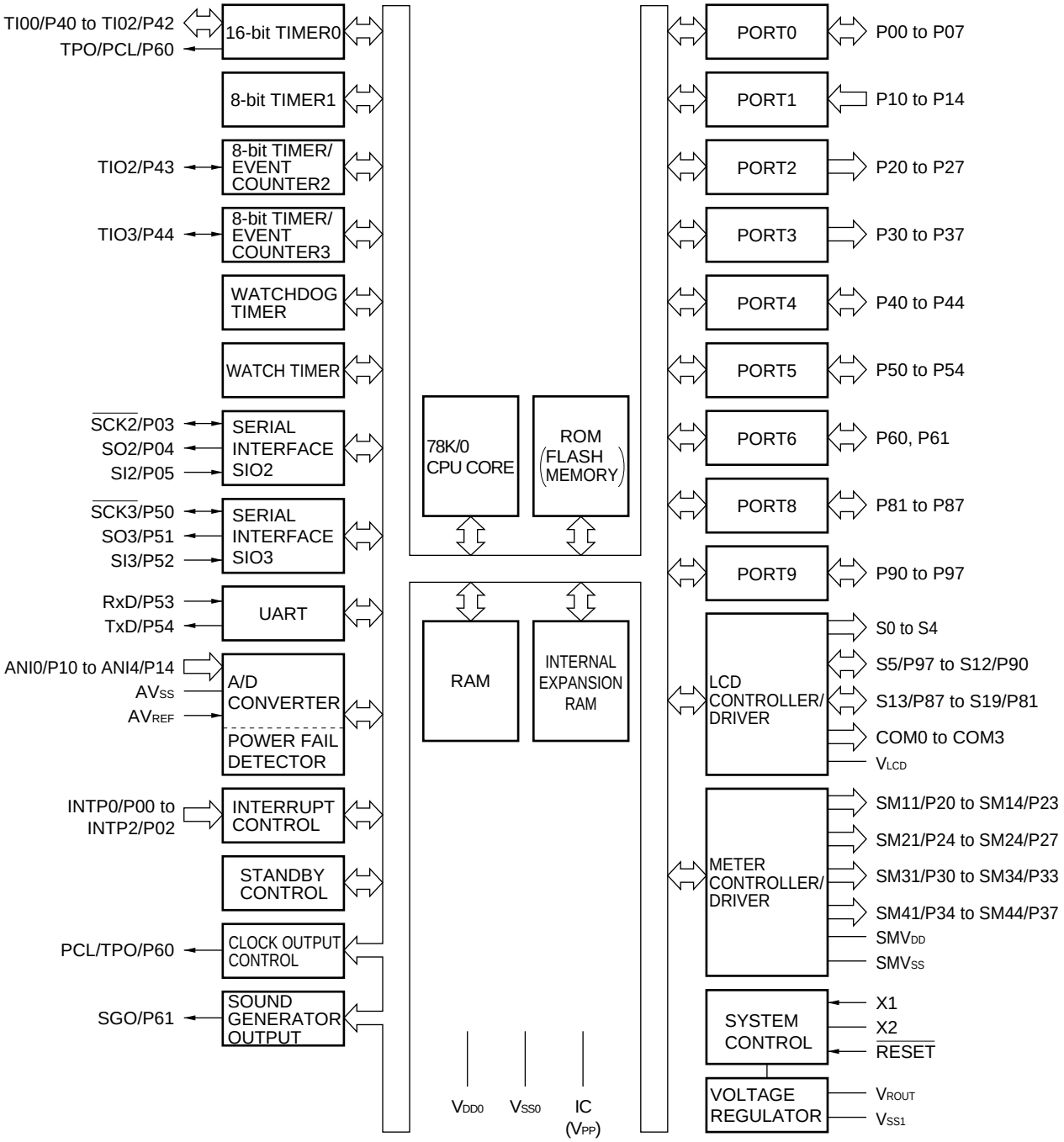
Remark Some documents use the name fluorescent indicator panel (FIP) instead of vacuum fluorescent display (VFD). The functions of FIP and VFD are the same.

The major functional differences among the subseries are shown below.

Subseries Name \ Function		ROM Capacity	Timer				8-Bit A/D	10-Bit A/D	8-Bit D/A	Serial Interface	I/O	V_{DD} MIN. Value	External Expansion
			8-Bit	16-Bit	Watch	WDT							
Control	μPD78075B	32 K to 40 K	4 ch	1 ch	1 ch	1 ch	8 ch	–	2 ch	3 ch (UART: 1 ch)	88	1.8 V	Available
	μPD78078	48 K to 60 K											
	μPD78070A	–									61	2.7 V	
	μPD780058	24 K to 60 K	2 ch							3 ch (time-division UART: 1ch)	68	1.8 V	
	μPD78058F	48 K to 60 K								3 ch (UART: 1 ch)	69	2.7 V	
	μPD78054	16 K to 60 K										2.0 V	
	μPD780065	40 K to 48 K							–	4 ch (UART: 1 ch)	60	2.7 V	
	μPD780078	48 K to 60 K		2 ch			–	8 ch		3 ch (UART: 2 ch)	52	1.8 V	
	μPD780034A	8 K to 32 K		1 ch						3 ch (UART: 1 ch)	51		
	μPD780024A						8 ch	–					
	μPD78014H									2 ch	53		
	μPD78018F	8 K to 60 K											
	μPD78083	8 K to 16 K		–	–					1 ch (UART: 1 ch)	33		–
Inverter control	μPD780988	16 K to 60 K	3 ch	**Note**	–	1 ch	–	8 ch	–	3 ch (UART: 2 ch)	47	4.0 V	Available
VFD	μPD780208	32 K to 60 K	2 ch	1 ch	1 ch	1 ch	8 ch	–	–	2 ch	74	2.7 V	–
	μPD780232	16 K to 24 K	3 ch	–	–		4 ch				40	4.5 V	
	μPD78044H	32 K to 48 K	2 ch	1 ch	1 ch		8 ch			1 ch	68	2.7 V	
	μPD78044F	16 K to 40 K								2 ch			
LCD drive	μPD780338	48 K to 60 K	3 ch	2 ch	1 ch	1 ch	–	10 ch	1 ch	2 ch (UART: 1 ch)	54	1.8 V	–
	μPD780328										62		
	μPD780318										70		
	μPD780308	48 K to 60 K	2 ch	1 ch			8 ch	–	–	3 ch (time-division UART: 1 ch)	57	2.0 V	
	μPD78064B	32 K								2 ch (UART: 1 ch)			
	μPD78064	16 K to 32 K											
Bus interface	μPD780948	60 K	2 ch	2 ch	1 ch	1 ch	8 ch	–	–	3 ch (UART: 1 ch)	79	4.0 V	Available
	μPD78098B	40 K to 60 K		1 ch					2 ch		69	2.7 V	–
	μPD780814	32 K to 60 K		2 ch			12 ch		–	2 ch (UART: 1 ch)	46	4.0 V	
Meter control	μPD780958	48 K to 60 K	4 ch	2 ch	–	1 ch	–	–	–	2 ch (UART: 1 ch)	69	2.2 V	–
Dash board control	μPD780852	32 K to 40 K	3 ch	1 ch	1 ch	1 ch	5 ch	–	–	3 ch (UART: 1 ch)	56	4.0 V	–
	μPD780828B	32 K to 60 K									59		

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

1.7 Block Diagram



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

2. Memory type in parentheses is for the μPD78F0852.

1.8 Outline of Function

Item \ Part Number		μPD780851(A)	μPD780852(A)	μPD78F0852
Internal memory	ROM	32 Kbytes (Mask ROM)	40 Kbytes (Mask ROM)	40 Kbytes (Flash memory)
	High-speed RAM	1,024 bytes		
	Expanded RAM	512 bytes		
	LCD display RAM	20 × 4 bits		
General register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)		
Minimum instruction execution time		On-chip minimum instruction execution timer variable function 0.24 μs/0.48 μs/0.95 μs/1.91 μs/3.81 μs (in operation at 8.38 MHz)		
Instruction set		• 16-bit operation • Multiply/divide (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, and Boolean operation) • BCD adjust, etc.		
I/O port (including pins shared with segment signal output)		Total: 56 • CMOS input: 5 • CMOS output: 16 • CMOS input/output: 35		
A/D converter		• 8-bit resolution × 5 channels • Power-fail detection function		
LCD controller/driver		• Segment signal outputs: 20 max. • Common signal outputs: 4 max. • Bias: 1/3 bias only		
Serial interface		• 3-wire serial I/O mode: 2 channels • UART mode: 1 channel		
Timer		• 16-bit timer: 1 channel • 8-bit timer: 1 channel • 8-bit timer/event counter: 2 channels • Watch timer: 1 channel • Watchdog timer: 1 channel		
Timer outputs		2 (8-bit PWM output: 2)		
Meter controller/driver		PWM output (8-bit resolution): 16 Can set pulse width with a precision of 8 + 1 bits with 1-bit addition function		
Sound generator		1 channel		
Clock output		65.5 kHz, 131 kHz, 262 kHz, 524 kHz, 1.04 MHz, 2.09 MHz, 4.19 MHz, 8.38 MHz (@ 8.38-MHz operation with main system clock)		
Vectored interrupt source	Maskable	Internal: 16, External: 3		
	Non-maskable	Internal: 1		
	Software	1		
★ Power supply voltage		V_{DD} = SMV_{DD} = 4.0 to 5.5 V		
Operating ambient temperature		T_A = –40°C to +85°C		
Package		80-pin plastic QFP (14 × 14 mm)		

CHAPTER 2 PIN FUNCTION

2.1 Pin Function List

(1) Port Pins

Pin Name	Input/Output	Function	After Reset	Alternate Function
P00 to P02	Input/Output	Port 0 8-bit input/output port. Input/output mode can be specified in 1-bit units. On-chip pull-up resistor can be used by software.	Input	INTP0 to INTP2
P03				$\overline{\text{SCK2}}$
P04				SO2
P05				SI2
P06, P07				–
P10 to P14	Input	Port 1 5-bit input only port.	Input	ANI0 to ANI4
P20 to P23	Output	Port 2 8-bit output only port.	Hi-Z	SM11 to SM14
P24 to P27				SM21 to SM24
P30 to P33	Output	Port 3 8-bit output only port.	Hi-Z	SM31 to SM34
P34 to P37				SM41 to SM44
P40 to P42	Input/Output	Port 4 5-bit input/output port. Input/output mode can be specified in 1-bit units.	Input	TI00 to TI02
P43, P44				TIO2, TIO3
P50	Input/Output	Port 5 5-bit input/output port. Input/output mode can be specified in 1-bit units.	Input	$\overline{\text{SCK3}}$
P51				SO3
P52				SI3
P53				RxD
P54				TxD
P60	Input/Output	Port 6 2-bit input/output port. Input/output mode can be specified in 1-bit units.	Input	PCL/TPO
P61				SGO
P81 to P87	Input/Output	Port 8 7-bit input/output port. Input/output mode can be specified in 1-bit units. Can be set in I/O port mode or segment output mode in 2-bit units with the LCD display control register (LCDC).	Input	S19 to S13
P90 to P97	Input/Output	Port 9 8-bit input/output port. Input/output mode can be specified in 1-bit units. Can be set in I/O port mode or segment output mode in 2-bit units with the LCD display control register (LCDC).	Input	S12 to S5

(2) Non-port pins

Pin Name	Input/Output	Function	After Reset	Alternate Function
INTP0 to INTP2	Input	External interrupt request input with specifiable valid edges (rising edge, falling edge, and both rising and falling edges).	Input	P00 to P02
SI2	Input	Serial interface SIO2 serial data input.	Input	P05
SO2	Output	Serial interface SIO2 serial data output.	Input	P04
$\overline{\text{SCK2}}$	Input/Output	Serial interface SIO2 serial clock input/output.	Input	P03
SI3	Input	Serial interface SIO3 serial data input.	Input	P52
SO3	Output	Serial interface SIO3 serial data output.	Input	P51
$\overline{\text{SCK3}}$	Input/Output	Serial interface SIO3 serial clock input/output.	Input	P50
RxD	Input	Asynchronous serial interface serial data input.	Input	P53
TxD	Output	Asynchronous serial interface serial data output.	Input	P54
TI00	Input	Capture trigger signal input to capture register (CR00).	Input	P40
TI01		Capture trigger signal input to capture register (CR01).		P41
TI02		Capture trigger signal input to capture register (CR02).		P42
TIO2	Input/Output	8-bit timer (TM2) input/output (also used for 8-bit PWM output).	Input	P43
TIO3		8-bit timer (TM3) input/output (also used for 8-bit PWM output).		P44
TPO	Output	Prescaler signal output of 16-bit timer (TM0).	Input	PCL/P60
PCL	Output	Clock output (for main system clock trimming).	Input	TPO/P60
SGO	Output	Sound generator signal output.	Input	P61
S0 to S4	Output	Segment signal output of LCD controller/driver.	Output	—
S5 to S12			Input	P97 to P90
S13 to S19				P87 to P81
COM0 to COM3	Output	Common signal output of LCD controller/driver.	Output	—
V_{LCD}	—	LCD driving power supply.	—	—
SM11 to SM14	Output	Meter control signal output.	Hi-Z	P20 to P23
SM21 to SM24				P24 to P27
SM31 to SM34				P30 to P33
SM41 to SM44				P34 to P37
ANI0 to ANI4	Input	A/D converter analog input.	Input	P10 to P14
AV_{REF}	Input	A/D converter reference voltage input (shared with analog power supply (AV_{DD})).	—	—
AV_{SS}	—	A/D converter ground potential. Connect to V_{SS0}.	—	—
$\overline{\text{RESET}}$	Input	System reset input.	—	—
X1	Input	Crystal connection for main system clock oscillation.	—	—
X2	—		—	—
SMV_{DD}	—	Power supply for meter controller/driver.	—	—
SMV_{SS}	—	Ground potential for meter controller/driver.	—	—
V_{DD0}	—	Port block positive power supply.	—	—
V_{SS0}	—	Port block ground potential.	—	—

Pin Name	Input/Output	Function	After Reset	Alternate Function
V_{ROUT}	—	Regulator output pin for power supply of pins other than port pins. Connect this pin to V_{SS0} or V_{SS1} via a 0.1-μF capacitor.	—	—
V_{SS1}	—	Ground potential (except for port block).	—	—
V_{PP}	—	High-voltage application for program write/verify. Connect directly to V_{SS0} or V_{SS1} in normal operation mode (μPD78F0852 only).	—	—
IC	—	Internally connected. Connect directly to V_{SS0} or V_{SS1}.	—	—

2.2 Description of Pin Functions

2.2.1 P00 to P07 (Port 0)

These pins constitute an 8-bit input/output port. In addition, they also function as external interrupt request input, serial interface data input/output, and clock input/output pins.

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

(1) Port mode

In this mode, P00 to P07 function as an 8-bit input/output port.

P00 to P07 can be specified as an input or output port in 1-bit units with port mode register 0 (PM0). On-chip pull-up resistor can be used in 1-bit units with the pull-up resistor option register 0 (PU0).

(2) Control mode

In this mode, P00 to P07 function as external interrupt request input, serial interface data input/output, and clock input/output pins.

(a) INTP0 to INTP2

These are external interrupt input pins for which the valid edge (rising edge, falling edge, and both the rising and falling edges) can be specified.

(b) SI2

Serial interface serial data input pin.

(c) SO2

Serial interface serial data output pin.

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

Serial interface serial clock input/output pin.

2.2.2 P10 to P14 (Port 1)

These pins constitute a 5-bit input only port. In addition, they also function as A/D converter analog input pins.

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

(1) Port mode

In this mode, P10 to P14 function as a 5-bit input only port.

(2) Control mode

In this mode, P10 to P14 function as A/D converter analog input pins (ANI0 to ANI4).

2.2.3 P20 to P27 (Port 2)

These pins constitute an 8-bit output only port. In addition, they also function as PWM output pins for meter control. The following operation modes can be specified in 1-bit units.

(1) Port mode

In this mode, P20 to P27 function as an 8-bit output only port. They go into a high-impedance state when 1 is set to port mode register 2 (PM2).

(2) Control mode

In this mode, P20 to P27 function as PWM output pins (SM11 to SM14 and SM21 to SM24) for meter control.

2.2.4 P30 to P37 (Port 3)

These pins constitute an 8-bit output only port. In addition, they also function as PWM output pins for meter control. The following operation modes can be specified in 1-bit units.

(1) Port mode

In this mode, P30 to P37 function as an 8-bit output only port. They go into a high-impedance state when 1 is set to port mode register 3 (PM3).

(2) Control mode

In this mode, P30 to P37 function as PWM output pins (SM31 to SM34 and SM41 to SM44) for meter control.

2.2.5 P40 to P44 (Port 4)

These pins constitute a 5-bit input/output port. In addition, they also function as timer input/output pins. The following operation modes can be specified in 1-bit units.

(1) Port mode

In this mode, P40 to P44 function as a 5-bit input/output port. They can be set in the input or output port in 1-bit units with the port mode register 4 (PM4).

(2) Control mode

In this mode, P40 to P44 function as timer input/output pins.

(a) TIO2, TIO3

8-bit timer input/output pins.

(b) TI00 to TI02

These pins input a capture trigger signal to the 16-bit timer capture registers (CR00 to CR02).

2.2.6 P50 to P54 (Port 5)

These pins constitute a 5-bit input/output port. In addition, they also function as serial interface data input/output and clock input/output pins.

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

(1) Port mode

In this mode, P50 to P54 function as a 5-bit input/output port. They can be set in the input or output port in 1-bit units with the port mode register 5 (PM5).

(2) Control mode

In this mode, P50 to P54 function as serial interface data input/output and clock input/output pins.

(a) SI3

Serial interface serial data input pin.

(b) SO3

Serial interface serial data output pin.

(c) $\overline{\text{SCK3}}$

Serial interface serial clock input/output pin.

(d) RxD, TxD

Asynchronous serial interface serial data input/output pins.

2.2.7 P60, P61 (Port 6)

These pins constitute a 2-bit input/output port. In addition, they also function as clock output, sound generator signal output, and prescaler signal output pins.

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

(1) Port mode

In this mode, P60 and P61 function as a 2-bit input/output port. They can be set in the input or output port in 1-bit units with the port mode register 6 (PM6).

(2) Control mode

In this mode, P60 and P61 function as clock output, sound generator signal output, and prescaler signal output pins.

(a) PCL

Clock output pin.

(b) SGO

Sound generator (with amplitude) signal output pin.

(c) TPO

Prescaler signal output pin of the 16-bit timer.

2.2.8 P81 to P87 (Port 8)

These pins constitute a 7-bit input/output port. In addition, they also function as segment signal output pins of the LCD controller/driver.

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

(1) Port mode

In this mode, P81 to P87 function as a 7-bit input/output port. They can be set in the input or output port in 1-bit units with the port mode register 8 (PM8).

(2) Control mode

In this mode, P81 to P87 function as segment signal output pins (S13 to S19) of the LCD controller/driver.

2.2.9 P90 to P97 (Port 9)

These pins constitute an 8-bit input/output port. In addition, they also function as segment signal output pins of the LCD controller/driver.

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

(1) Port mode

In this mode, P90 to P97 function as an 8-bit input/output port. They can be set in the input or output port in 1-bit units with the port mode register 9 (PM9).

(2) Control mode

In this mode, P90 to P97 function as segment signal output pins (S5 to S12) of the LCD controller/driver.

2.2.10 COM0 to COM3

These pins output common signals from the LCD controller/driver during 4-time division drive in 1/3 bias mode (COM0 to COM3 outputs).

2.2.11 V_{LCD}

This pin supplies a voltage to drive an LCD.

2.2.12 AV_{REF}

This is an A/D converter reference voltage input pin. This pin also functions as an analog power supply pin (AV_{DD}). Supply power to this pin when the A/D converter is used.

When A/D converter is not used, connect this pin to V_{DD0}.

2.2.13 AV_{SS}

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

2.2.14 $\overline{\text{RESET}}$

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

2.2.15 X1 and X2

Crystal resonator connect pins for main system clock oscillation.

When using an external clock supply, input it to X1 and its inverted signal to X2.

2.2.16 SMV_{DD}

This pin supplies a positive power to the meter controller/driver.

2.2.17 SMV_{SS}

This is the ground pin of the meter controller/driver.

2.2.18 V_{DD0}

Positive power supply port pin.

2.2.19 V_{ROUT}

This is a regulator output pin for the power supply to pins other than port pins. Connect this pin to V_{SS0} or V_{SS1} via a 0.1-μF capacitor**Note**. Do not use this pin to supply power to other ICs.

Note The size of the capacitor to be connected will be finalized after evaluation.

2.2.20 V_{SS0}, V_{SS1}

The V_{SS0} pin is the ground potential port pin.

The V_{SS1} pin is the ground potential pin except for port block.

2.2.21 V_{PP} (μPD78F0852 only)

A high voltage should be applied to this pin when the flash memory programming mode is set and when the program is written or verified.

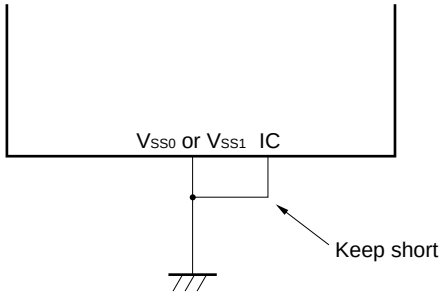
Directly connect this pin to V_{SS0} or V_{SS1} in the normal operation mode.

2.2.22 IC (Mask ROM version only)

The IC (Internally Connected) pin is provided to set the test mode to check the μPD780852 Subseries before shipment. In the normal operation mode, directly connect this pin to the V_{SS0} or V_{SS1} pin with as short a wiring length as possible.

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

- Directly connect the IC pin to the V_{SS0} or V_{SS1} pin.



2.3 Input/Output Circuits and Recommended Connection of Unused Pins

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

Table 2-1. Pin Input/Output Circuit Types

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
P00/INTP0 to P02/INTP2	8-A	Input/output	Independently connect to V_{SS0} via a resistor.
P03/$\overline{SCK2}$			
P04/SO2			
P05/SI2			
P06, P07			
P10/ANI0 to P14/ANI4	9	Input	Independently connect to V_{DD0} or V_{SS0} via a resistor.
P20/SM11 to P23/SM14	4	Output	Leave open.
P24/SM21 to P27/SM24			
P30/SM31 to P33/SM34			
P34/SM41 to P37/SM44			
P40/TI00 to P42/TI02	8	Input/output	Independently connect to V_{DD0} or V_{SS0} via a resistor.
P43/TIO2			
P44/TIO3			
P50/$\overline{SCK3}$			
P51/SO3	5		
P52/SI3	8		
P53/RxD			
P54/TxD	5		
P60/PCL/TPO			
P61/SGO			
P81/S19 to P87/S13	17-G		
P90/S12 to P97/S5			
S0 to S4	17	Output	Leave open.
COM0 to COM3	18		
V_{LCD}	—	—	
$\overline{RESET}$	2	Input	—
SMV_{DD}	—	—	Connect to V_{DD0}.
SMV_{SS}			Connect to V_{SS0}.
AV_{REF}			Connect to V_{DD0}.
AV_{SS}			Connect to V_{SS0}.
V_{PP}			Connect directly to V_{SS0} or V_{SS1}.
IC			

Figure 2-1. I/O Circuits of Pins (1/2)

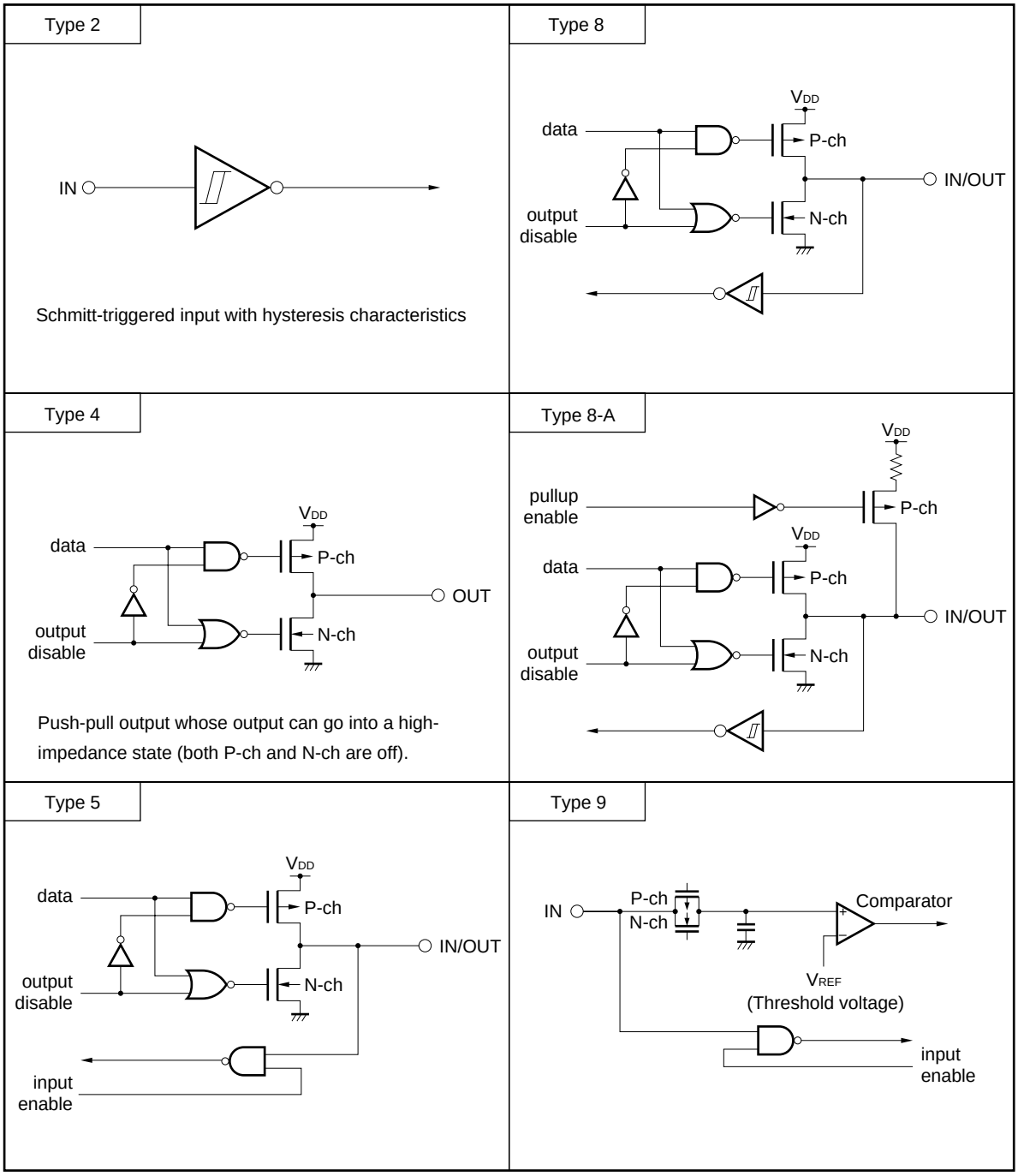
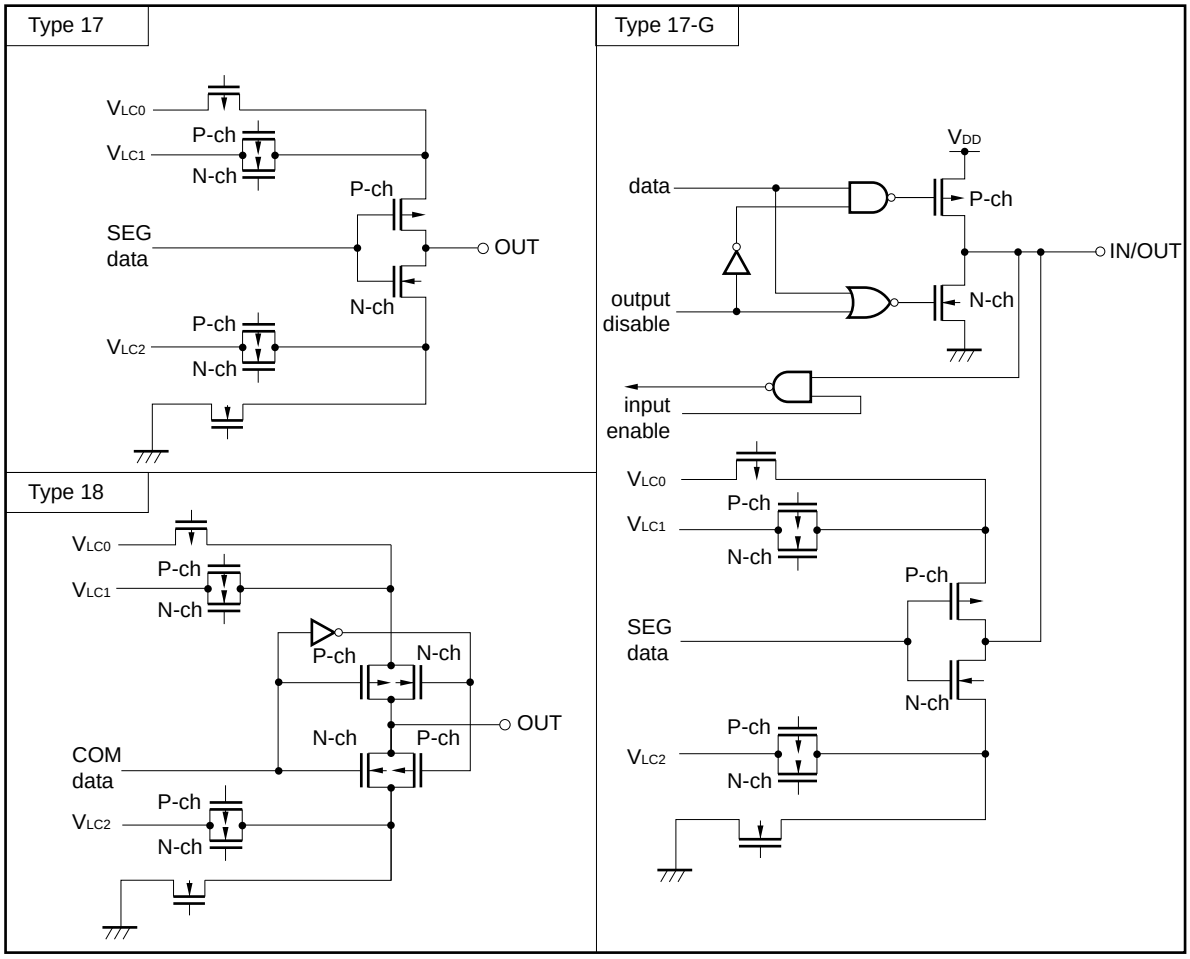


Figure 2-1. I/O Circuits of Pins (2/2)



[MEMO]

CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Spaces

The μPD780852 Subseries can access a 64-Kbyte memory space. Figures 3-1 to 3-3 show memory maps of the respective devices.

Figure 3-1. Memory Map (μPD780851(A))

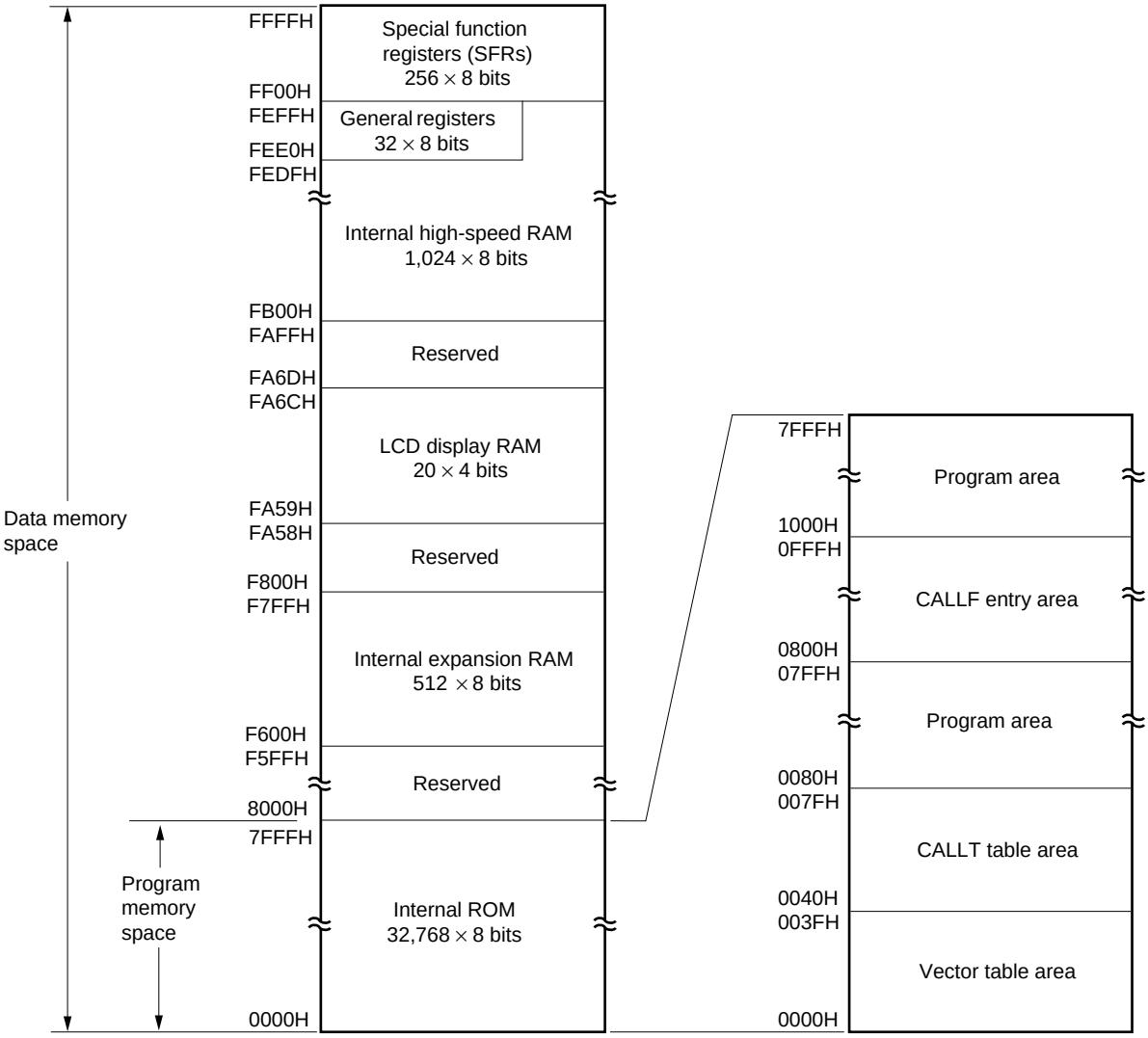


Figure 3-2. Memory Map (μPD780852(A))

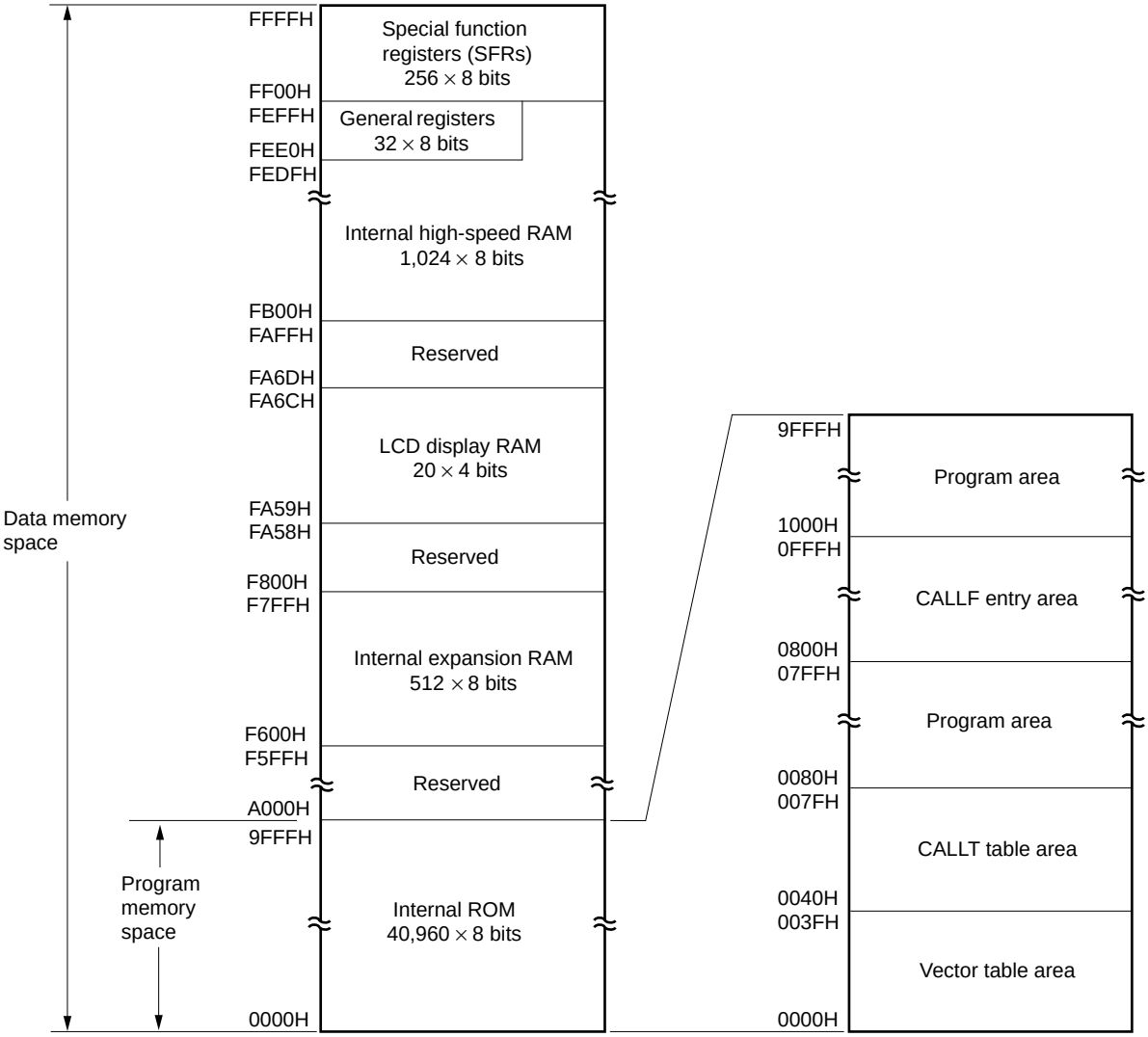
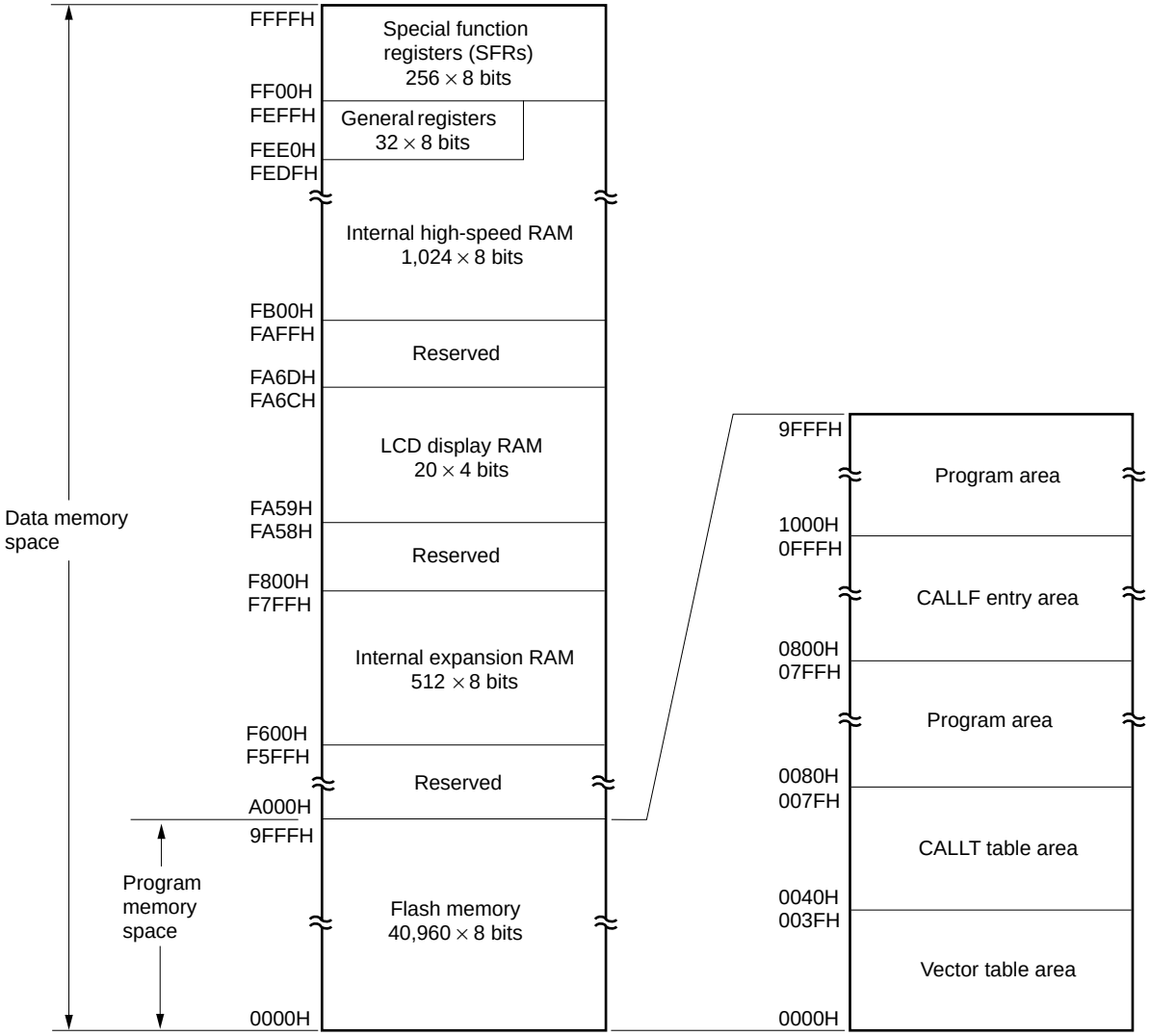


Figure 3-3. Memory Map (μPD78F0852)



3.1.1 Internal program memory space

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

The μPD780852 Subseries incorporate internal ROM (or flash memory), as listed below.

Table 3-1. Internal Memory Capacity

Part Number	Internal ROM	
	Type	Capacity
μPD780851(A)	Mask ROM	32,768 × 8 bits (0000H to 7FFFH)
μPD780852(A)		40,960 × 8 bits (0000H to 9FFFH)
μPD78F0852	Flash Memory	40,960 × 8 bits (0000H to 9FFFH)

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

(1) Vector table area

The 64-byte area 0000H to 003FH is reserved as a vector table area. This area stores program start addresses to which execution branches when the $\overline{\text{RESET}}$ signal is input or when an interrupt request is generated.

Of a 16-bit address, the lower 8 bits are stored at an even address and the higher 8 bits are stored at an odd address.

Table 3-2. Vector Table

Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input
0004H	INTWDT
0006H	INTAD
0008H	INTOVF
000AH	INTTM00
000CH	INTTM01
000EH	INTTM02
0010H	INTP0
0012H	INTP1
0014H	INTP2
0016H	INTCSI3
0018H	INTSER
001AH	INTSR
001CH	INTST
001EH	INTTM1
0020H	INTTM2
0022H	INTTM3
0024H	INTCSI2
0026H	INTWTI
0028H	INTWT
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 μPD780852 Subseries have the following RAM.

(1) Internal high-speed RAM

The configuration of the internal high-speed RAM is 1,024 $\times$ 8 bits at addresses FB00H to FEFFH. The 32-byte area FEE0H to FEFFH is allocated with four general-purpose register banks composed of eight 8-bit registers. The internal high-speed RAM can be used as stack memory.

(2) LCD display RAM

An LCD display RAM is allocated to an area of 20 $\times$ 4 bits at addresses FA59H to FA6CH. The LCD display RAM can also be used as a normal RAM.

(3) Internal expansion RAM

An internal expansion RAM is allocated to an area of 512 $\times$ 8 bits at addresses F600H to F7FFH.

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 (see **Table 3-3 Special-Function Register List** in **3.2.3 Special-function registers (SFRs)**).

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

3.1.4 Data memory addressing

Addressing refers to the method of specifying the address of the instruction to be executed next or the address of the register or memory relevant to the execution of instructions. The address of an instruction to be executed next is addressed by the program counter (PC) (for details, see **3.3 Instruction Address Addressing**).

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the μPD780852 Subseries, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFRs) and general-purpose registers are available for use. Data memory addressing is illustrated in Figures 3-4 to 3-6. For the details of each addressing mode, see **3.4 Operand Address Addressing**.

Figure 3-4. Data Memory Addressing (μPD780851(A))

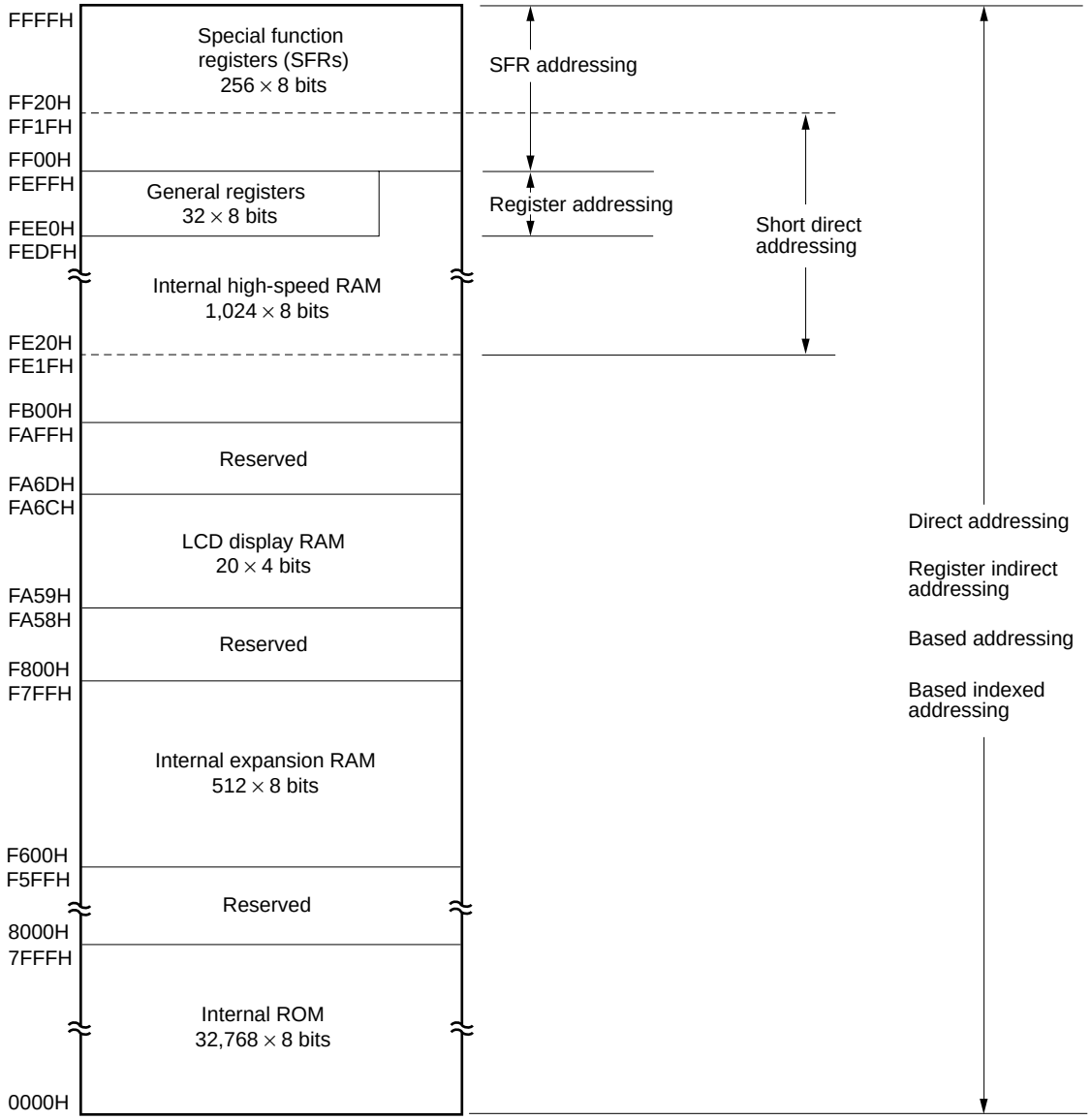


Figure 3-5. Data Memory Addressing (μPD780852(A))

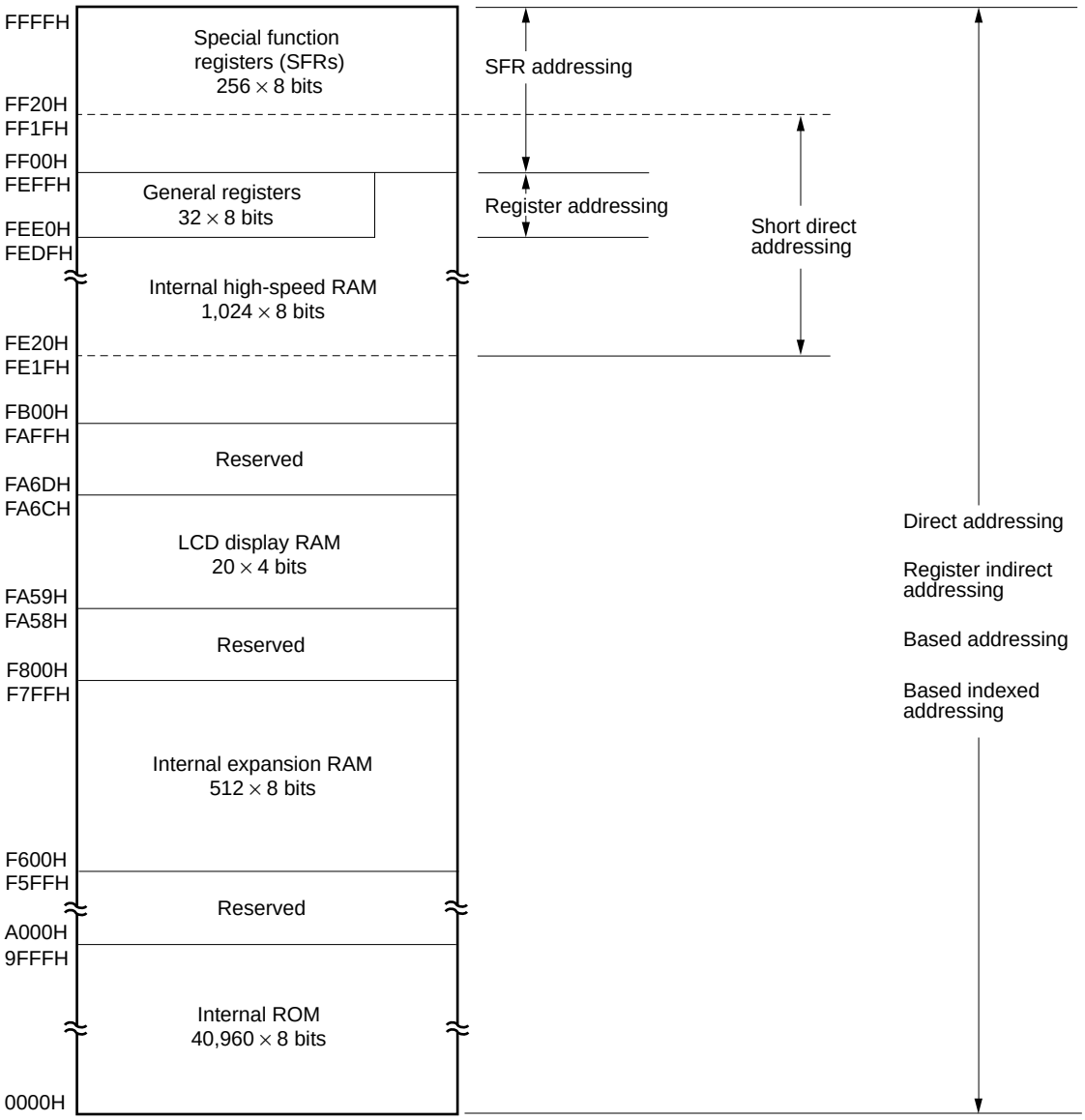
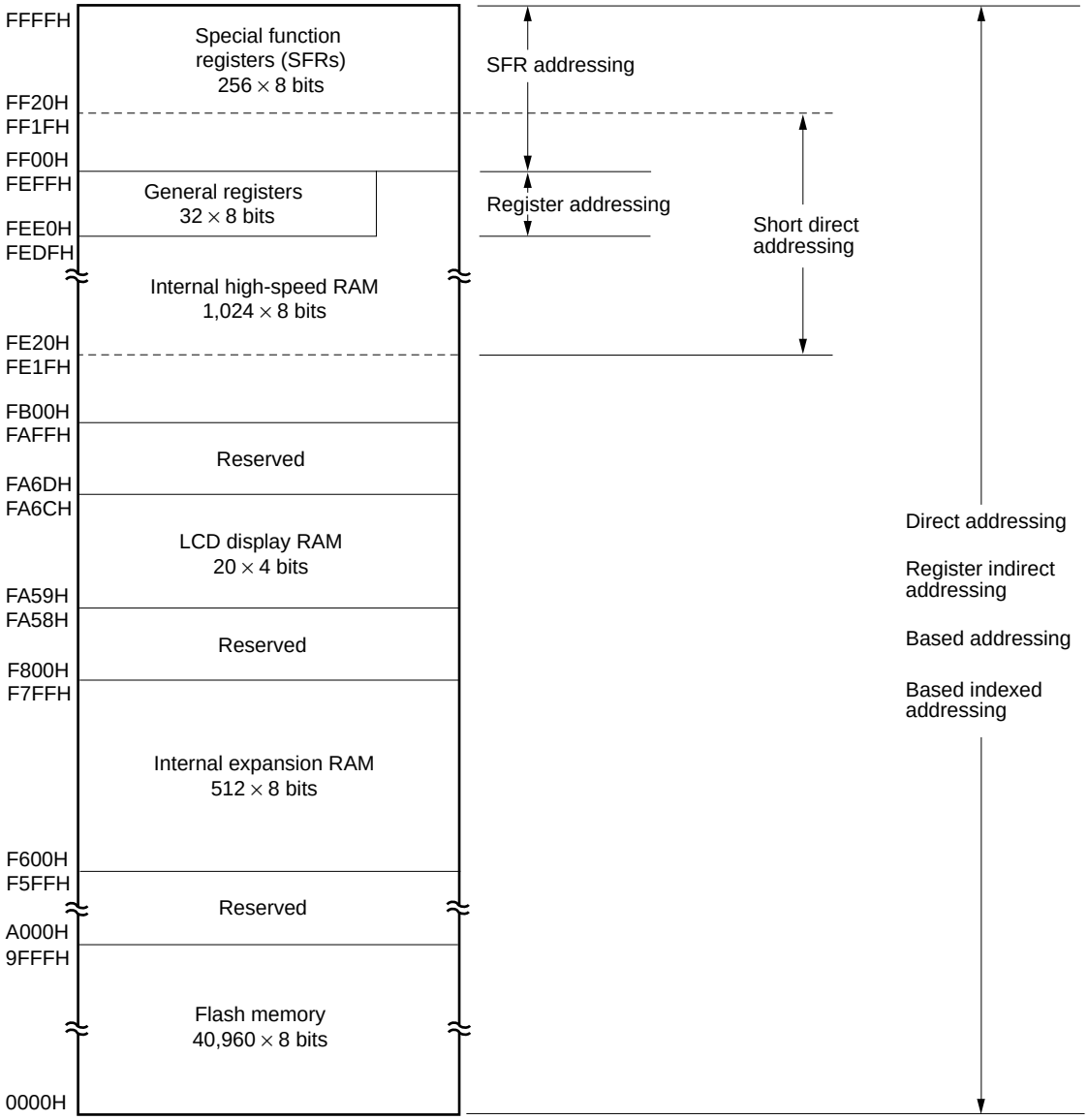


Figure 3-6. Data Memory Addressing (μPD78F0852)



3.2 Processor Registers

The μPD780852 Subseries incorporate the following processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

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

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

Figure 3-7. Program Counter Configuration

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

(2) Program status word (PSW)

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

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

Figure 3-8. Program Status Word Configuration

	7							0
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY

(a) Interrupt enable flag (IE)

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

When 0, the IE is set to DI, and only non-maskable interrupt request becomes acknowledgeable. Other interrupt requests are all disabled. When 1, the IE is set to EI and interrupt request acknowledge enable is controlled with an in-service priority flag (ISP), an interrupt mask flag for various interrupt sources and a priority specify flag.

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

(b) Zero flag (Z)

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

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

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

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

(d) Auxiliary carry flag (AC)

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

(e) In-service priority flag (ISP)

This flag manages the priority of acknowledgeable maskable vectored interrupts. When this flag is 0, low-level vectored interrupts specified with a priority specify flag register (PR0L, PR0H, PR1L) (see **19.3 (3) Priority specify flag registers (PR0L, PR0H, PR1L)**) are disabled for acknowledgement. Actual acknowledgement is controlled with the interrupt enable flag (IE).

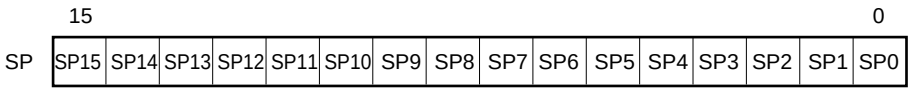
(f) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Figure 3-9. Stack Pointer Configuration



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

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

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

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

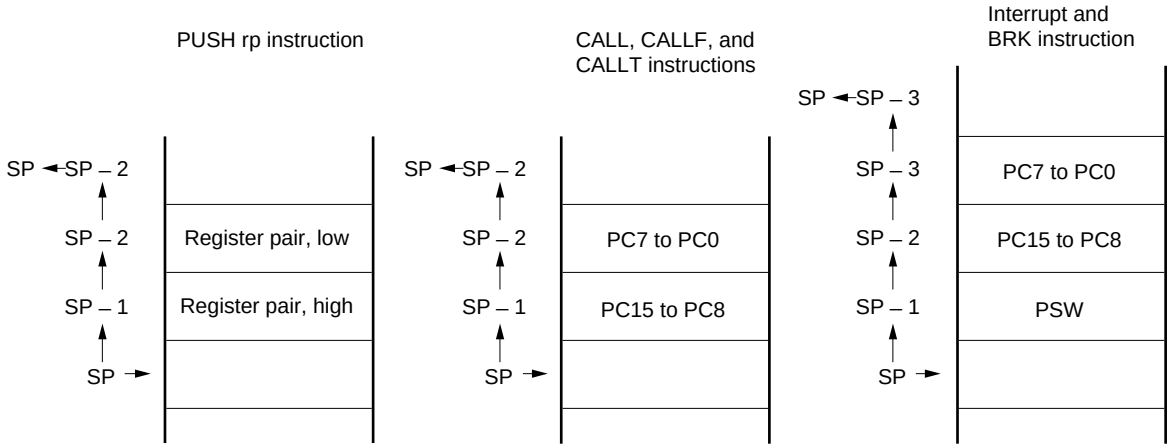
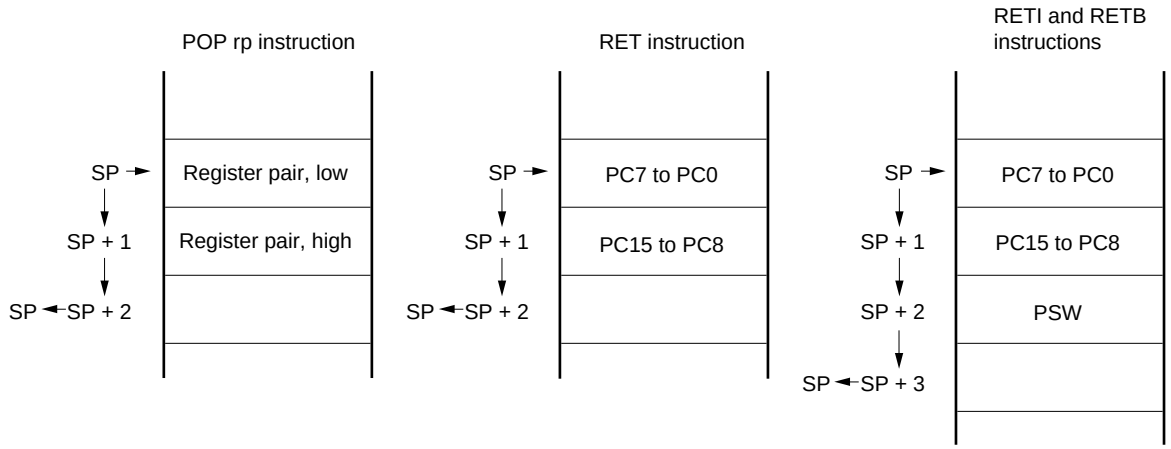


Figure 3-11. Data to Be Reset from Stack Memory



3.2.2 General registers

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

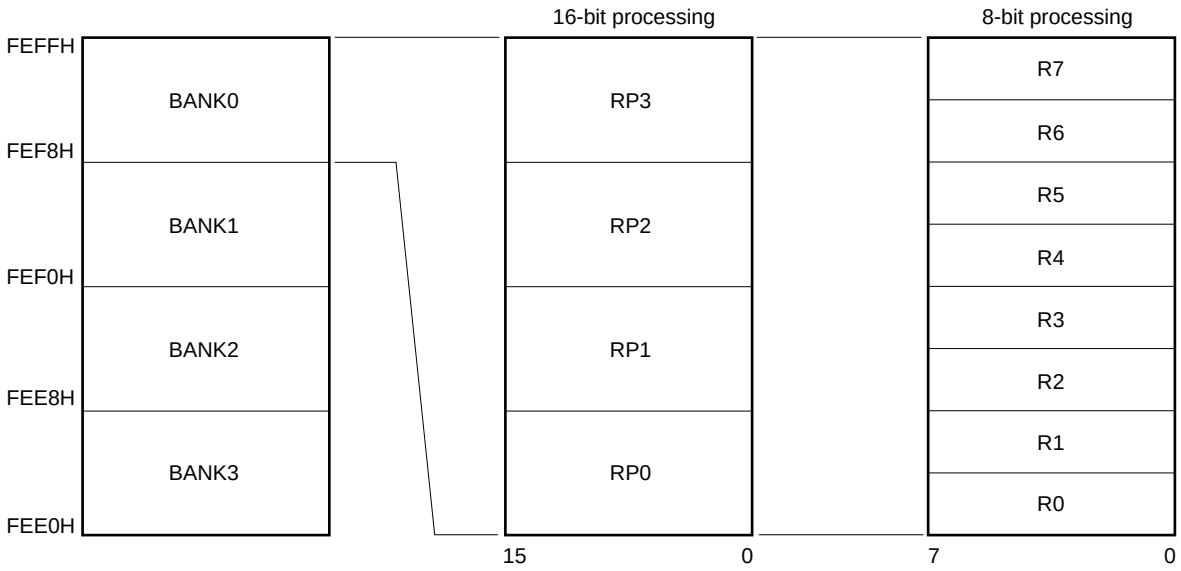
Each register can also be used as an 8-bit register. Two 8-bit registers can be used in pairs as a 16-bit register (AX, BC, DE, and HL).

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

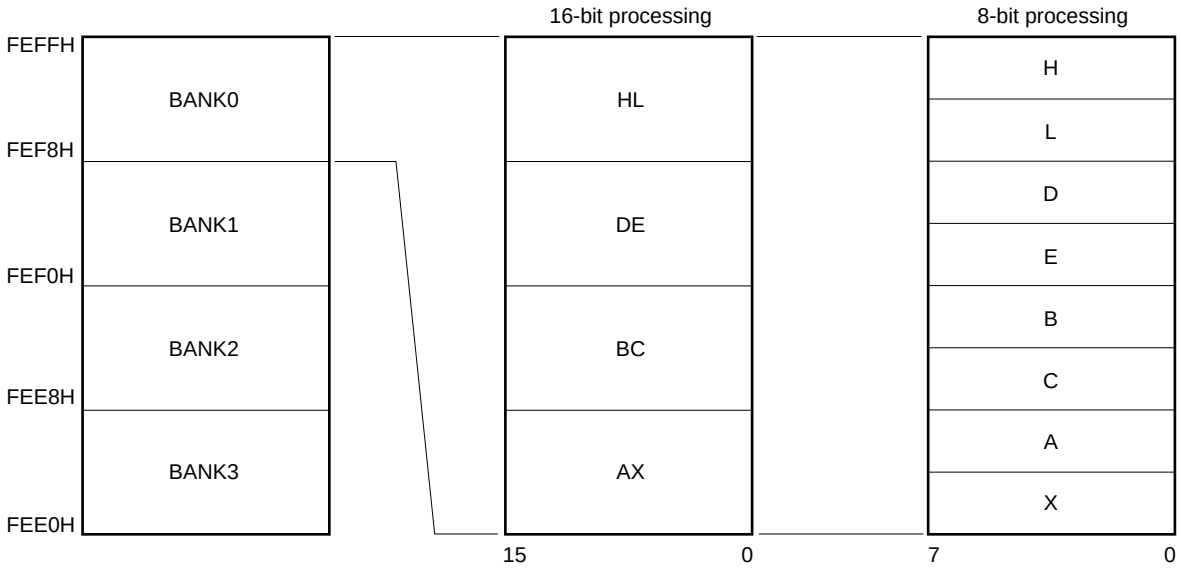
Register banks to be used for instruction execution are set with the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupt processing for each bank.

Figure 3-12. General Register Configuration

(a) Absolute Name



(b) Function Name



3.2.3 Special-function registers (SFRs)

Unlike the general registers, these registers have special functions.

They are allocated in the FF00H to FFFFH area.

The special-function registers can be manipulated like the general registers, with the operation, transfer and bit manipulation instructions. The bit units (1, 8, or 16 bits) for the manipulation vary for each register.

Each manipulation bit unit can be specified as follows.

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

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

- Symbol
  Symbol indicating the address of a special-function register. It is a reserved word in the RA78K/0, and is defined via the header file “sfrbit.h” in the CC78K/0. It can be described as an instruction operand when the RA78K/0, ID78K0-NS, ID78K0, and SM78K0 are used.
- R/W
  Indicates whether the corresponding special-function register can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulatable bit units
  Indicates the manipulatable bit unit (1, 8, or 16). “—” indicates a bit unit for which manipulation is not possible.
- After reset
  Indicates each register status upon $\overline{\text{RESET}}$ input.

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

Address	Special-Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0	R/W	○	○	—	00H
FF01H	Port 1	P1	R	—	○	—	
FF02H	Port 2	P2	R/W Note	○	○	—	
FF03H	Port 3	P3		○	○	—	
FF04H	Port 4	P4	R/W	○	○	—	
FF05H	Port 5	P5		○	○	—	
FF06H	Port 6	P6		○	○	—	
FF08H	Port 8	P8		○	○	—	
FF09H	Port 9	P9		○	○	—	
FF0AH	8-bit compare register 1	CR1		—	○	—	
FF0BH	8-bit compare register 2	CR2		—	○	—	
FF0CH	8-bit compare register 3	CR3		—	○	—	
FF0DH	8-bit counter 1	TM1	R	—	○	—	
FF0EH	8-bit counter 2	TM2		—	○	—	
FF0FH	8-bit counter 3	TM3		—	○	—	
FF10H	Capture register 00	CR00		—	—	○	0000H
FF11H							
FF12H	Capture register 01	CR01		—	—	○	
FF13H							
FF14H	Capture register 02	CR02		—	—	○	
FF15H							
FF16H	16-bit timer register 0	TM0		—	—	○	
FF17H							
FF18H	Serial I/O shift register 3	SIO3	R/W	—	○	—	00H
FF19H	Transmit shift register	TXS	W	—	○	—	FFH
	Receive buffer register	RXB	R	—	○	—	FFH
FF1BH	A/D conversion result register	ADCR1	R	—	○	—	00H
FF1FH	Serial I/O shift register 2	SIO2	R/W	—	○	—	

Note When PM2 and PM3 are set to 00H, read operation is enabled. Moreover, when PM2 and PM3 are set to FFH, these ports go into a high-impedance state.

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

Address	Special-Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF20H	Port mode register 0	PM0	R/W	○	○	—	FFH
FF22H	Port mode register 2	PM2		○	○	—	
FF23H	Port mode register 3	PM3		○	○	—	
FF24H	Port mode register 4	PM4		○	○	—	
FF25H	Port mode register 5	PM5		○	○	—	
FF26H	Port mode register 6	PM6		○	○	—	
FF28H	Port mode register 8	PM8		○	○	—	
FF29H	Port mode register 9	PM9		○	○	—	
FF30H	Pull-up resistor option register	PU0		○	○	—	00H
FF40H	Clock output selection register	CKS		○	○	—	
FF41H	Watch timer mode control register	WTM		○	○	—	
FF42H	Watchdog timer clock select register	WDCS		○	○	—	
FF48H	External interrupt rising edge enable register	EGP		○	○	—	
FF49H	External interrupt falling edge enable register	EGN		○	○	—	
FF4AH	LCD timer control register[Note]	LCDTM	W	○	○	—	
FF61H	Compare register (sin side)	MCMP10	R/W	○	○	—	
FF62H	Compare register (cos side)	MCMP11		○	○	—	
FF63H	Compare register (sin side)	MCMP20		○	○	—	
FF64H	Compare register (cos side)	MCMP21		○	○	—	
FF65H	Compare register (sin side)	MCMP30		○	○	—	
FF66H	Compare register (cos side)	MCMP31		○	○	—	
FF67H	Compare register (sin side)	MCMP40		○	○	—	
FF68H	Compare register (cos side)	MCMP41		○	○	—	
FF69H	Timer mode control register	MCNTC		○	○	—	
FF6AH	Port mode control register	PMC		○	○	—	
FF6BH	Compare control register 1	MCMPC1		○	○	—	
FF6CH	Compare control register 2	MCMPC2		○	○	—	
FF6DH	Compare control register 3	MCMPC3		○	○	—	
FF6EH	Compare control register 4	MCMPC4		○	○	—	
FF70H	Prescaler mode register	PRM0		○	○	—	
FF71H	Capture/compare control register	CRC0		○	○	—	
FF72H	16-bit timer mode control register	TMC0		○	○	—	
FF73H	Timer clock select register 1	TCL1		—	○	—	
FF74H	Timer clock select register 2	TCL2		—	○	—	
FF75H	Timer clock select register 3	TCL3		—	○	—	
FF76H	8-bit timer mode control register 1	TMC1		○	○	—	04H
FF77H	8-bit timer mode control register 2	TMC2		○	○	—	
FF78H	8-bit timer mode control register 3	TMC3		○	○	—	

Note LCDTM is a register that must be set when debugging the μPD780852 with an in-circuit emulator (IE-78K0-NS).

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

Address	Special-Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FF80H	A/D converter mode register	ADM1		R/W	○	○	—	00H
FF81H	Analog input channel specification register	ADS1			○	○	—	
FF82H	Power-fail compare mode register	PFM			○	○	—	
FF83H	Power-fail compare threshold value register	PFT			○	○	—	
FF84H	Serial operation mode register 3	CSIM3			○	○	—	
FF85H	Asynchronous serial interface mode register	ASIM			○	○	—	
FF86H	Asynchronous serial interface status register	ASIS		R	—	○	—	
FF87H	Baud rate generator control register	BRGC		R/W	—	○	—	
FF89H	D/A converter mode register Note 1	DAM1		W	○	○	—	
FF94H	Sound generator control register	SGCR		R/W	○	○	—	
FF95H	Sound generator buzzer control register	SGBR			○	○	—	
FF96H	Sound generator amplitude register	SGAM			○	○	—	
FF98H	Serial operation mode register 2	CSIM2			○	○	—	
FF99H	Serial receive data buffer register	SIRB2			—	○	—	Undefined
FF9AH	Serial receive data buffer status register	SRBS2			—	○	—	00H
FFA0H	Oscillator mode register Note 2	OSCM			○	○	—	
FFB0H	LCD display mode register	LCDM			○	○	—	
FFB2H	LCD display control register	LCDC			○	○	—	
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 specify flag register 0L	PR0	PR0L		○	○	○	
FFE9H	Priority specify flag register 0H		PR0H		○	○		
FFEAH	Priority specify flag register 1L	PR1L			○	○	—	
FFF0H	Memory size switching register	IMS			—	○	—	CFH Note 3
FFF4H	Internal expansion RAM size switching register	IXS			—	○	—	0CH Note 4
FFF9H	Watchdog timer mode register	WDTM			○	○	—	00H
FFFAH	Oscillation stabilization time select register	OSTS			—	○	—	04H
FFFBH	Processor clock control register	PCC			○	○	—	

Notes 1. DAM0 is a register that must be set when debugging the μPD780852 with an in-circuit emulator (IE-78K0-NS). Set this register when emulating a power-fail detection function.

2. μPD780851(A), 780852(A) only

3. The initial value of this register is CFH. Set the following value to this register of each model.
μPD780851(A): C8H
μPD780852(A): CAH
μPD78F0852 (to set the same memory map as μPD780851(A)): C8H
μPD78F0852 (to set the same memory map as μPD780852(A)): CAH

4. Although the initial value of this register is 0CH, set this register to 0BH.

3.3 Instruction Address Addressing

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

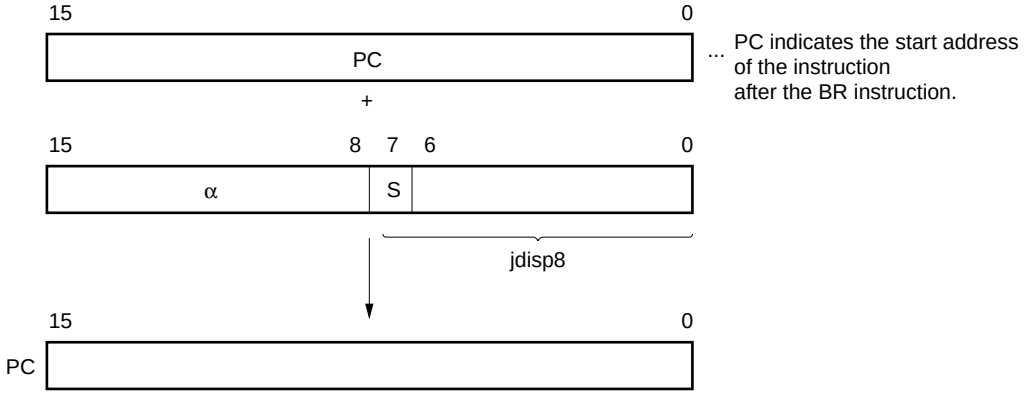
3.3.1 Relative addressing

[Function]

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

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

[Operation]



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

3.3.2 Immediate addressing

[Function]

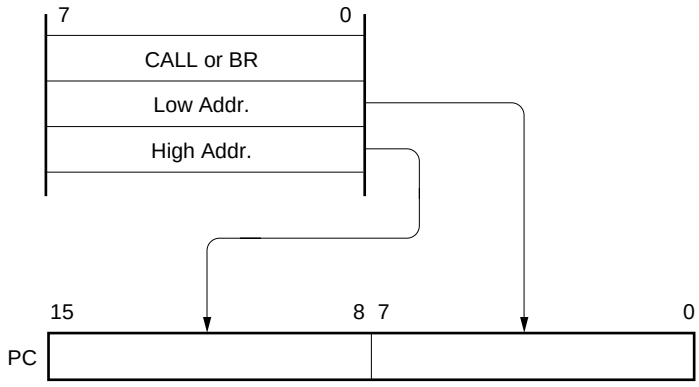
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

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

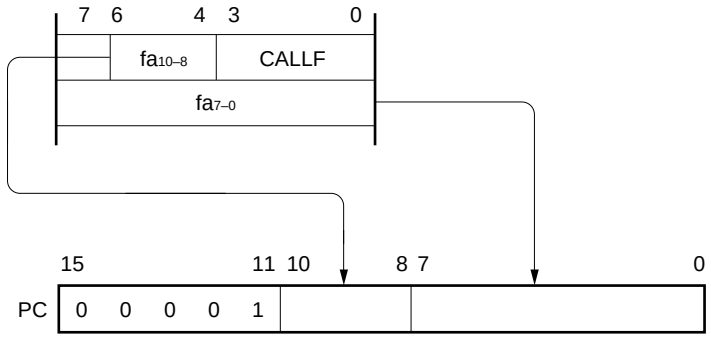
CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space. The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Operation]

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



In the case of CALLF !addr11 instruction



3.3.3 Table indirect addressing

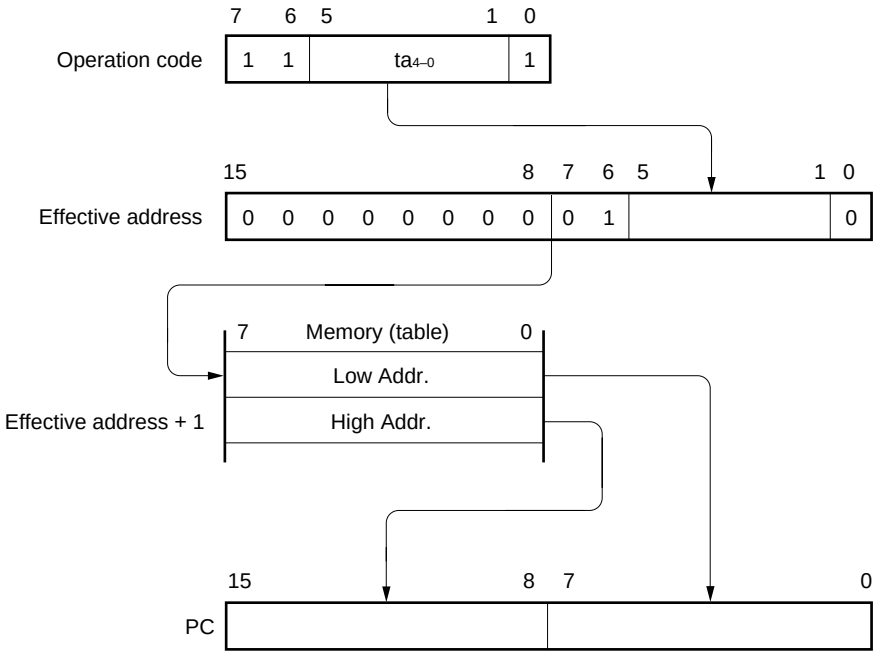
[Function]

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

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

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

[Operation]



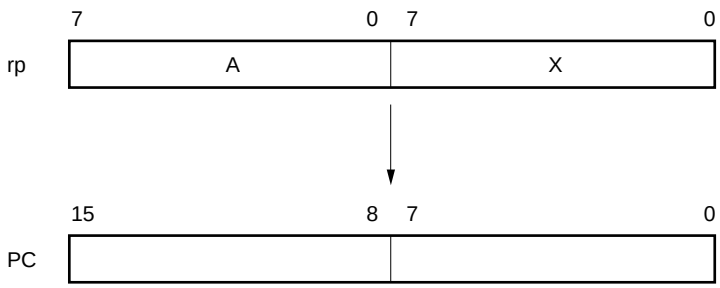
3.3.4 Register addressing

[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.

[Operation]



3.4 Operand Address Addressing

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

3.4.1 Implied addressing

[Function]

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

Of the μPD780852 Subseries instruction words, the following instructions employ implied addressing.

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

[Operand format]

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

[Description example]

In the case of MULU X

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

3.4.2 Register addressing

[Function]

The general register to be specified is accessed as an operand with the register specify code (Rn and RPn) in an operation code and with the register bank select flags (RBS0 and RBS1).

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

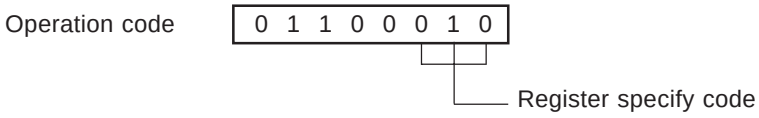
[Operand format]

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

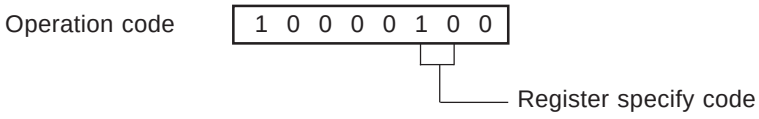
'r' and 'rp' can be described with function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) as well as absolute names (R0 to R7 and RP0 to RP3).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



3.4.3 Direct addressing

[Function]

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

[Operand format]

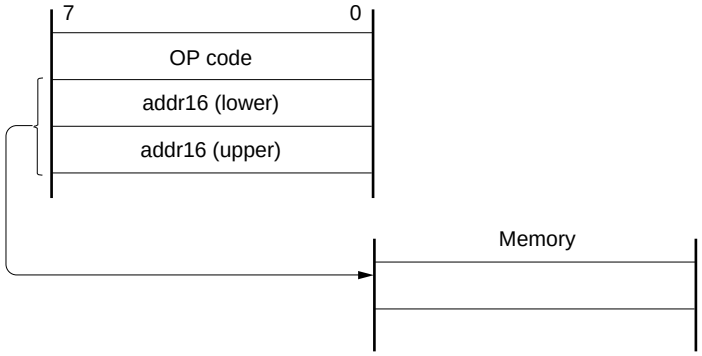
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

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

[Operation]



3.4.4 Short direct addressing

[Function]

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

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

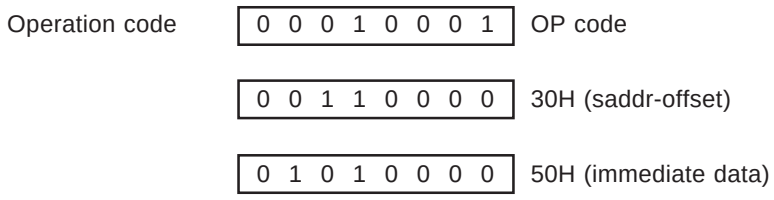
When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is set to 0. When it is at 00H to 1FH, bit 8 is set to 1.

[Operand format]

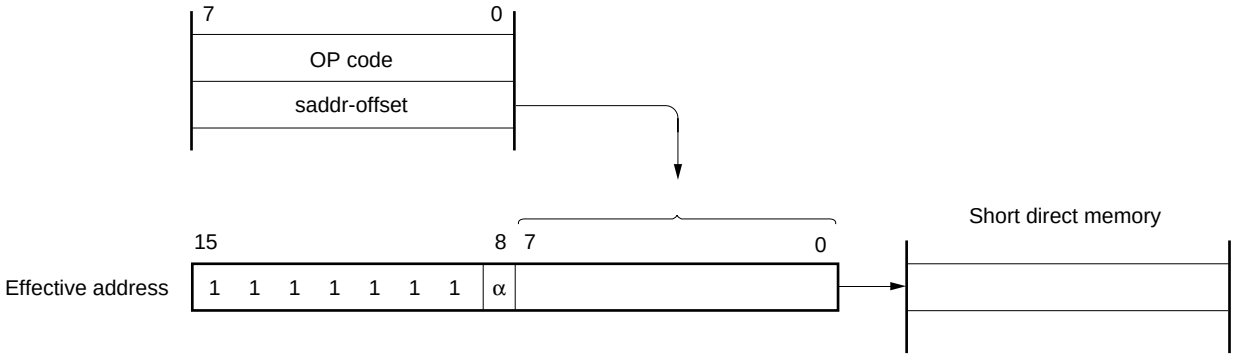
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

[Description example]

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



[Operation]



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

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

3.4.5 Special-function register (SFR) addressing

[Function]

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

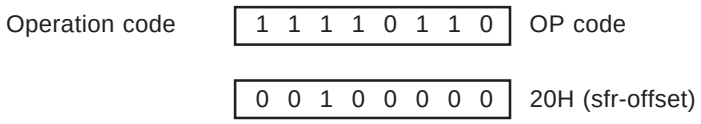
This addressing is applied to the 240-byte spaces FF00H to FFCFH and FFE0H to FFFFH. However, the SFR mapped at FF00H to FF1FH can be accessed with short direct addressing.

[Operand format]

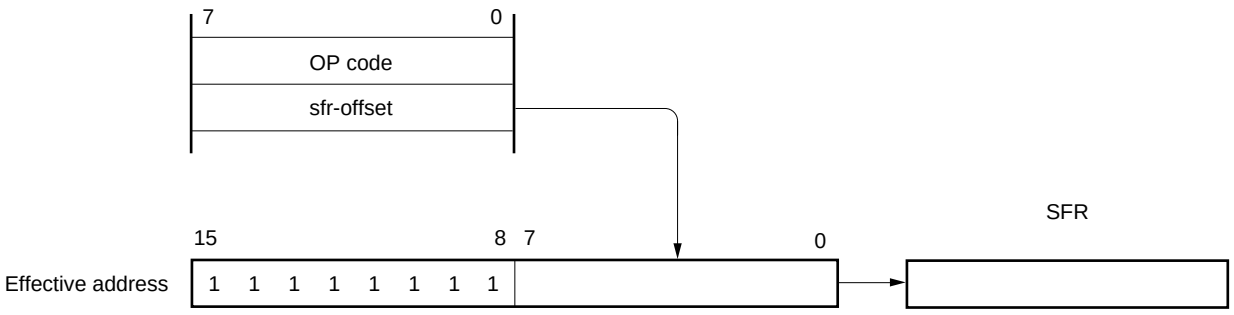
Identifier	Description
sfr	Special-function register name
sfrp	16-bit manipulatable special-function register name (even address only)

[Description example]

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



[Operation]



3.4.6 Register indirect addressing

[Function]

This addressing is to address a memory area to be manipulated by using as an operand address the contents of a register pair specified by the register bank select flags (RBS0 and RBS1) and the register pair specification code in the operation code. This addressing can be carried out for all the memory spaces.

[Operand format]

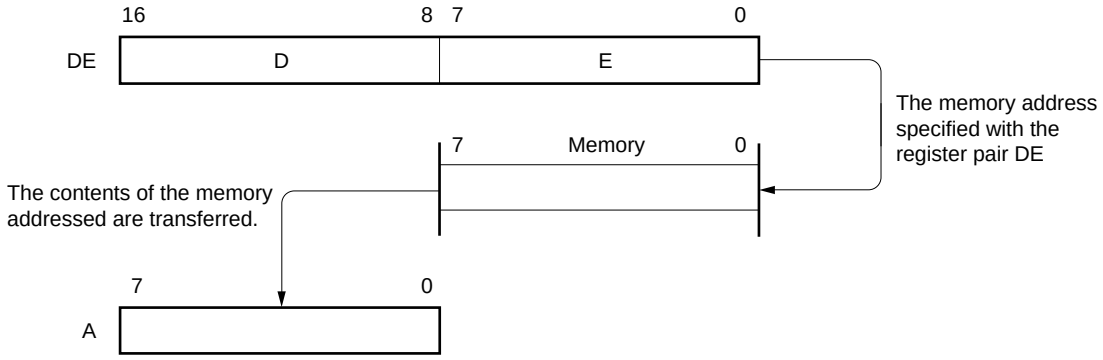
Identifier	Description
—	[DE], [HL]

[Description example]

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

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

[Operation]



3.4.7 Based addressing

[Function]

8-bit immediate data is added as offset data to the contents of the base register, that is, the HL register pair in an instruction word of the register bank specified with the register bank select flags (RBS0 and RBS1) and the sum is used to address the memory. Addition is performed by expanding the offset data as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all the memory spaces.

[Operand format]

Identifier	Description
—	[HL + byte]

[Description example]

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

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

3.4.8 Based indexed addressing

[Function]

The B or C register contents specified in an instruction word are added to the contents of the base register, that is, the HL register pair in an instruction word of the register bank specified with the register bank select flags (RBS0 and RBS1) and the sum is used to address the memory. Addition is performed by expanding the B or C register contents as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all the memory spaces.

[Operand format]

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

[Description example]

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

Operation code	1	0	1	0	1	0	1	1

3.4.9 Stack addressing

[Function]

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

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

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

[Description example]

In the case of PUSH DE

Operation code	1	0	1	1	0	1	0	1

[MEMO]

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The μPD780852 Subseries are provided with five input port pins, sixteen output port pins, and thirty-five input/output port pins. Figure 4-1 shows the port configuration. Every port can be manipulated in 1-bit or 8-bit units controlled in various ways. Moreover, the port pins can also serve as I/O pins of the internal hardware.

Figure 4-1. Port Types

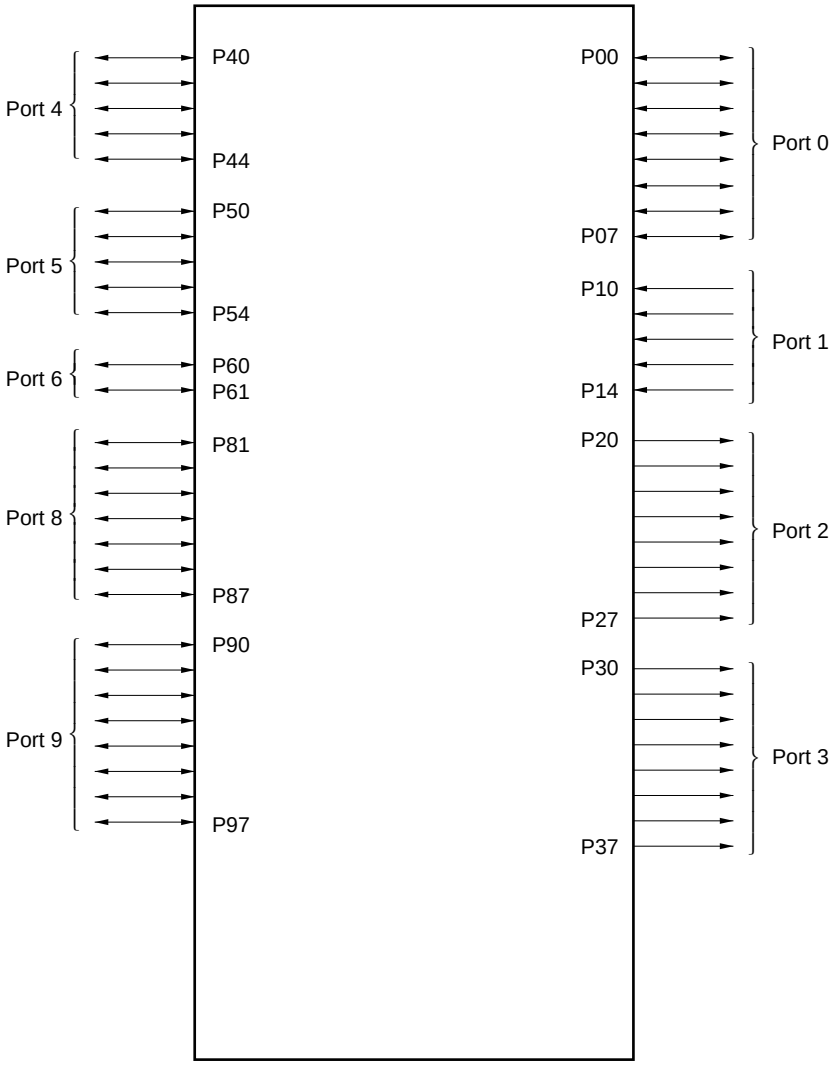


Table 4-1. Port Functions

Pin Name	Input/Output	Function	Alternate Function
P00 to P02	Input/Output	Port 0 8-bit input/output port. Input/output mode can be specified in 1-bit units. On-chip pull-up resistor can be used by software.	INTP0 to INTP2
P03			$\overline{\text{SCK2}}$
P04			SO2
P05			SI2
P06, P07			—
P10 to P14	Input	Port 1 5-bit input only port.	ANI0 to ANI4
P20 to P23	Output	Port 2 8-bit output only port.	SM11 to SM14
P24 to P27			SM21 to SM24
P30 to P33	Output	Port 3 8-bit output only port.	SM31 to SM34
P34 to P37			SM41 to SM44
P40 to P42	Input/Output	Port 4 5-bit input/output port. Input/output mode can be specified in 1-bit units.	TI00 to TI02
P43, P44			TIO2, TIO3
P50	Input/Output	Port 5 5-bit input/output port. Input/output mode can be specified in 1-bit units.	$\overline{\text{SCK3}}$
P51			SO3
P52			SI3
P53			RxD
P54			TxD
P60	Input/Output	Port 6 2-bit input/output port. Input/output mode can be specified in 1-bit units.	PCL/TPO
P61			SGO
P81 to P87	Input/Output	Port 8 7-bit input/output port. Input/output mode can be specified in 1-bit units. Can be set in input/output port or segment output mode in 2-bit units with the LCD display control register (LCDC).	S19 to S13
P90 to P97	Input/Output	Port 9 8-bit input/output port. Input/output mode can be specified in 1-bit units. Can be set in input/output port or segment output mode in 2-bit units with the LCD display control register (LCDC).	S12 to S5

4.2 Port Configuration

A port consists of the following hardware.

Table 4-2. Port Configuration

Item	Configuration
Control register	Port mode register (PMm: m = 0, 2 to 6, 8, 9) Pull-up resistor option register (PU0)
Port	Total: 56 (5 inputs, 16 outputs, 35 inputs/outputs)
Pull-up resistor	Total: 8 (software specifiable: 8)

4.2.1 Port 0

Port 0 is an 8-bit input/output port with output latch. P00 to P07 pins can specify the input mode/output mode in 1-bit units with the port mode register 0 (PM0). On-chip pull-up resistor can be used in 1-bit units with a pull-up resistor option register 0 (PU0).

Alternate functions include external interrupt request input, serial interface data input/output, and clock input/output.

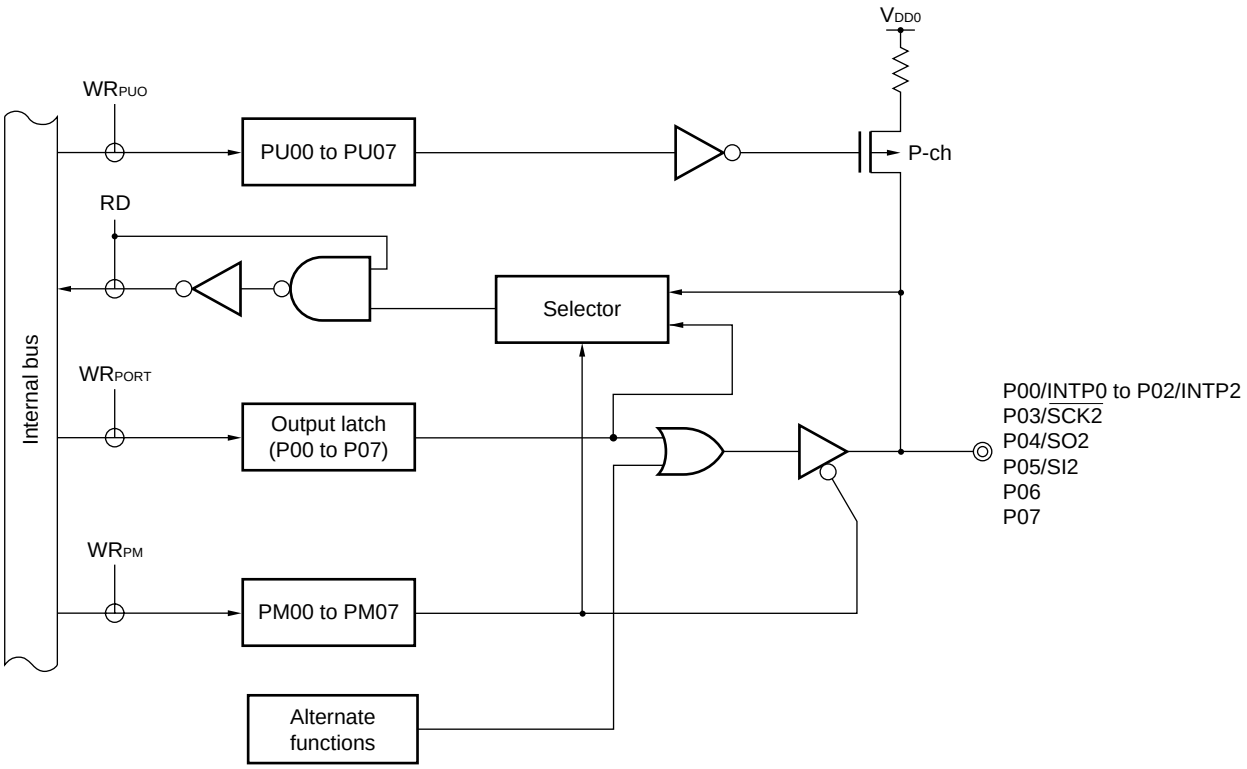
$\overline{\text{RESET}}$ input sets port 0 to input mode.

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

Cautions 1. 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.

2. When port 0 is used as the serial interface pins, an I/O and output latches must be set according to the functions to be used. For an explanation of how to set these latches, refer to the description of the format of the serial operation mode register.

Figure 4-2. P00 to P07 Block Diagram



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

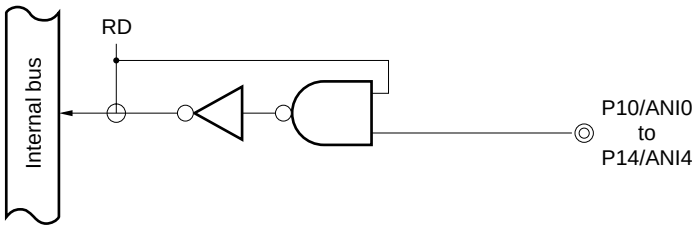
4.2.2 Port 1

Port 1 is a 5-bit input only port.

Alternate functions include an A/D converter analog input.

Figure 4-3 shows a block diagram of port 1.

Figure 4-3. P10 to P14 Block Diagram



RD: Port 1 read signal

4.2.3 Port 2

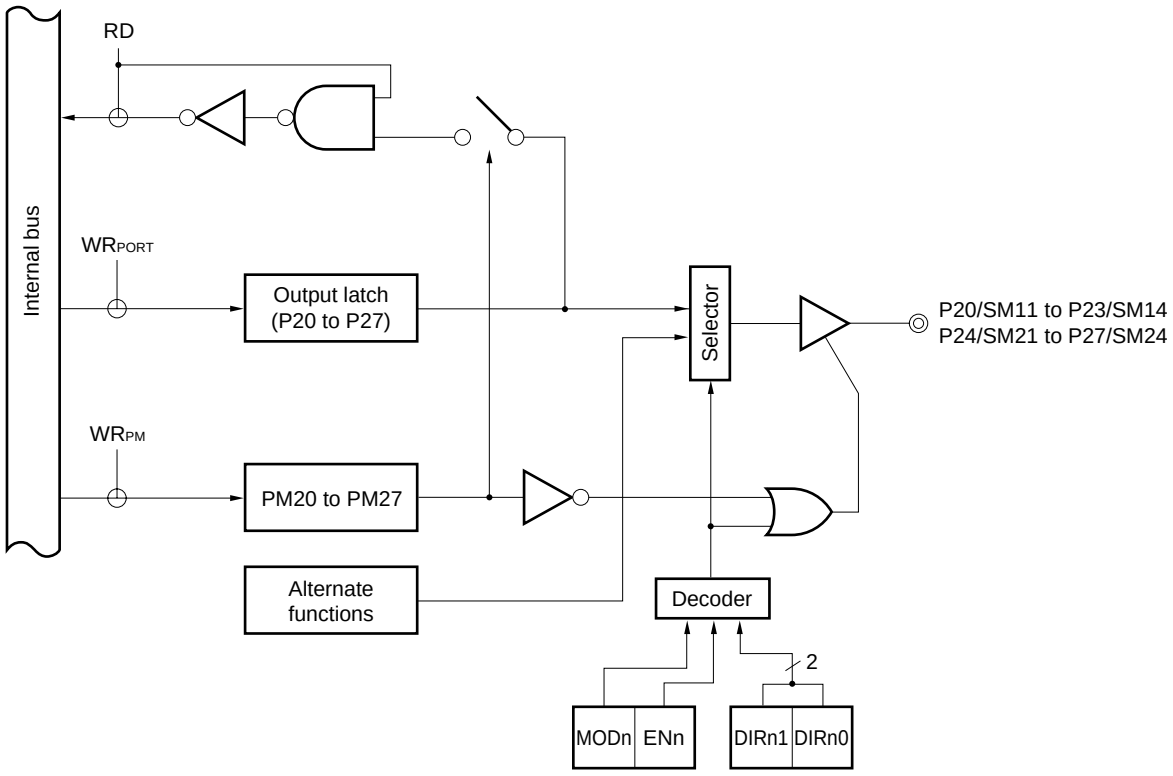
Port 2 is an 8-bit output only port with output latch. P20 to P27 pins go into a high-impedance state when the ENn of port mode control register (PMC) is set to 0 and the port mode register 2 (PM2) is set to 1.

Alternate functions include meter control PWM output.

$\overline{\text{RESET}}$ input sets port 2 to high-impedance state.

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

Figure 4-4. P20 to P27 Block Diagram



PM: Port mode register

RD: Port 2 read signal

WR: Port 2 write signal

Caution When PM2 is set to 0, read operation is enabled. When PM2 is set to 1, read operation is disabled.

Remark n = 1, 2

4.2.4 Port 3

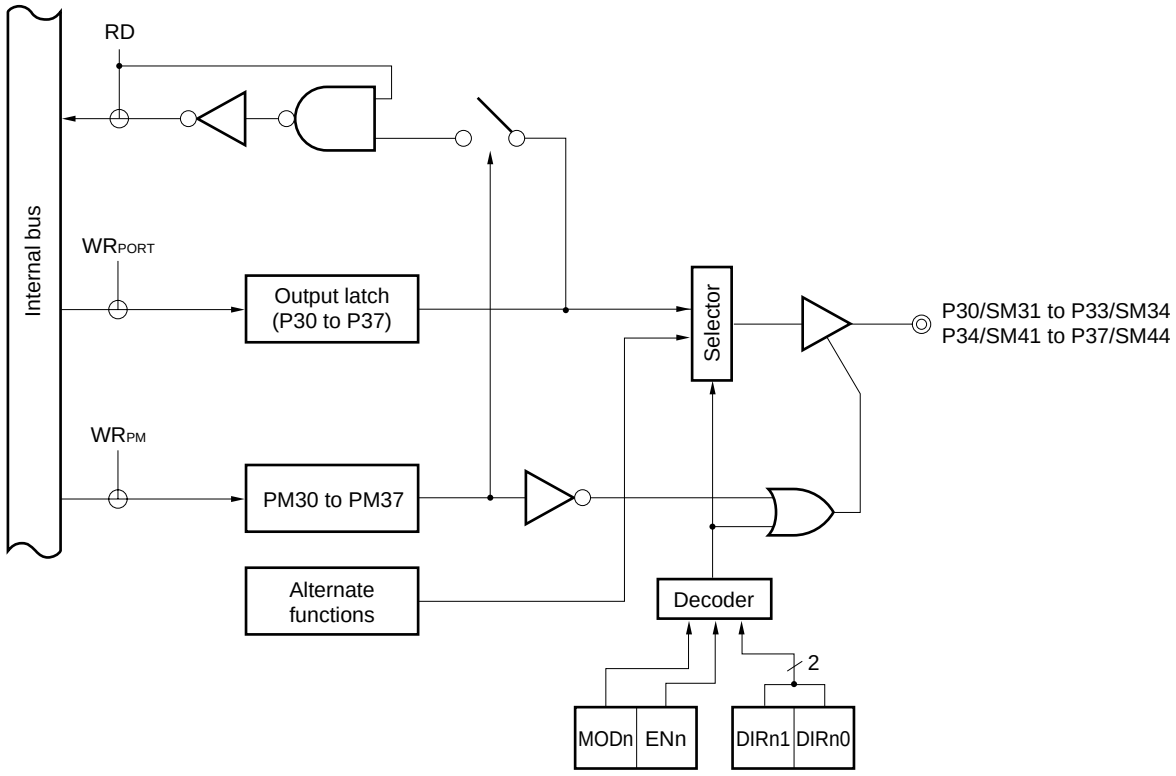
Port 3 is an 8-bit output only port with output latch. P30 to P37 pins go into a high-impedance state when the ENn of port mode control register (PMC) is set to 0 and the port mode register 3 (PM3) is set to 1.

Alternate functions include meter control PWM output.

$\overline{\text{RESET}}$ input sets port 3 to high-impedance state.

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

Figure 4-5. P30 to P37 Block Diagram



PM: Port mode register
RD: Port 3 read signal
WR: Port 3 write signal

Caution When PM3 is set to 0, read operation is enabled. When PM3 is set to 1, read operation is disabled.

Remark n = 3, 4

4.2.5 Port 4

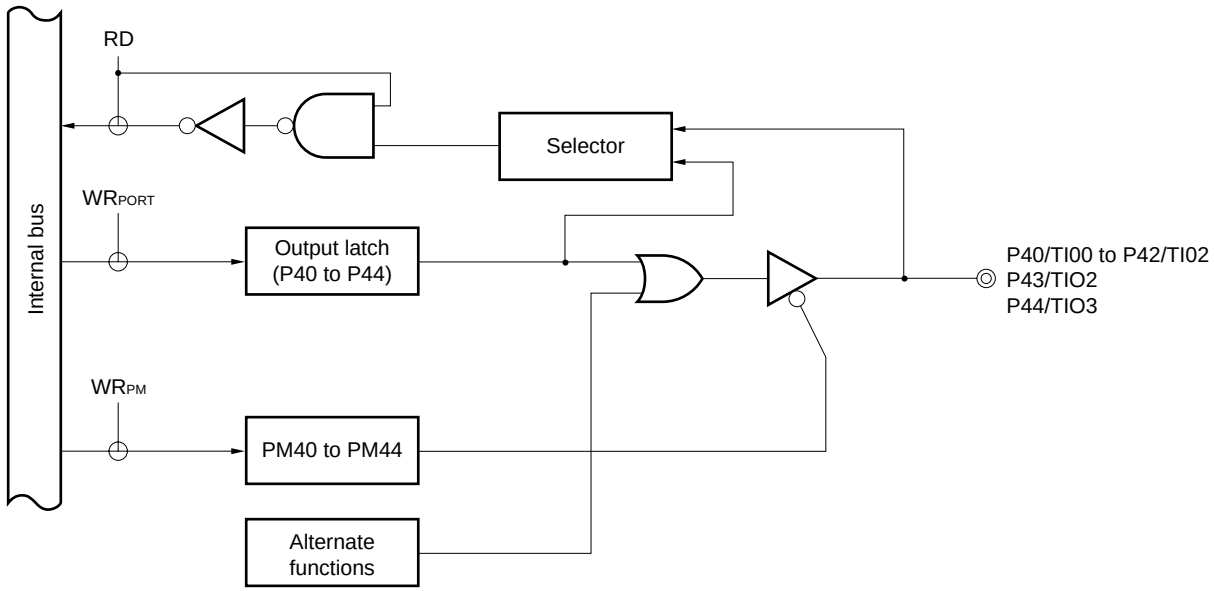
Port 4 is a 5-bit input/output port with output latch. P40 to P44 pins can specify the input mode/output mode in 1-bit units with the port mode register 4 (PM4).

Alternate functions also include timer input/output.

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

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

Figure 4-6. P40 to P44 Block Diagram



PM: Port mode register

RD: Port 4 read signal

WR: Port 4 write signal

4.2.6 Port 5

Port 5 is a 5-bit input/output port with output latch. P50 to P54 pins can specify the input mode/output mode in 1-bit units with the port mode register 5 (PM5).

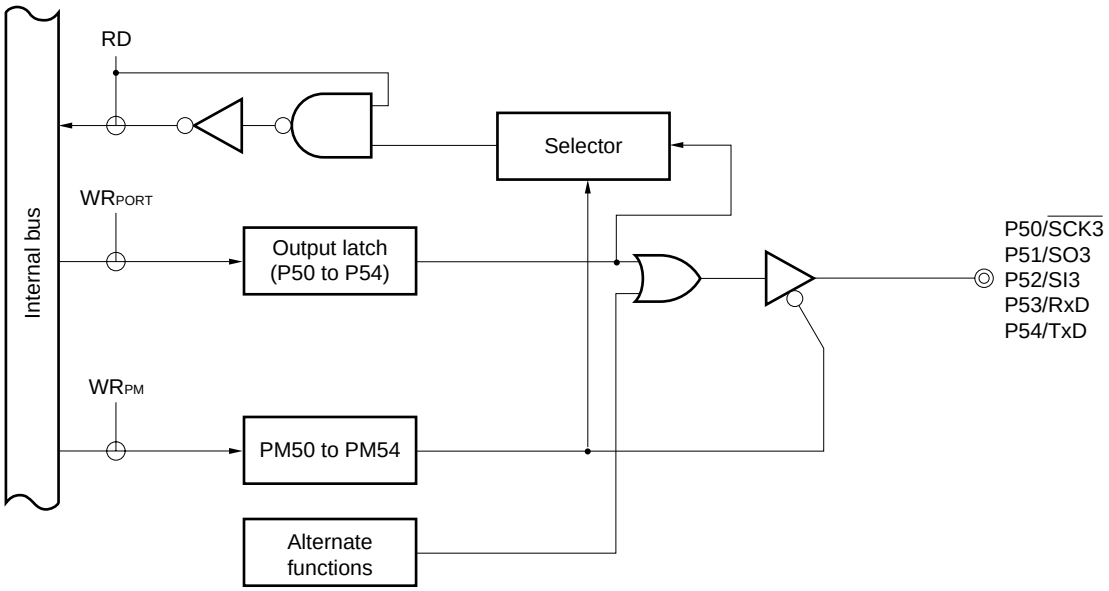
Alternate functions include serial interface data input/output and clock input/output.

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

Figure 4-7 shows a block diagram of port 5.

Caution When port 0 is used as the serial interface pins, an I/O and output latches must be set according to the functions to be used. For an explanation of how to set these latches, refer to the description of the format of the serial operation mode register.

Figure 4-7. P50 to P54 Block Diagram



PM: Port mode register

RD: Port 5 read signal

WR: Port 5 write signal

4.2.7 Port 6

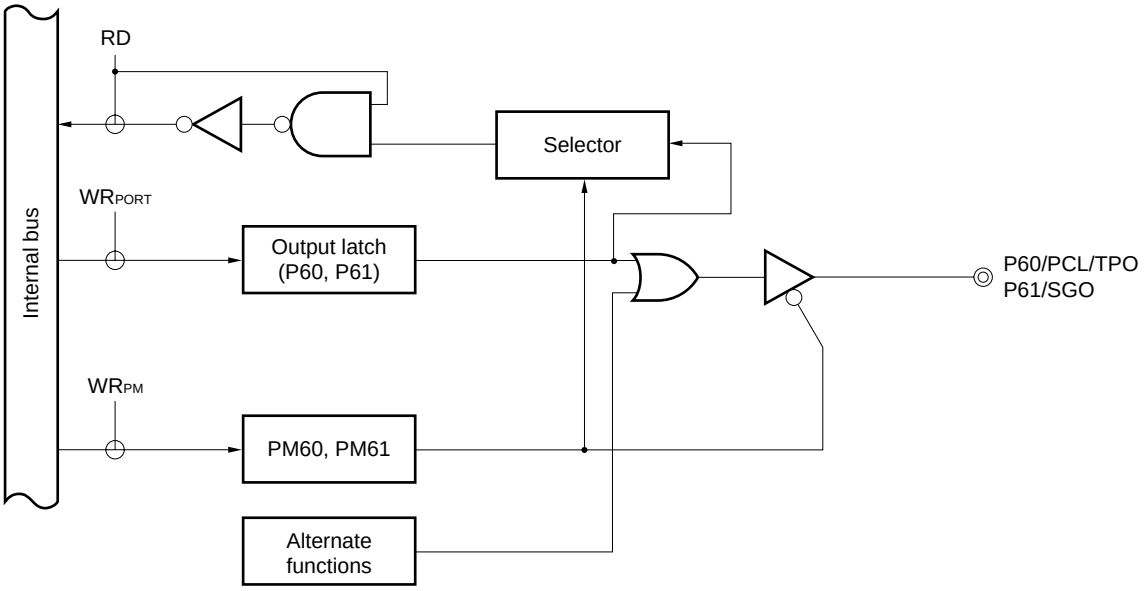
Port 6 is a 2-bit input/output port with output latch. P60 and P61 pins can specify the input mode/output mode in 1-bit units with the port mode register 6 (PM6).

Alternate functions include clock output and sound generator output.

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

Figure 4-8 shows a block diagram of port 6.

Figure 4-8. P60 and P61 Block Diagram



PM: Port mode register

RD: Port 6 read signal

WR: Port 6 write signal

4.2.8 Port 8

Port 8 is a 7-bit input/output port with output latch. P81 to P87 pins can specify the input mode/output mode in 1-bit units with the port mode register 8 (PM8).

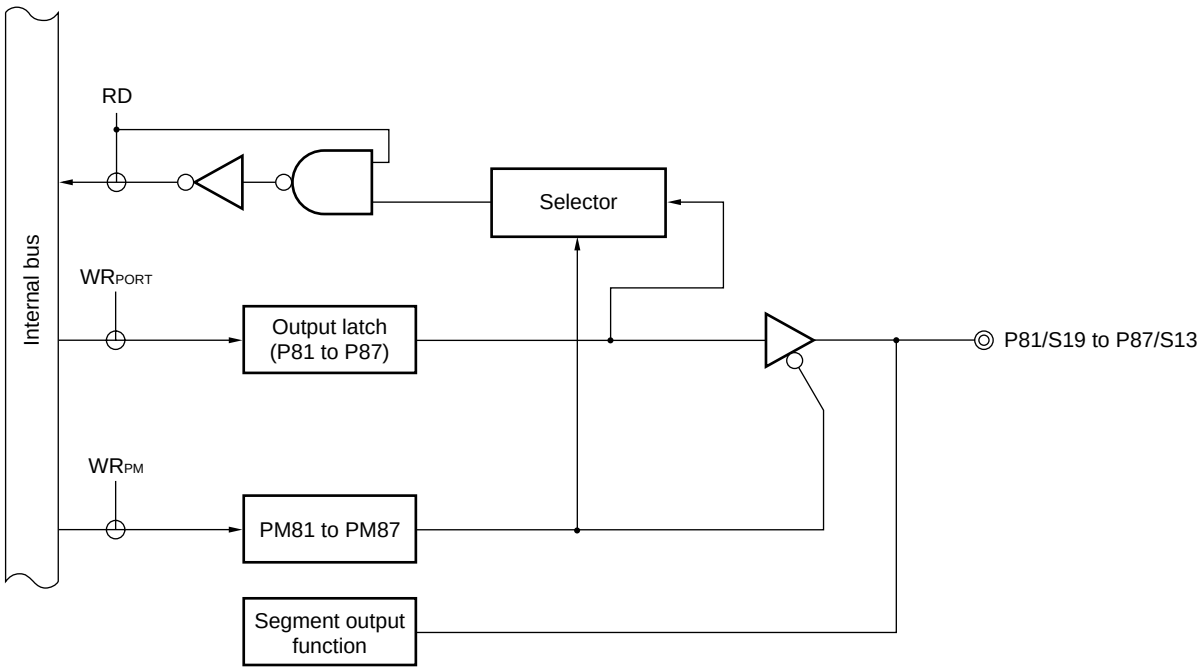
Alternate functions also include segment signal output of the LCD controller/driver.

Segment output and input/output port can be switched by setting the LCD display control register (LCDC).

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

Figure 4-9 shows block diagram of port 8.

Figure 4-9. P81 to P87 Block Diagram



PM: Port mode register

RD: Port 8 read signal

WR: Port 8 write signal

4.2.9 Port 9

Port 9 is an 8-bit input/output port with output latch. P90 to P97 pins can specify the input mode/output mode in 1-bit units with the port mode register 9 (PM9).

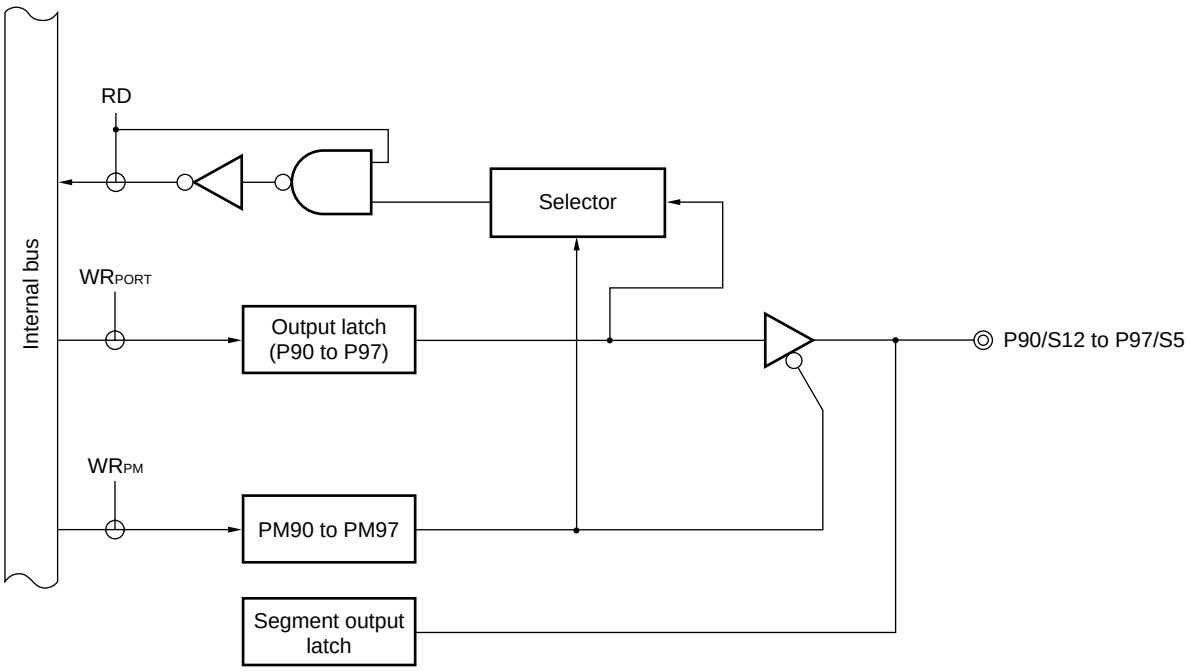
Alternate functions also include segment signal output of the LCD controller/driver.

Segment output and input/output port can be switched by setting the LCD display control register (LCDC).

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

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

Figure 4-10. P90 to P97 Block Diagram



PM: Port mode register

RD: Port 9 read signal

WR: Port 9 write signal

4.3 Port Function Control Registers

The following two types of registers control the ports.

- Port mode registers (PM0, PM2 to PM6, PM8, PM9)
- Pull-up resistor option register (PU0)

(1) Port mode registers (PM0, PM2 to PM6, PM8, PM9)

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

PM0, PM2 to PM6, PM8, and PM9 are independently set with a 1-bit or 8-bit memory manipulation instruction.

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

When a port pin is used as an alternate function pin, set the port mode register and output latch corresponding to the port in accordance with the function to be used.

Cautions 1. Pins P10 to P17 are input-only pins, and pins P20 to P27 and P30 to P37 are output-only pins.

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

3. Ports 2 and 3 that can be also used as meter driving PWM signal output pins go into a high-impedance state when 1 is set to PM2× and PM3×, respectively.

Figure 4-11. Port Mode Register (PM0, PM4 to PM6, PM8, PM9) Format

Address: FF20H After Reset: FFH R/W

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

Address: FF24H After Reset: FFH R/W

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

Address: FF25H After Reset: FFH R/W

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

Address: FF26H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	1	PM61	PM60

Address: FF28H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	1

Address: FF29H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90

PMmn	Pmn Pin Input/Output Mode Selection (m = 0, 4 to 6, 8, 9 ; n = 0 to 7)
0	Output Mode (Output buffer on)
1	Input Mode (Output buffer off)

Figure 4-12. Port Mode Register (PM2, PM3) Format

Address: FF22H After Reset: FFH R/W

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

Address: FF23H After Reset: FFH R/W

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

PMmn	Pmn Pin Input/Output Mode Selection (m = 2, 3 ; n = 0 to 7)
0	Output Mode (Output buffer on)
1	High-impedance state (Output buffer off) **Note**

Note When 0 is set to ENn of port mode control register (PMC)

(2) Pull-up resistor option register (PU0)

This register is used to set whether to use an on-chip pull-up resistor at port 0 or not. By setting the PU0, the on-chip pull-up resistor of the corresponding port pin can be used.

PU0 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Caution When the on-chip pull-up resistor is used, the pull-up resistor is not cut off even when the port is set to the output mode. To use the port in the output mode, clear the corresponding pull-up resistor option register to 0.

Figure 4-13. Pull-Up Resistor Option Register (PU0) Format

Address: FF30H After Reset: 00H R/W

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

PU0n	P0n Pin On-Chip Pull-Up Resistor Selection (n = 0 to 7)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

4.4 Port Function Operations

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

4.4.1 Writing to input/output port

(1) Output mode

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

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

(2) Input mode

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

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

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

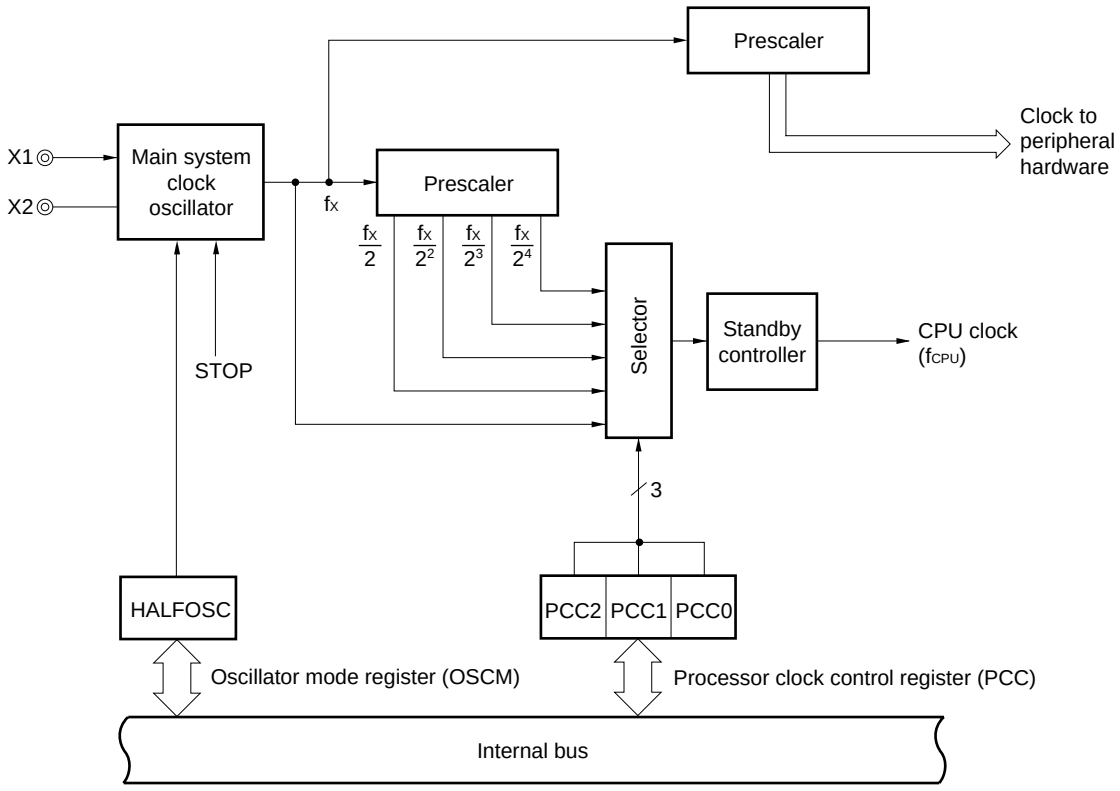
5.1 Clock Generator Functions

The clock generator generates the clock to be supplied to the CPU and peripheral hardware.

- **Main system clock oscillator**
  This circuit oscillates at frequencies of 4.00 to 8.38 MHz. Oscillation can be stopped by executing the STOP instruction.

Figure 5-1 shows clock generator block diagram.

Figure 5-1. Clock Generator Block Diagram



5.2 Clock Generator Configuration

The clock generator consists of the following hardware.

Table 5-1. Clock Generator Configuration

Item	Configuration
Control register	Processor clock control register (PCC) Oscillator mode register (OSCM) **Note**
Oscillator	Main system clock oscillator

Note μPD780851(A), 780852(A) only

5.3 Clock Generator Control Registers

The following two types of registers are used to control the clock generator.

- Processor clock control register (PCC)
- Oscillator mode register (OSCM)

(1) Processor clock control register (PCC)

PCC sets the division ratio of the CPU clock.

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

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

Figure 5-2. Processor Clock Control Register (PCC) Format

Address: FFFBH After Reset: 04H R/W

Symbol	7	6	5	4	3	2	1	0
PCC	0	0	0	0	0	PCC2	PCC1	PCC0

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

Caution Bits 3 to 7 must be set to 0.

Remark f_X: Main system clock oscillation frequency

The fastest instructions of the μPD780852 Subseries are executed in two CPU clocks. Therefore, the relation between the CPU clock (f_{CPU}) and the minimum instruction execution time is as shown in Table 5-2.

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

CPU Clock (f_{CPU})	Minimum Instruction Execution Time: $2/f_{CPU}$	
	f_{CPU} = 8 MHz	f_{CPU} = 8.38 MHz
f_X	0.25 μs	0.24 μs
$f_X/2$	0.5 μs	0.48 μs
$f_X/2^2$	1 μs	0.95 μs
$f_X/2^3$	2 μs	1.91 μs
$f_X/2^4$	4 μs	3.81 μs

Remark f_X: Main system clock oscillation frequency

(2) Oscillator mode register (OSCM)

The μPD780851(A) and μPD780852(A) can be set to the reduced current consumption mode by setting OSCM (only when operated at f_X = 4 to 4.19 MHz).

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

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

Figure 5-3. Oscillator Mode Register (OSCM) Format

Address: FFA0H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OSCM	HALFOSC	0	0	0	0	0	0	0

HALFOSC	Oscillator Mode Selection
0	Normal operation mode
1	Reduced current consumption mode (only when operated at f_X = 4 to 4.19 MHz)

Cautions 1. This function is available only when the device is operated at f_X = 4 to 4.19 MHz. In other cases, be sure not to set 1 to HALFOSC.

2. When using in normal operation mode, setting OSCM is not necessary.

3. Only the first setting of OSCM is effective.

Remark f_X: Main system clock oscillation frequency

5.4 System Clock Oscillator

5.4.1 Main system clock oscillator

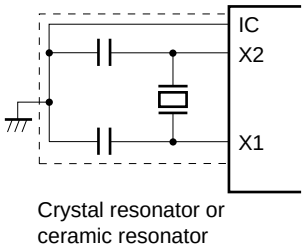
The main system clock oscillator oscillates with a crystal resonator or a ceramic resonator (standard: 8.38 MHz) connected to the X1 and X2 pins.

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

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

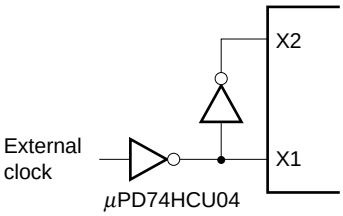
Figure 5-4. External Circuit of Main System Clock Oscillator

(a) Crystal and ceramic oscillation



Crystal resonator or
ceramic resonator

(b) External clock



Cautions 1. Do not execute the STOP instruction while an external clock is input. This is because if the STOP instruction is executed, the main system clock operation is stopped, and the X2 pin is connected to V_{DD}, via a pull-up resistor.

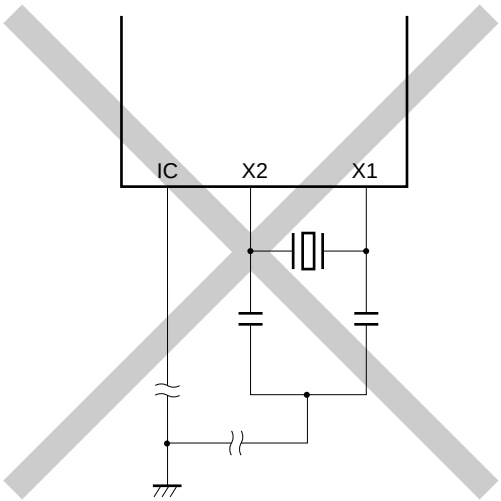
2. When using a main system clock oscillator, carry out wiring in the broken line area in Figure 5-4 as follows to avoid influence of wiring capacity.

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

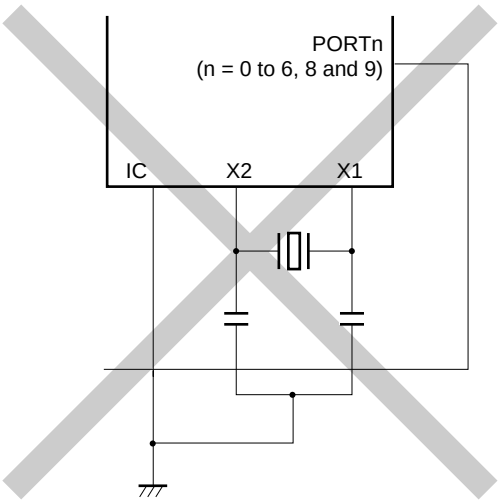
Figure 5-5 shows examples of resonator having bad connection.

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

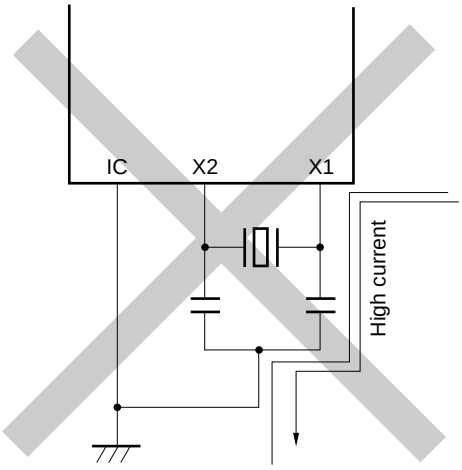
(a) Too long wiring



(b) Crossed signal line



(c) Wiring near high alternating current



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

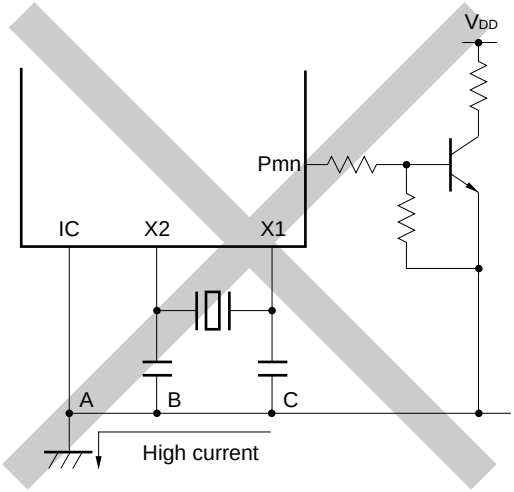
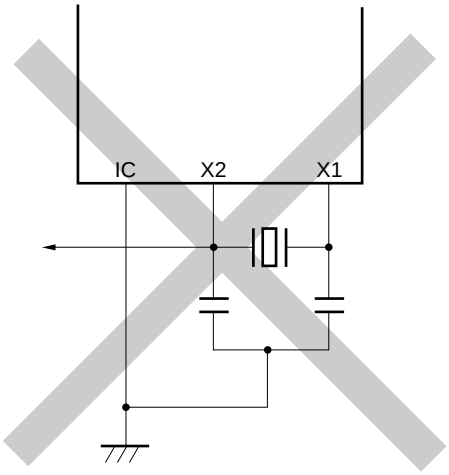


Figure 5-5. Incorrect Examples of Resonator Connection (2/2)

(e) Signals are fetched



5.4.2 Divider circuit

The divider circuit divides the output of the main system clock oscillator (f_X) to generate various clocks.

5.5 Clock Generator Operations

The clock generator generates the following clocks and controls the operation modes of the CPU, such as the standby mode:

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

The operation of the clock generator is determined by the processor clock control register (PCC) and oscillator mode register (OSCM) as follows:

(a) The slowest mode (3.81 μs: at 8.38-MHz operation) of the main system clock is selected when the $\overline{\text{RESET}}$ signal is generated (PCC = 04H). While a low level is input to the $\overline{\text{RESET}}$ pin, oscillation of the main system clock is stopped.

(b) Five types of CPU clocks (0.24 μs, 0.48 μs, 0.95 μs, 1.91 μs, and 3.81 μs: at 8.38-MHz operation) can be selected by the PCC setting.

(c) Two standby modes, STOP and HALT, can be used.

(d) The clock to the peripheral hardware is supplied by dividing the main system clock. The other peripheral hardware is stopped when the main system clock is stopped (except, however, the external clock input operation).

(e) The μPD780851(A) and μPD780852(A) can be set to the reduced current consumption mode by setting OSCM (only when operated at f_X = 4 to 4.19 MHz). Setting 1 to bit 7 (HALFOSC) of OSCM will reduce the power consumption.

Cautions 1. This function is available only when the device is operated at f_X = 4 to 4.19 MHz. In other cases, be sure not to set 1 to HALFOSC.
2. When using in normal operation mode, setting OSCM is not necessary.
3. Only the first setting of OSCM is effective.

5.6 Changing Setting of CPU Clock

5.6.1 Time required for switching CPU clock

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

Actually, the specified clock is not selected immediately after the setting of PCC has been changed, and the old clock is used for the duration of several instructions after that (see **Table 5-3**).

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

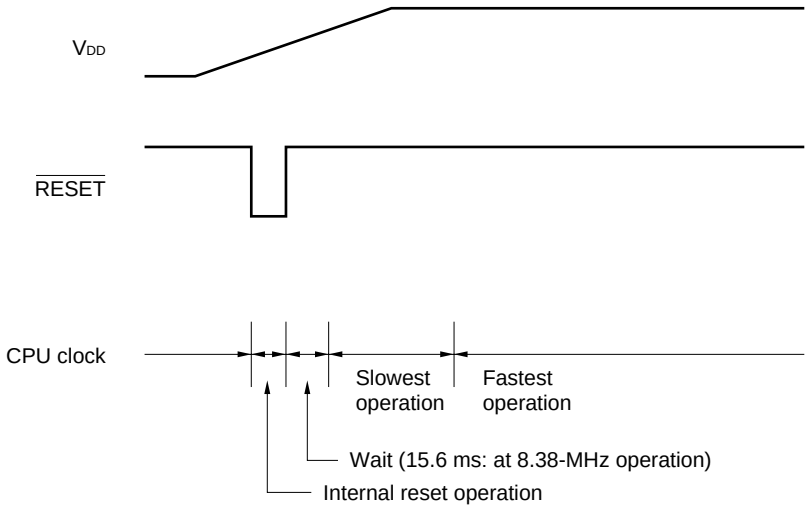
Set Value before Switching			Set Value after Switching														
PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0
			0	0	0	0	0	1	0	1	0	0	1	1	1	0	0
0	0	0				16 instructions			16 instructions			16 instructions			16 instructions		
0	0	1	8 instructions						8 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 of the CPU clock before switching.

5.6.2 Switching CPU clock

The following figure illustrates how the CPU clock switches.

Figure 5-6. Switching CPU Clock



<1> The CPU is reset when the $\overline{\text{RESET}}$ pin is made low on power application. The effect of resetting is released when the $\overline{\text{RESET}}$ pin is later made high, and the main system clock starts oscillating. At this time, the time during which oscillation stabilizes ($2^{17}/f_X$) is automatically secured.
After that, the CPU starts instruction execution at the slowest speed of the main system clock (3.81 μs: at 8.38-MHz operation).

<2> After a lapse of time long enough for the V_{DD} voltage to rise to the level at which the CPU can operate at its maximum speed, rewrite the contents of the processor clock control register (PCC) to execute the maximum-speed operation.

[MEMO]

CHAPTER 6 16-BIT TIMER 0 TM0

6.1 Outline of Internal Timer of μPD780852 Subseries

This chapter explains the 16-bit timer 0. Before that, the internal timers of the μPD780852 Subseries, and the related functions are briefly explained below.

(1) 16-bit timer 0 TM0

The TM0 can be used for pulse widths measurement, divided output of input pulse.

(2) 8-bit timer 1 TM1

The TM1 can be used for an interval timer (see **CHAPTER 7 8-BIT TIMER 1 TM1**).

(3) 8-bit timer/event counters 2 TM2 and 3 TM3

TM2 and TM3 can be used to serve as an interval timer and an external event counter and to output square waves with any selected frequency and PWM (see **CHAPTER 8 8-BIT TIMER/EVENT COUNTERS 2 TM2 AND 3 TM3**).

(4) Watch timer

This timer can set a flag every 0.5 sec. and simultaneously generates interrupt request at the preset time intervals (see **CHAPTER 9 WATCH TIMER**).

(5) Watchdog timer

This timer can perform the watchdog timer function or generate non-maskable interrupt request, maskable interrupt request and $\overline{\text{RESET}}$ at the preset time intervals (see **CHAPTER 10 WATCHDOG TIMER**).

(6) Clock output controller

Clock output supplies other devices with the divided main system clock (see **CHAPTER 11 CLOCK OUTPUT CONTROLLER**).

Table 6-1. Timer/Event Counter Operations

		16-Bit Timer TM0	8-Bit Timer TM1	8-Bit Timer/Event Counter TM2, TM3	Watch Timer	Watchdog Timer
Operating mode	Interval timer	2 channels	1 channel	2 channels	1 channel Note 1	1 channel Note 2
	External event counter	–	–	○	–	–
Function	Timer output	–	–	○	–	–
	PWM output	–	–	○	–	–
	Pulse width measurement	○	–	–	–	–
	Square-wave output	–	–	○	–	–
	Divided output	○	–	–	–	–
	Interrupt request	○	○	○	○	○

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

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

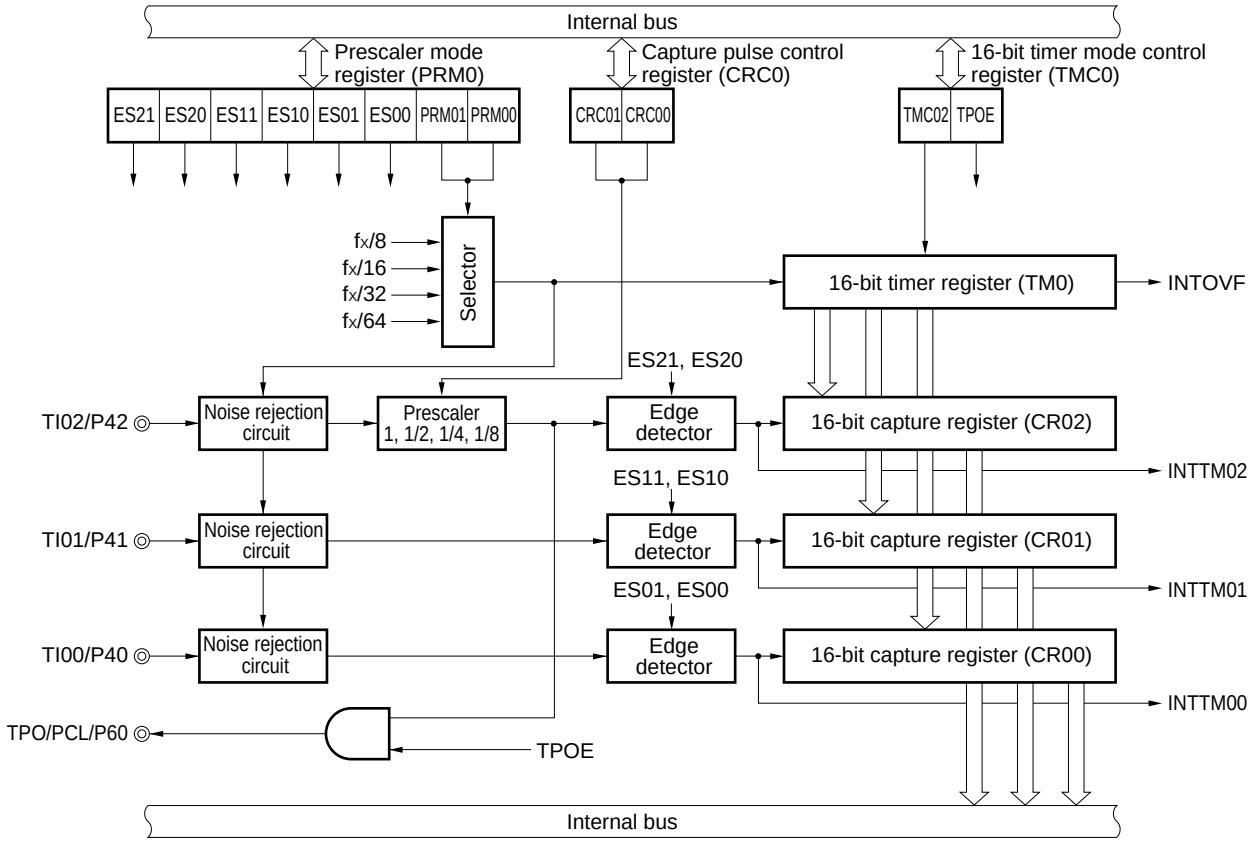
6.2 16-Bit Timer 0 TM0 Functions

The 16-bit timer 0 TM0 has the following functions.

- Pulse width measurement
- Divided output of input pulse

Figure 6-1 shows 16-bit timer 0 TM0 block diagram.

Figure 6-1. Timer 0 TM0 Block Diagram



(1) Pulse width measurement

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

(2) Divided output of input pulse

The frequency of an input signal can be divided and the divided signal can be output.

6.3 16-Bit Timer 0 TM0 Configuration

16-bit timer 0 TM0 consists of the following hardware.

Table 6-2. 16-Bit Timer 0 TM0 Configuration

Item	Configuration
Timer register	16 bits × 1 (TM0)
Register	Capture register: 16 bits × 3 (CR00 to CR02)
Control register	16-bit timer mode control register (TMC0) Capture pulse control register (CRC0) Prescaler mode register (PRM0) Port mode register 4 (PM4)

(1) 16-bit timer register 0 (TM0)

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

The counter is incremented in synchronization with the rising edge of an input clock. If the count value is read during operation, the count value at that point is read. The value of the counter continues to be incremented even while the count value is being read.

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

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

<2> Clear TMC02

(2) Capture register 00 (CR00)

The valid edge of the TI00 pin can be selected as the capture trigger. Setting of the TI00 valid edge is performed with the prescaler mode register (PRM0). When the valid edge of the TI00 is detected, an interrupt request (INTTM00) is generated.

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

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

(3) Capture register 01 (CR01)

The valid edge of the TI01 pin can be selected as the capture trigger. Setting of the TI01 valid edge is performed with the prescaler mode register (PRM0). When the valid edge of the TI01 is detected, an interrupt request (INTTM01) is generated.

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

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

(4) Capture register 02 (CR02)

The valid edge of the TI02 pin can be selected as the capture trigger. Setting of the TI02 valid edge is performed with the prescaler mode register (PRM0). When the valid edge of the TI02 is detected, an interrupt request (INTTM02) is generated.

CR02 is set with a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes CR02 to undefined.

6.4 16-Bit Timer 0 TM0 Control Registers

The following four types of registers are used to control 16-bit timer 0 TM0.

- 16-bit timer mode control register (TMC0)
- Capture pulse control register (CRC0)
- Prescaler mode register (PRM0)
- Port mode register 4 (PM4)

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

This register sets the 16-bit timer operation mode and controls the prescaler output signals.

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

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

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

Address: FF72H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TMC0	0	0	0	0	0	TMC02	0	TPOE

TMC02	TM0 Operation Mode Selection
0	Operation stop (TM0 cleared to 0)
1	Operation enabled

TPOE	Prescaler Output Control
0	Prescaler signal output disabled
1	Prescaler signal output enabled

Cautions 1. Before changing the operation mode, stop the timer operation (by setting 0 to TMC02).

2. Bits 1 and 3 to 7 must be set to 0.

(2) Capture pulse control register (CRC0)

This register specifies the division ratio of the capture pulse input to the 16-bit capture register (CR02) from an external source.

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

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

Figure 6-3. Capture Pulse Control Register (CRC0) Format

Address: FF71H After Reset: 00H R/W

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

CRC01	CRC00	Capture Pulse Selection
0	0	Does not divide capture pulse
0	1	Divides capture pulse by 2
1	0	Divides capture pulse by 4
1	1	Divides capture pulse by 8

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

2. Bits 2 to 7 must be set to 0.

(3) Prescaler mode register (PRM0)

This register is used to set TM0 count clock and valid edge of TI00 to TI02 input.

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

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

Figure 6-4. Prescaler Mode Register (PRM0) Format

Address: FF70H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM0	ES21	ES20	ES11	ES10	ES01	ES00	PRM01	PRM00

ESn1	ESn0	TI0n Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM01	PRM00	Count Clock Selection
0	0	$f_X/2^3$
0	1	$f_X/2^4$
1	0	$f_X/2^5$
1	1	$f_X/2^6$

Caution Timer operation must be stopped before setting PRM0.

Remarks 1. f_X: Main system clock oscillation frequency.

2. n = 0 to 2

(4) Port mode register 4 (PM4)

This register sets port 4 to the input or output mode in 1-bit units.

★ To use the P40/TI00 to P42/TI02 pins as timer input pins, set PM40 to PM42 to 1. PM4 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM4 to FFH.

Figure 6-5. Port Mode Register 4 (PM4) Format

Address: FF24H After Reset: FFH R/W

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

PM4n	P4n Pin I/O Mode Selection (n = 0 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

6.5 16-Bit Timer 0 TM0 Operations

6.5.1 Pulse width measurement operations

It is possible to measure the pulse width of the signals input to the TI00/P40 to TI02/P42 pins using the 16-bit timer register (TM0). TM0 is used in free-running mode.

(1) Pulse width measurement with free-running counter and one capture register (TI00)

When the edge specified by prescaler mode register (PRM0) is input to the TI00/P40 pin, the value of TM0 is taken into 16-bit capture register 00 (CR00) and an external interrupt request signal (INTTM00) is set.

Any of three edge specifications can be selected—rising, falling, or both edges—by means of bits 2 and 3 (ES00 and ES01) of prescaler mode register (PRM0).

For TI00 pin valid edge detection, sampling is performed at the count clock selected by PRM0, and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

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

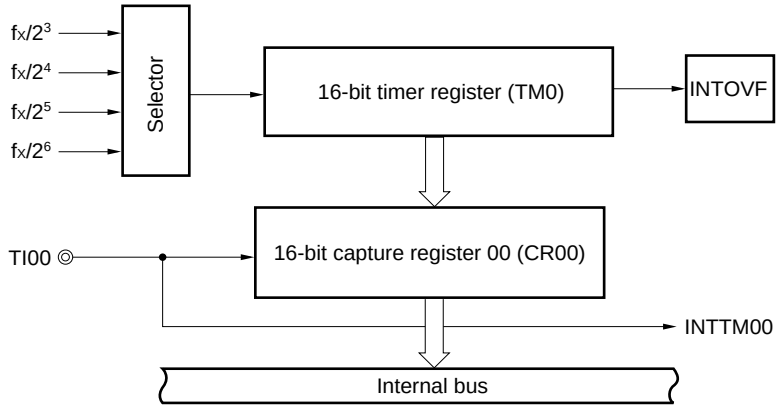
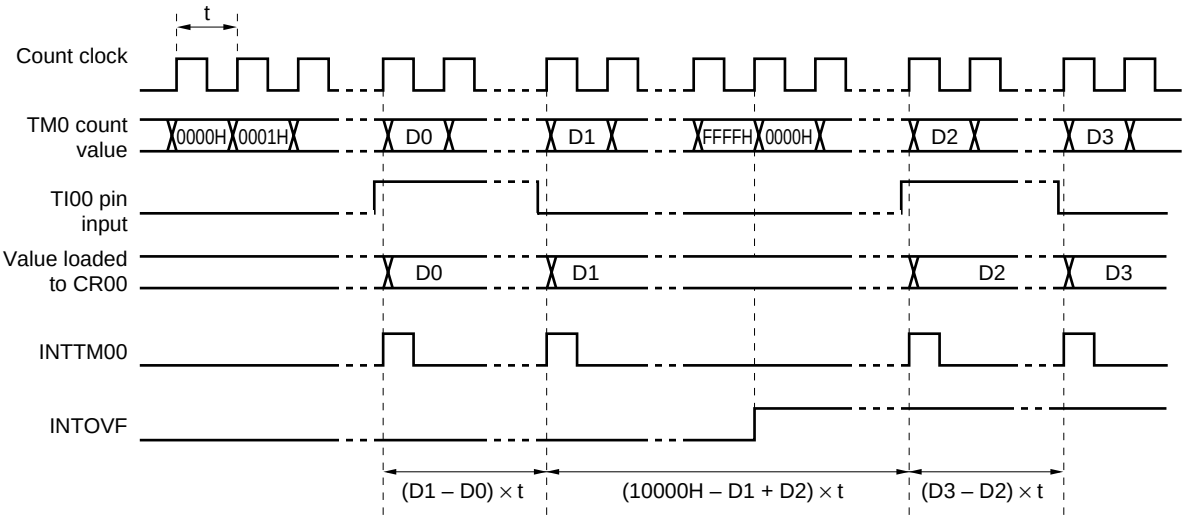


Figure 6-7. Pulse Width Measurement Operation Timing by Free-Running Counter and One Capture Register (with Both Edges Specified)



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

The 16-bit timer register (TM0) allows simultaneous measurement of the pulse widths of the three signals input to the TI00/P40 to TI02/P42 pins.

When the edge specified by bits 2 and 3 (ES00 and ES01) of prescaler mode register (PRM0) is input to the TI00/P40 pin, the value of TM0 is taken into 16-bit capture register 00 (CR00) and an external interrupt request signal (INTTM00) is set.

Also, when the edge specified by bits 4 and 5 (ES10 and ES11) of PRM0 is input to the TI01/P41 pin, the value of TM0 is taken into 16-bit capture register 01 (CR01) and an external interrupt request signal (INTTM01) is set.

When the edge specified by bits 6 and 7 (ES20 and ES21) of PRM0 is input to the TI02/P42 pin, the value of TM0 is taken into 16-bit capture register 02 (CR02) and external interrupt request signal (INTTM02) is set.

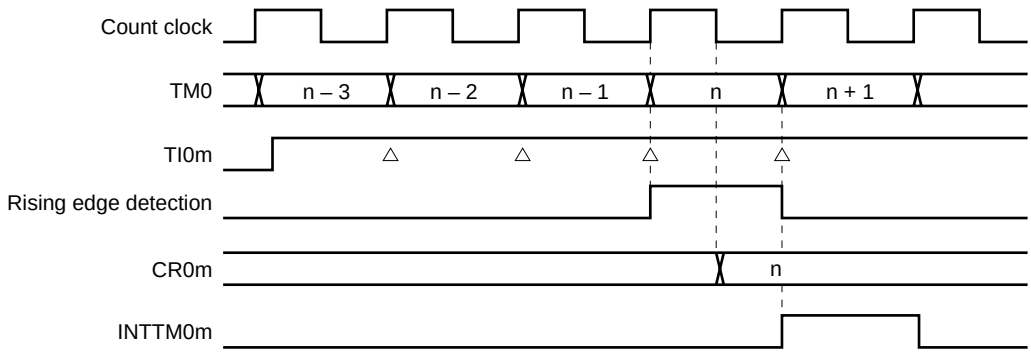
Any of three edge specifications can be selected—rising, falling, or both edges—as the valid edges for the TI00/P40 to TI02/P42 pins by means of bits 2 and 3 (ES00 and ES01), bits 4 and 5 (ES10 and ES11), and bits 6 and 7 (ES20 and ES21) of PRM0, respectively.

For TI00/P40 to TI02/P42 pins valid edge detection, sampling is performed at the interval selected by means of PRM0, and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

- **Capture operation (free-running mode)**

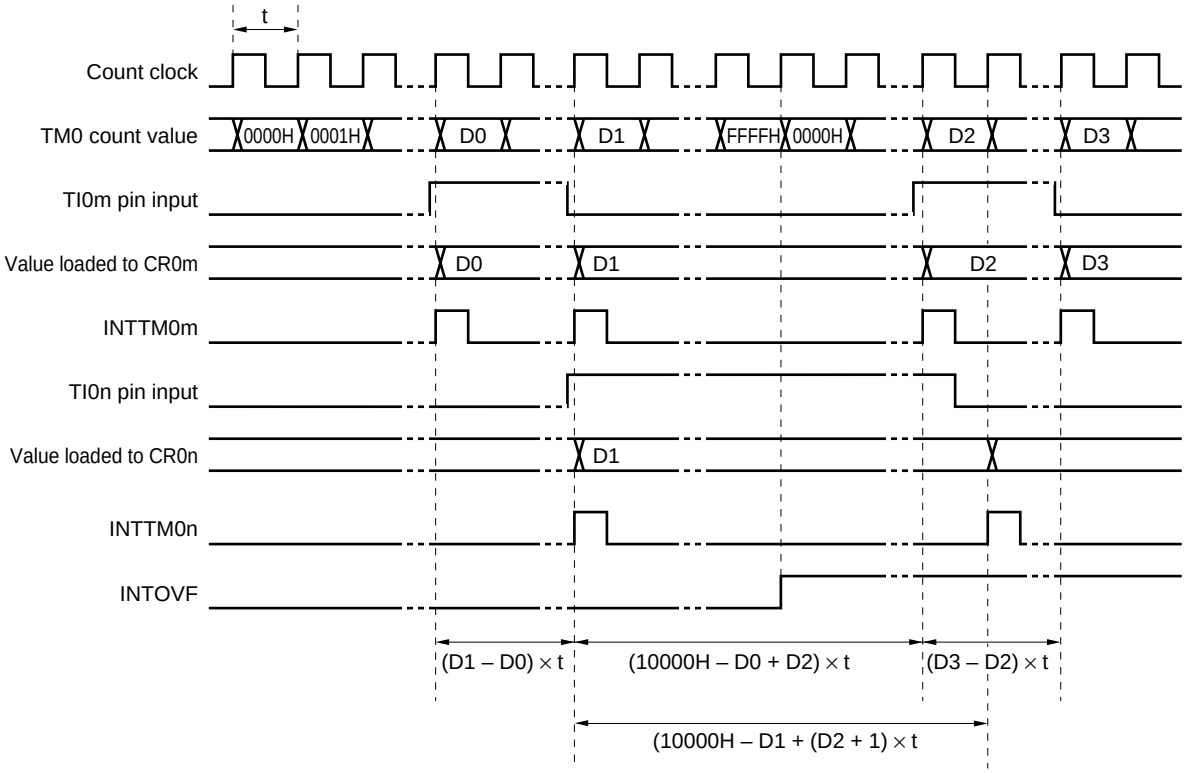
  Capture register operation in capture trigger input is shown.

Figure 6-8. CR0m Capture Operation with Rising Edge Specified



Remark m = 0 to 2

Figure 6-9. Pulse Width Measurement Operation Timing by Free-Running Counter (with Both Edges Specified)



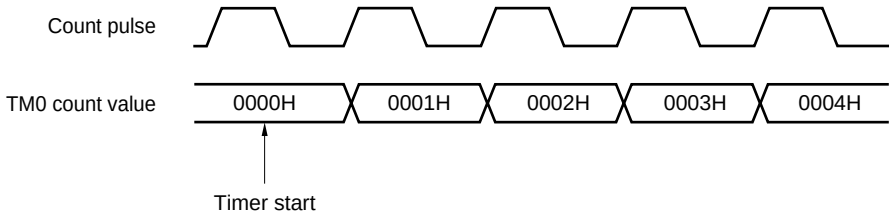
Remark m = 0 to 2, n = 1, 2

6.6 16-Bit Timer 0 TM0 Cautions

(1) Timer start errors

An error with a maximum of one clock may occur until counting is started after timer start. This is because the 16-bit timer register (TM0) is started asynchronously with the count pulse.

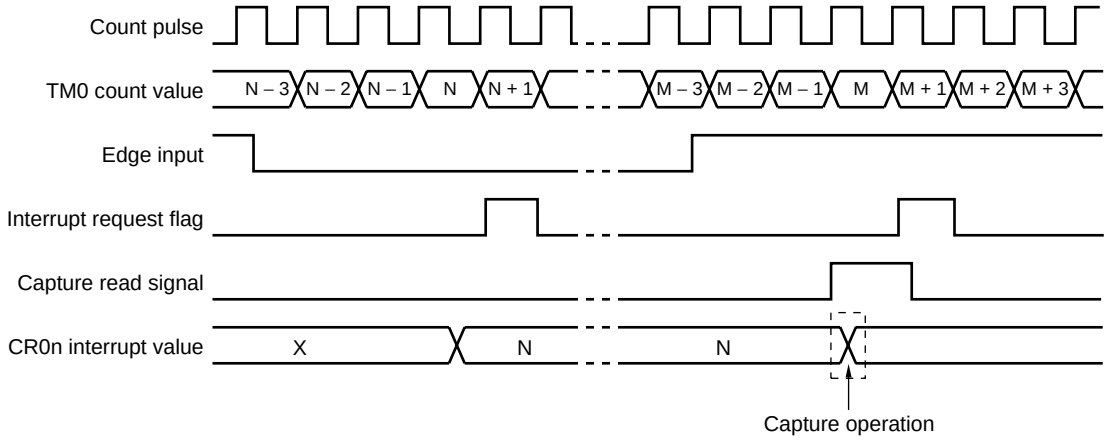
Figure 6-10. 16-Bit Timer Register Start Timing



(2) Capture register data retention timings

If the valid edge of the TI0n/P4n pin is input during 16-bit capture register 0n (CR0n) read, CR0n performs capture operation, but the capture value is not guaranteed. However, the interrupt request flag (INTTM0n) is set upon detection of the valid edge.

★ **Figure 6-11. Capture Register Data Retention Timing**



Remark n = 0 to 2

(3) Valid edge setting

Set the valid edge of the TI0n/P4n pin after setting bit 2 (TMC02) of the 16-bit timer mode control register (TMC0) to 0, and then stopping timer operation. Valid edge setting is carried out with bits 2 to 7 (ESn0 and ESn1) of the prescaler mode register (PRM0).

Remark n = 0 to 2

(4) Occurrence of INTTM0n

INTTM0n occurs even if no capture pulse exists, immediately after the timer operation has been started (TMC02 of TMC0 has been set to 1) with a high level applied to input pins TI00 to TI02 of 16-bit timer 0, and with the rising edge (with ESn1 and ESn0 of PRM0 set to 0, 1), or both the rising and falling edges (with ESn1 and ESn0 of PRM0 set to 1, 1) selected. However, INTTM0n does not occur if a low level is applied to TI00 to TI02.

Remark n = 0 to 2

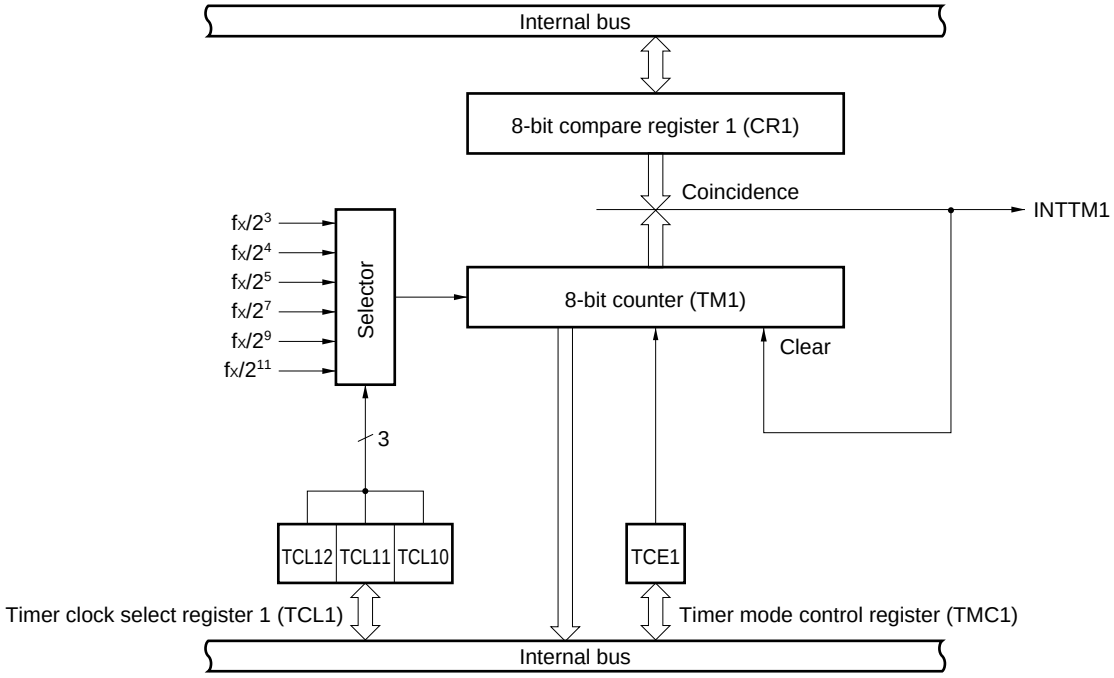
CHAPTER 7 8-BIT TIMER 1 TM1

7.1 8-Bit Timer 1 TM1 Functions

The 8-bit timer 1 TM1 operates as an 8-bit interval timer.

Figure 7-1 shows timer 1 TM1 block diagram.

Figure 7-1. 8-Bit Timer 1 TM1 Block Diagram



7.2 8-Bit Timer 1 TM1 Configuration

8-bit timer 1 TM1 consists of the following hardware.

Table 7-1. 8-Bit Timer 1 TM1 Configuration

Item	Configuration
Timer register	8-bit counter 1 (TM1)
Register	8-bit compare register 1 (CR1)
Control register	Timer clock select register 1 (TCL1) 8-bit timer mode control register 1 (TMC1)

(1) 8-bit counter 1 (TM1)

TM1 is an 8-bit read-only register which counts the count pulses.

The counter is incremented in synchronization with the rising edge of the count clock. When count value is read during operation, count clock input is temporary stopped, and then the count value is read. In the following situations, the count value is set to 00H.

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

<2> Clear TCE1

<3> Match between TM1 and CR1

(2) 8-bit compare register 1 (CR1)

The value set in the CR1 is constantly compared with the 8-bit counter 1 (TM1) count value, and an interrupt request (INTTM1) is generated if they match.

It is possible to rewrite the value of CR1 within 00H to FFH during count operation.

7.3 8-Bit Timer 1 TM1 Control Registers

The following two types of registers are used to control 8-bit timer 1 TM1.

- Timer clock select register 1 (TCL1)
- 8-bit timer mode control register 1 (TMC1)

(1) Timer clock select register 1 (TCL1)

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

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

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

Figure 7-2. Timer Clock Select Register 1 (TCL1) Format

Address: FF73H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL1	0	0	0	0	0	TCL12	TCL11	TCL10

TCL12	TCL11	TCL10	Count Clock Selection
0	1	0	$f_X/2^3$ (1.04 MHz)
0	1	1	$f_X/2^4$ (523 kHz)
1	0	0	$f_X/2^5$ (261 kHz)
1	0	1	$f_X/2^7$ (65.4 kHz)
1	1	0	$f_X/2^9$ (16.3 kHz)
1	1	1	$f_X/2^{11}$ (4.09 kHz)
Other than above			Setting prohibited

Cautions 1. When rewriting TCL1 to other data, stop the timer operation beforehand.

2. Bits 3 to 7 must be set to 0.

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses apply to operation with f_X = 8.38 MHz

(2) 8-bit timer mode control register 1 (TMC1)

TMC1 is a register that controls the counting operation of the 8-bit counter 1 (TM1).

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

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

Figure 7-3. 8-Bit Timer Mode Control Register 1 (TMC1) Format

Address: FF76H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TMC1	TCE1	0	0	0	0	1	0	0

TCE1	TM1 Count Operation Control
0	After clearing counter to 0, count operation disabled
1	Count operation start

Caution Bits 0, 1, and 3 to 6 must be set to 0, and bit 2 must be set to 1.

7.4 8-Bit Timer 1 TM1 Operations

7.4.1 8-bit interval timer operation

The 8-bit timer 1 operates as an interval timer which generates interrupt requests repeatedly at intervals of the count value preset to 8-bit compare register 1 (CR1).

When the count values of the 8-bit counter 1 (TM1) match the values set to CR1, counting continues with the TM1 values cleared to 0 and the interrupt request signal (INTTM1) is generated.

Count clock of the TM1 can be selected with bits 0 to 2 (TCL10 to TCL12) of the timer clock select register 1 (TCL1).

[Setting]

<1> Set the registers.
- TCL1: Select count clock.
- CR1: Compare value

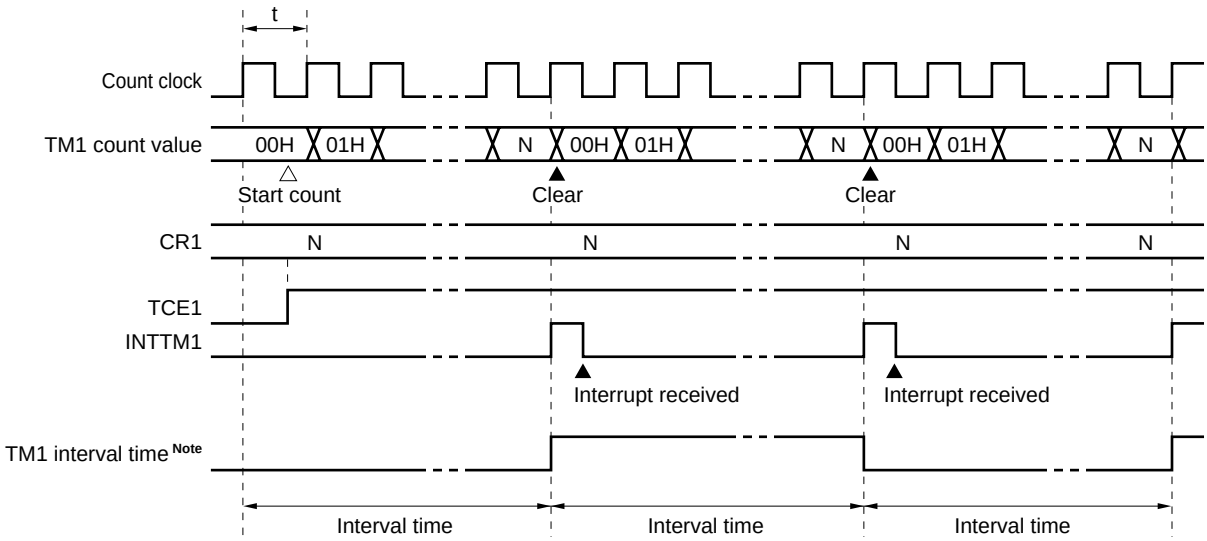
<2> After TCE1 = 1 is set, count operation starts.

<3> If the values of TM1 and CR1 match, the INTTM1 is generated and TM1 is cleared to 00H.

<4> INTTM1 generates repeatedly at the same interval. Set TCE1 to 0 to stop count operation.

Figure 7-4. Interval Timer Operation Timings (1/3)

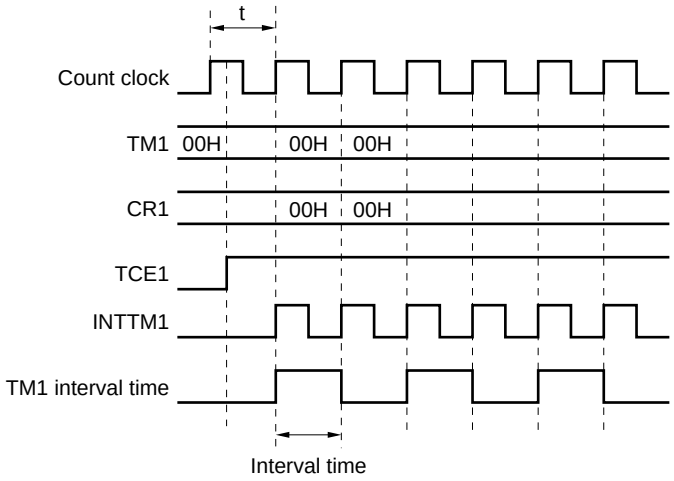
(a) Basic operation



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

Figure 7-4. Interval Timer Operation Timings (2/3)

(b) When CR1 = 00H



(c) When CR1 = FFH

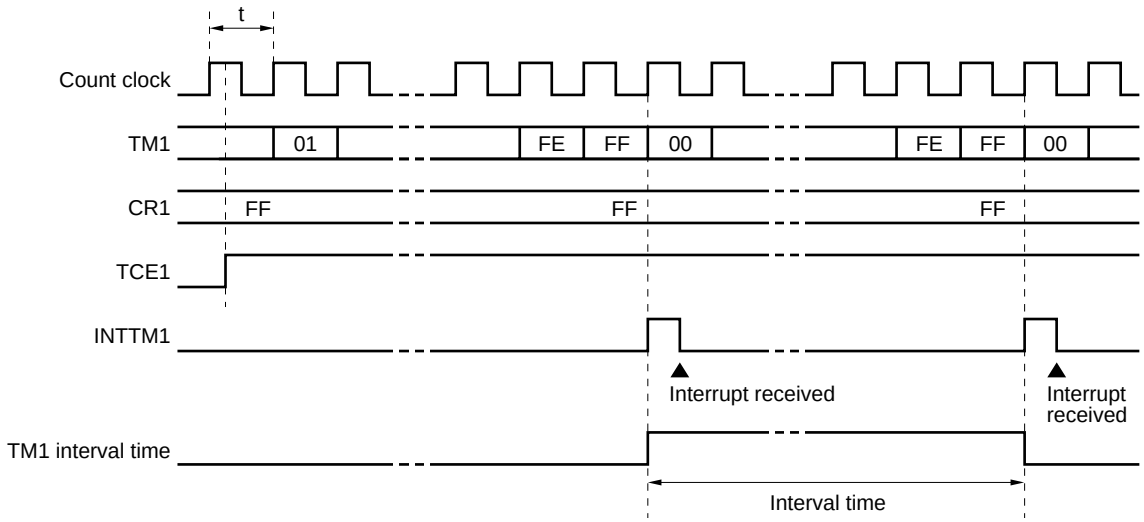
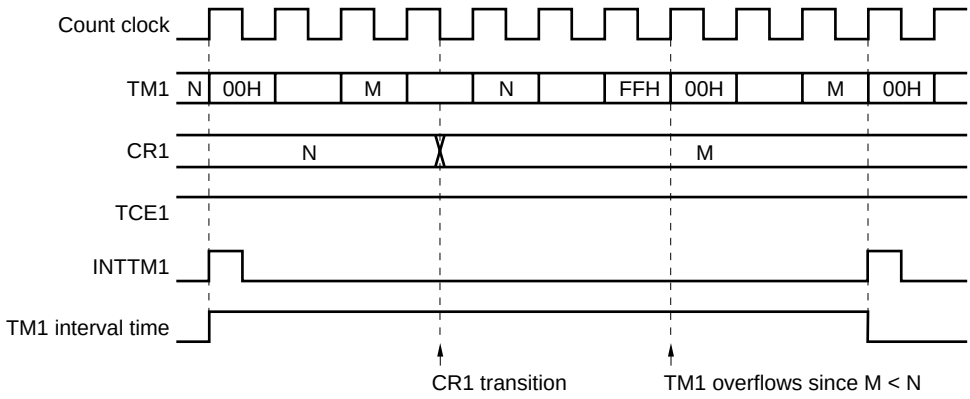
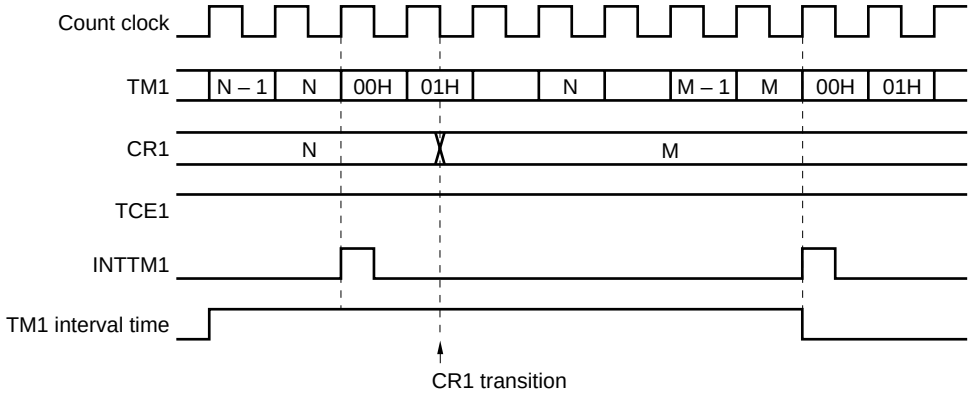


Figure 7-4. Interval Timer Operation Timings (3/3)

(d) Operated by CR1 transition (M < N)



(e) Operated by CR1 transition (M > N)

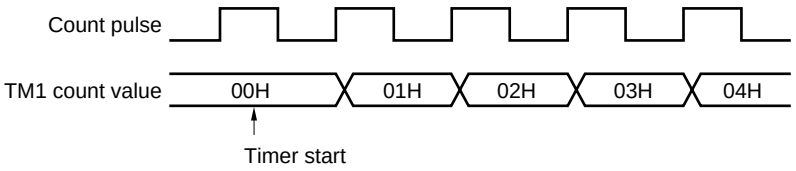


7.5 8-Bit Timer 1 TM1 Cautions

(1) Timer start errors

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

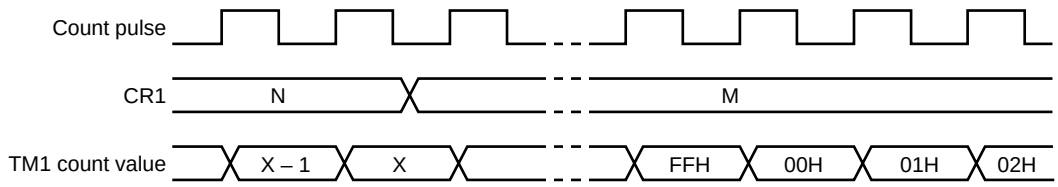
Figure 7-5. 8-Bit Timer 1 (TM1) Start Timing



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

If the values after the 8-bit compare register 1 (CR1) is changed are smaller than the value of 8-bit timer register 1 (TM1), TM1 continues counting, overflows and then restarts counting from 0. Thus, if the value (M) after CR1 change is smaller than value (N) before the change, it is necessary to restart the timer after changing CR1.

Figure 7-6. Timing after Compare Register Change during Timer Count Operation



Caution Always set TCE1 = 0 before setting the STOP state.

Remark N > X > M

(3) TM1 reading during timer operation

When TM1 is read during operation, choose a count clock which has a longer high/low level wave because 8-bit counter (TM1) is stopped temporary.

CHAPTER 8 8-BIT TIMER/EVENT COUNTERS 2 TM2 AND 3 TM3

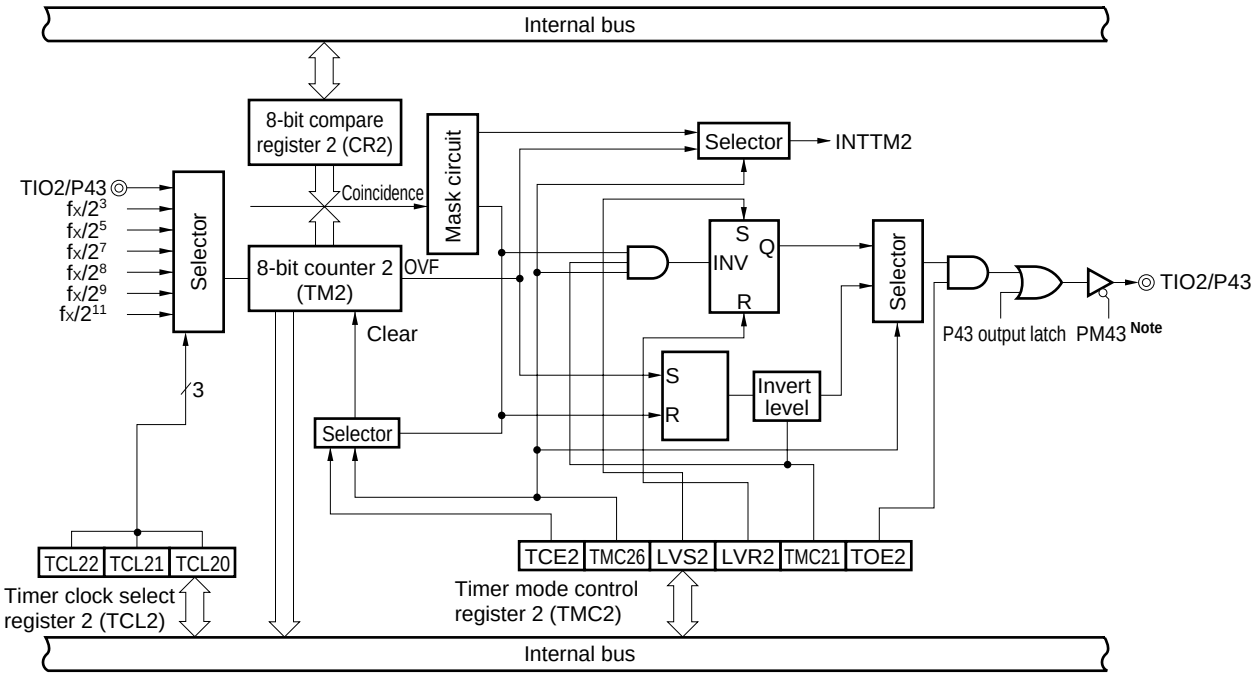
8.1 8-Bit Timer/Event Counters 2 TM2 and 3 TM3 Functions

The 8-bit timer/event counters 2 TM2 and 3 TM3 can have the following functions.

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

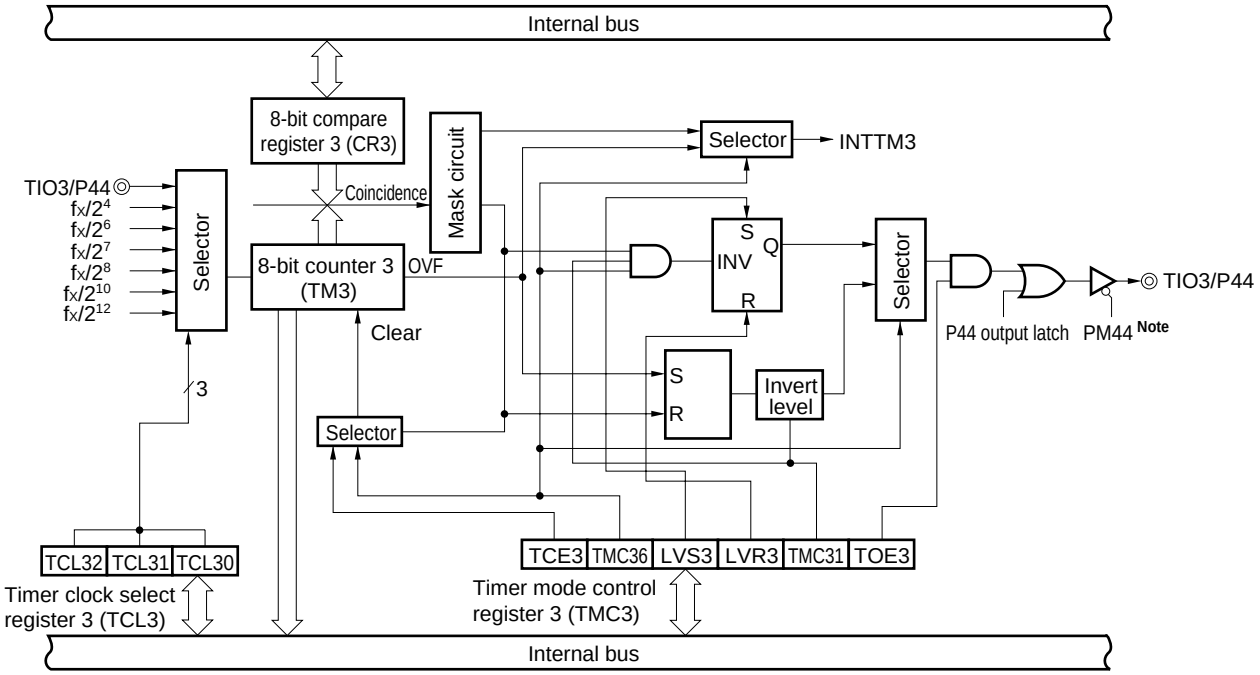
Figure 8-1 shows 8-bit timer/event counter 2 TM2 block diagram, and Figure 8-2 shows 8-bit timer/event counter 3 TM3 block diagram.

Figure 8-1. 8-Bit Timer/Event Counter 2 TM2 Block Diagram



Note Bit 3 of port mode register 4 (PM4)

Figure 8-2. 8-Bit Timer/Event Counter 3 TM3 Block Diagram



Note Bit 4 of port mode register 4 (PM4)

8.2 8-Bit Timer/Event Counters 2 TM2 and 3 TM3 Configurations

The 8-bit timer/event counters 2 TM2 and 3 TM3 consist of the following hardware.

Table 8-1. 8-Bit Timer/Event Counters 2 TM2 and 3 TM3 Configurations

Item	Configuration
Timer register	8-bit counter × 2 (TM2, TM3)
Register	8-bit compare register × 2 (CR2, CR3)
Timer output	2 (TIO2, TIO3)
Control register	Timer clock select registers 2, 3 (TCL2, TCL3) 8-bit timer mode control registers 2, 3 (TMC2, TMC3) Port mode register 4 (PM4)

(1) 8-bit counters 2, 3 (TM2, TM3)

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

The counter is incremented in synchronization with the rising edge of the count clock. When count value is read during operation, count clock input is temporary stopped, and then the count value is read. In the following situations, the count value is set to 00H.

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

<2> Clear TCE2 and TCE3

<3> Match between TM2 and TM3 and CR2 and CR3 in clear and start mode with match between TM2 and TM3 and CR2 and CR3

(2) 8-bit compare registers 2, 3 (CR2, CR3)

The value set in the CR2 is constantly compared with the 8-bit counter 2 (TM2) count value and the value set in the CR3 is constantly compared with the 8-bit counter 3 (TM3) count value, and interrupt requests (INTTM2 and INTTM3) are generated if they match (except PWM mode).

CR2 and CR3 are set with an 8-bit memory manipulation instruction. They are not set with a 16-bit memory manipulation instruction.

It is possible to rewrite the values of CR2 and CR3 within 00H to FFH during count operation.

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

8.3 8-Bit Timer/Event Counters 2 TM2 and 3 TM3 Control Registers

The following three types of registers are used to control 8-bit timer/event counters 2 TM2 and 3 TM3.

- Timer clock select registers 2, 3 (TCL2, TCL3)
- 8-bit timer mode control registers 2, 3 (TMC2, TMC3)
- Port mode register 4 (PM4)

(1) Timer clock select registers 2, 3 (TCL2, TCL3)

These registers set count clocks of 8-bit counters 2 and 3 (TM2 and TM3).

TCL2 and TCL3 are set with an 8-bit memory manipulation instruction.

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

Figure 8-3. Timer Clock Select Register 2 (TCL2) Format

Address: FF74H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL2	0	0	0	0	0	TCL22	TCL21	TCL20

TCL22	TCL21	TCL20	Count Clock Selection
0	0	0	TIO2 falling edge
0	0	1	TIO2 rising edge
0	1	0	$f_X/2^3$ (1.04 MHz)
0	1	1	$f_X/2^5$ (261 kHz)
1	0	0	$f_X/2^7$ (65.4 kHz)
1	0	1	$f_X/2^8$ (32.7 kHz)
1	1	0	$f_X/2^9$ (16.3 kHz)
1	1	1	$f_X/2^{11}$ (4.09 kHz)

Cautions 1. When rewriting TCL2 to other data, stop the timer operation beforehand.

2. Bits 3 to 7 must be set to 0.

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses apply to operation with f_X = 8.38 MHz

Figure 8-4. Timer Clock Select Register 3 (TCL3) Format

Address: FF75H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL3	0	0	0	0	0	TCL32	TCL31	TCL30

TCL32	TCL31	TCL30	Count Clock Selection
0	0	0	TIO3 falling edge
0	0	1	TIO3 rising edge
0	1	0	$f_X/2^4$ (523 kHz)
0	1	1	$f_X/2^6$ (130 kHz)
1	0	0	$f_X/2^7$ (65.4 kHz)
1	0	1	$f_X/2^8$ (32.7 kHz)
1	1	0	$f_X/2^{10}$ (8.18 kHz)
1	1	1	$f_X/2^{12}$ (2.04 kHz)

Cautions 1. When rewriting TCL3 to other data, stop the timer operation beforehand.
2. Bits 3 to 7 must be set to 0.

Remarks 1. f_X: Main system clock oscillation frequency
2. Figures in parentheses apply to operation with f_X = 8.38 MHz

(2) 8-bit timer mode control registers 2, 3 (TMC2, TMC3)

TMC2 and TMC3 are registers which sets up the following five types.

<1> 8-bit counters 2 and 3 (TM2 and TM3) count operation control
<2> 8-bit counters 2 and 3 (TM2 and TM3) operation mode selection
<3> Timer output F/F (flip flop) status setting
<4> Active level selection in timer F/F control or PWM (free-running) mode
<5> Timer output control

TMC2 and TMC3 are set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears these registers to 00H.

Figure 8-5. 8-Bit Timer Mode Control Registers 2 and 3 (TMC2 and TMC3) Format

Address: FF77H (TMC2) FF78H (TMC3) After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TMCn	TCEn	TMCn6	0	0	LVSn	LVRn	TMCn1	TOEn

TCEn	TMn Count Operation Control
0	After clearing counter to 0, count operation disabled (prescaler disabled)
1	Count operation start

TMCn6	TMn Operation Mode Selection
0	Clear and start mode by matching between TMn and CRn
1	PWM (Free-running) mode

LVSn	LVRn	Timer Output F/F Status Setting
0	0	No change
0	1	Timer output F/F reset (to 0)
1	0	Timer output F/F set (to 1)
1	1	Setting prohibited

TMCn1	In Other Modes (TMCn6 = 0)	In PWM Mode (TMCn6 = 1)
	Timer F/F Control	Active Level Selection
0	Inversion operation disabled	Active high
1	Inversion operation enabled	Active low

TOEn	Timer Output Control
0	Output disabled (Port mode)
1	Output enabled

Cautions 1. Bits 4 and 5 must be set to 0.

2. Bits 2 and 3 are write-only.

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

2. If LVSn and LVRn are read after data is set, they will be 0.

3. n = 2, 3

(3) Port mode register 4 (PM4)

This register sets port 4 to the input or output mode in 1-bit units.

To use the P43/TIO2 and P44/TIO3 pins as timer output pins, clear the output latches of PM43 and PM44 and P43 and P44 to 0.

PM4 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM4 to FFH.

Figure 8-6. Port Mode Register 4 (PM4) Format

Address: FF24H After Reset: FFH R/W

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

PM4n	P4n Pin I/O Mode Selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark n = 0 to 4

8.4 8-Bit Timer/Event Counters 2 TM2 and 3 TM3 Operations

8.4.1 8-bit interval timer operation

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

When the count values of the 8-bit counter n (TMn) match the values set to CRn, counting continues with the TMn values cleared to 0 and the interrupt request signal (INTTMn) is generated.

Count clock of the 8-bit timer register n (TMn) can be selected with the timer clock select register n (TCLn).

[Setting]

<1> Set the registers.
- TCLn: Select count clock.
- CRn: Compare value
- TMCn: Select clear and start mode by match of TMn and CRn.
  (TMCn = 0000×××0B × = don't care)

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

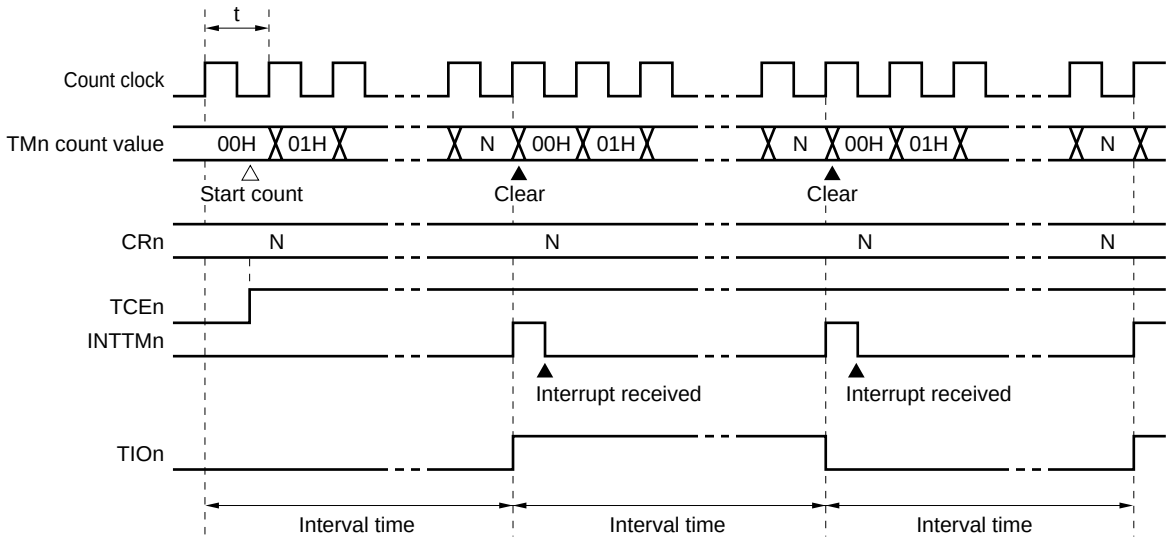
<3> If the values of TMn and CRn match, the INTTMn is generated and TMn is cleared to 00H.

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

Remark n = 2, 3

Figure 8-7. Interval Timer Operation Timings (1/3)

(a) Basic operation

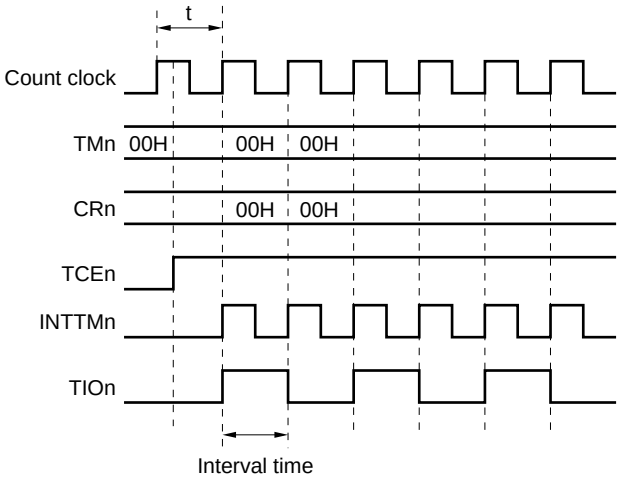


Remarks 1. Interval time = (N + 1) × t: N = 00H to FFH

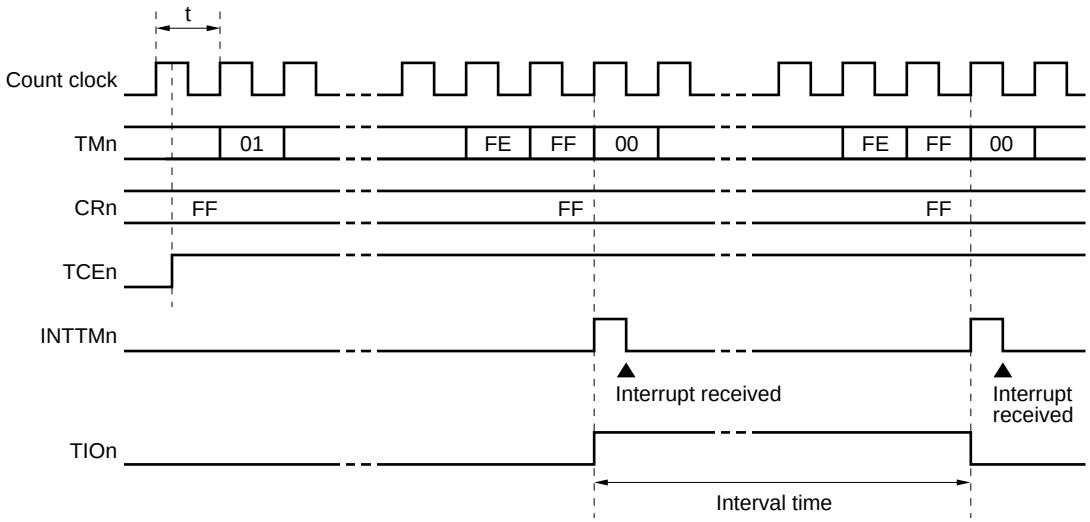
2. n = 2, 3

Figure 8-7. Interval Timer Operation Timings (2/3)

(b) When CRn = 00H



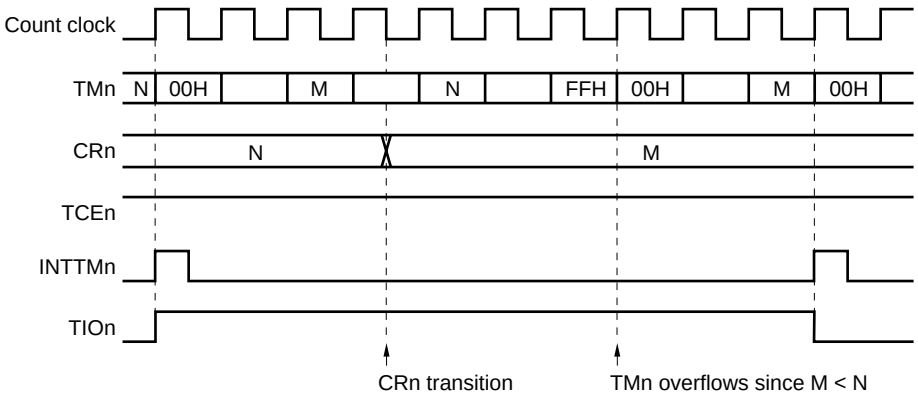
(c) When CRn = FFH



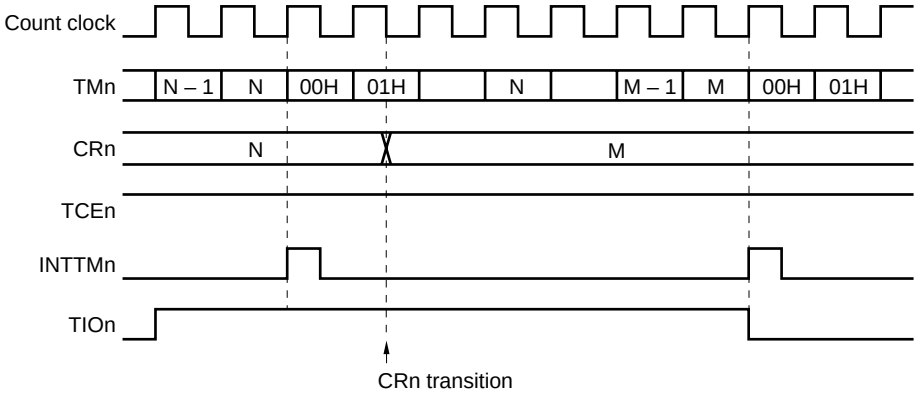
Remark n = 2, 3

Figure 8-7. Interval Timer Operation Timings (3/3)

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



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



Remark n = 2, 3

8.4.2 External event counter operation

The external event counter counts the number of external clock pulses to be input to the TIOn.

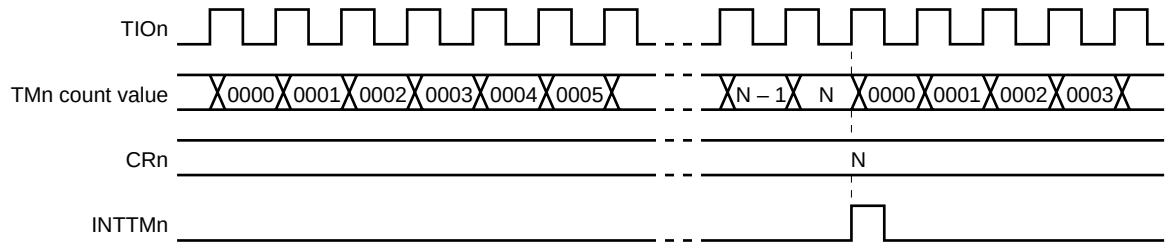
TMn is incremented each time the valid edge specified with the timer clock select register n (TCLn) is input. Either the rising or falling edge can be selected.

When the TMn counted values match the values of 8-bit compare register n (CRn), TMn is cleared to 0 and the interrupt request signal (INTTMn) is generated.

Whenever the TMn value matches the value of CRn, INTTMn is generated.

Remark n = 2, 3

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



Remark n = 2, 3

8.4.3 Square-wave output operation (8-bit resolution)

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

TIOn pin output status is inverted at intervals of the count value preset to CRn by setting bit 0 (TOEn) of 8-bit timer mode control register n (TMCn) to 1. This enables a square wave with any selected frequency to be output (duty = 50%).

[Setting]

<1> Set each register.

- Set port latch and port mode register to 0.
- TCLn: Select count clock
- CRn: Compare value
- TMCn: Clear and start mode by match of TMn and CRn

LVSn	LVRn	Timer Output F/F Status Setting
1	0	High-level output
0	1	Low-level output

Timer output F/F inversion enabled

Timer output enabled → TIOn = 1

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

<3> Timer output F/F is inverted by match of TMn and CRn. After INTTMn is generated, TMn is cleared to 00H.

<4> Timer output F/F is inverted at the same interval and square wave is output from TIOn.

Remark n = 2, 3

8.4.4 8-bit PWM output operation

8-bit timer/event counters operate as PWM output when bit 6 (TMCn6) of 8-bit timer mode control register n (TMCn) is set to 1.

The duty rate pulse determined by the value set to 8-bit compare register n (CRn) is output from TIOn.

Set the active level width of PWM pulse to CRn, and the active level can be selected with bit 1 (TMCn1) of TMCn.

Count clock can be selected with bit 0 to bit 2 (TCLn0 to TCLn2) of timer clock select register n (TCLn).

PWM output enable/disable can be selected with bit 0 (TOEn) of TMCn.

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

Remark n = 2, 3

(1) PWM output basic operation

[Setting]

<1> Set port latch (P43, P44) and port mode register 4 (PM43, PM44) to 0.
<2> Set active level width with 8-bit compare register (CRn).
<3> Select count clock with timer clock select register n (TCLn).
<4> Set active level with bit 1 (TMCn1) of TMCn.
<5> Count operation starts when bit 7 (TCEn) of TMCn is set to 1.
Set TCEn to 0 to stop count operation.

[PWM output operation]

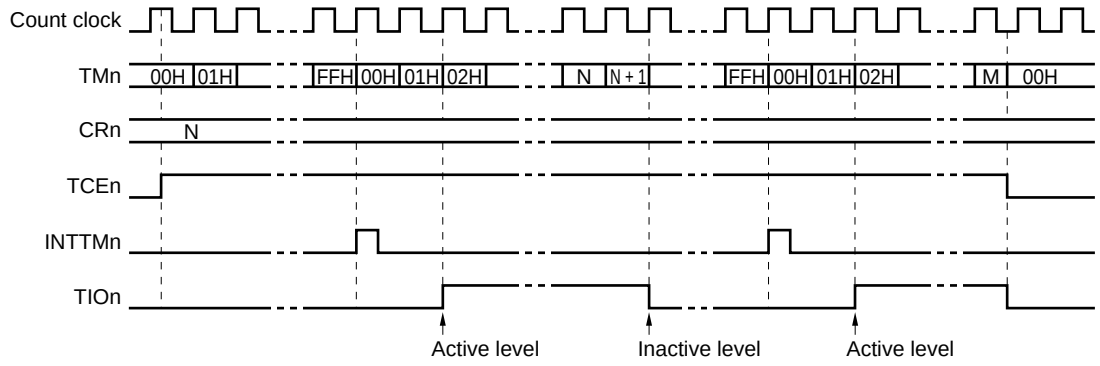
<1> PWM output (output from TIOn) outputs inactive level after count operation starts until overflow is generated.
<2> When overflow is generated, the active level set in <1> of setting is output.
The active level is output until CRn matches the count value of 8-bit counter n (TMn).
<3> After the CRn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

Remark n = 2, 3

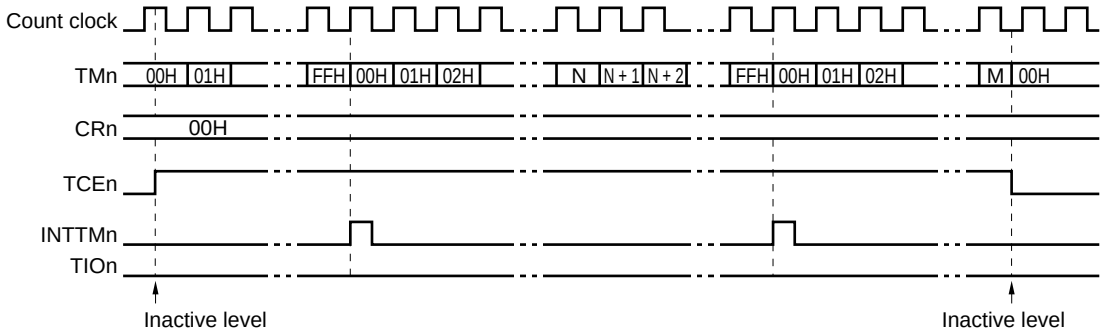
(a) PWM output basic operation

Figure 8-9. PWM Output Operation Timing

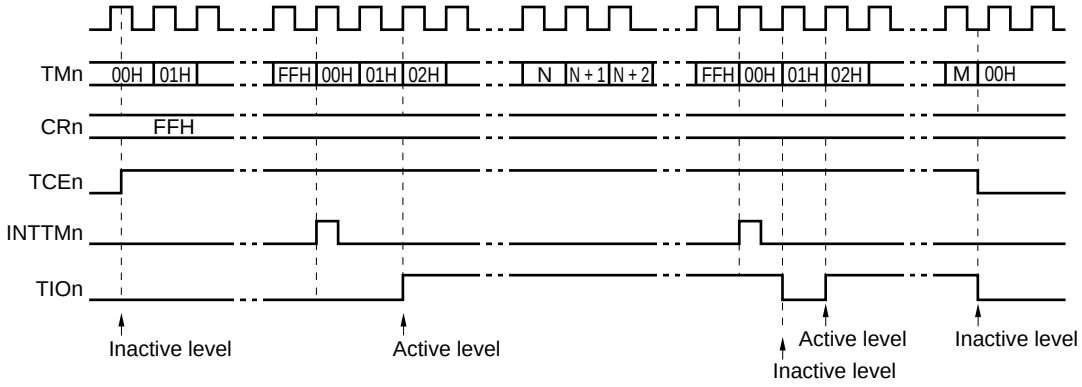
(i) Basic operation (active level = H)



(ii) CRn = 0



(iii) CRn = FFH

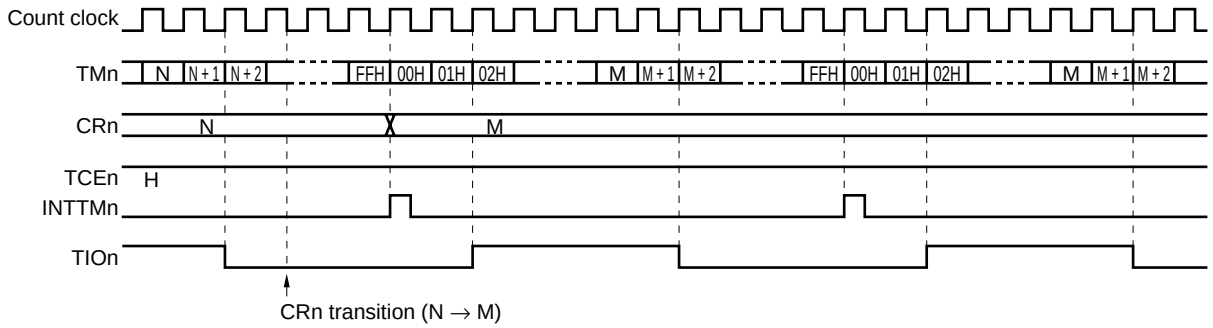


Remark n = 2, 3

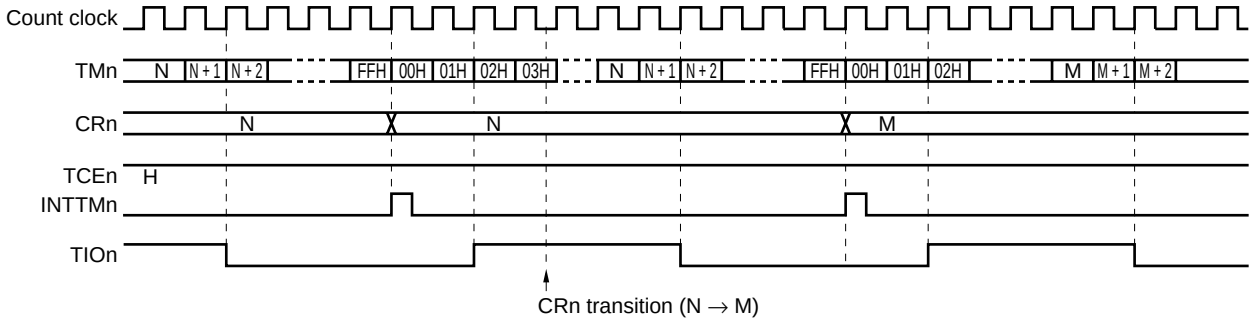
(b) Operation by change of CRn

Figure 8-10. Operation Timing by Change of CRn

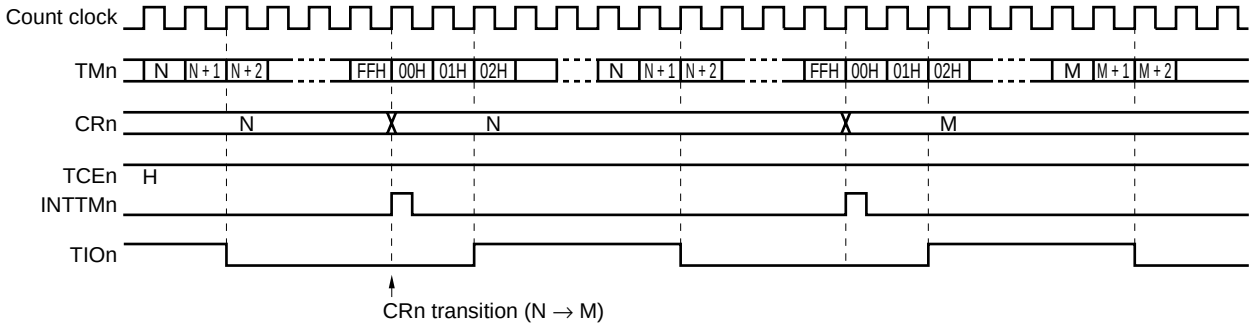
(i) Change of CRn value to N to M before overflow of TMn



(ii) Change of CRn value to N to M after overflow of TMn



(iii) Change of CRn value between two clocks (00H and 01H) after overflow of TMn



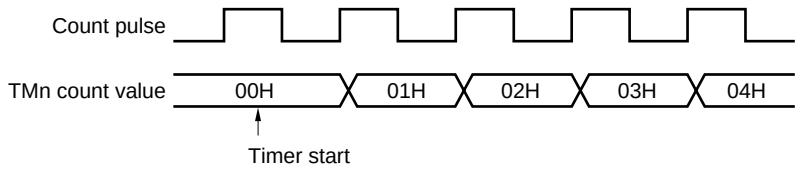
Remark n = 2, 3

8.5 8-Bit Timer/Event Counters 2 TM2 and 3 TM3 Cautions

(1) Timer start errors

An error with a maximum of one clock may occur concerning the time required for a match signal to be generated after timer start. This is because 8-bit counters 2 and 3 (TM2 and TM3) is started asynchronously with the count pulse.

Figure 8-11. 8-Bit Counters 2 and 3 (TM2 and TM3) Start Timing

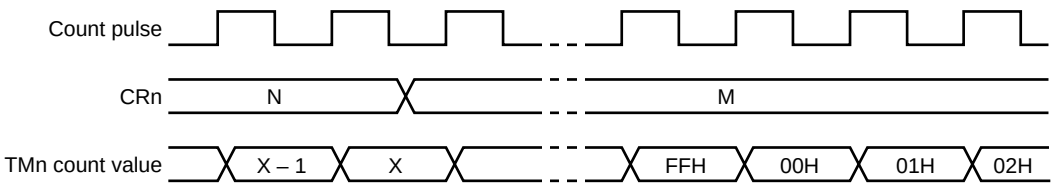


Remark n = 2, 3

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

If the values after the 8-bit compare registers 2 and 3 (CR2 and CR3) are changed are smaller than the value of 8-bit counters 2 and 3 (TM2 and TM3), TM2 and TM3 continue counting, overflow and then restart counting from 0. Thus, if the value (M) after CR2 and CR3 change is smaller than value (N) before the change, it is necessary to restart the timer after changing CR2 and CR3.

Figure 8-12. Timing after Compare Register Change during Timer Count Operation



Remarks 1. N > X > M

2. n = 2, 3

(3) TM2 and TM3 reading during timer operation

When TM2 and TM3 are read during operation, choose a select clock which has a longer high/low level wave because the select clock is stopped temporarily.

[MEMO]

CHAPTER 9 WATCH TIMER

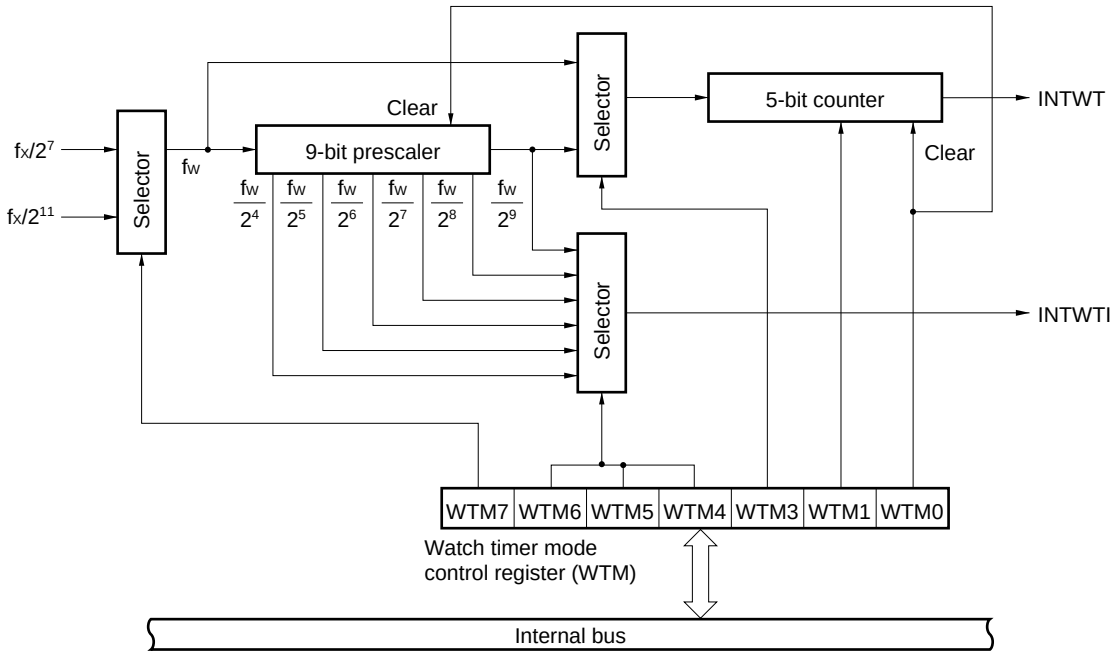
9.1 Watch Timer Functions

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch timer and the interval timer can be used simultaneously.
Figure 9-1 shows watch timer block diagram.

Figure 9-1. Watch Timer Block Diagram



(1) Watch timer

When the main system clock is used, interrupt requests (INTWT) are generated at 0.25 second (at f_X = 8.38-MHz operation) intervals.

(2) Interval timer

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

Table 9-1. Interval Timer Interval Time

Interval Time	When Operated at f_X = 8.38 MHz
$2^{12}/f_X$	489 μs
$2^{13}/f_X$	978 μs
$2^{14}/f_X$	1.96 ms
$2^{15}/f_X$	3.91 ms
$2^{16}/f_X$	7.82 ms
$2^{17}/f_X$	15.65 ms

Remark f_X: Main system clock oscillation frequency

9.2 Watch Timer Configuration

The watch timer consists of the following hardware.

Table 9-2. Watch Timer Configuration

Item	Configuration
Counter	5 bits × 1
Prescaler	9 bits × 1
Control register	Watch timer mode control register (WTM)

9.3 Watch Timer Control Register

The watch timer mode control register (WTM) is used to control the watch timer.

- **Watch timer mode control register (WTM)**
  This register sets the watch timer count clock, watch timer operation mode, prescaler interval time, and prescaler and 5-bit counter operation enable/disable.
  WTM is set with a 1-bit or 8-bit memory manipulation instruction.
  $\overline{\text{RESET}}$ input clears WTM to 00H.

Figure 9-2. Watch Timer Mode Control Register (WTM) Format

Address: FF41H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	0	WTM1	WTM0

WTM7	Watch Timer Count Clock Selection
0	$f_X/2^7$ (65.4 kHz)
1	$f_X/2^{11}$ (4.09 kHz)

WTM6	WTM5	WTM4	Prescaler Interval Time Selection
0	0	0	$2^4/f_W$ (3.91 ms)
0	0	1	$2^5/f_W$ (7.82 ms)
0	1	0	$2^6/f_W$ (15.6 ms)
0	1	1	$2^7/f_W$ (31.2 ms)
1	0	0	$2^8/f_W$ (62.5 ms)
1	0	1	$2^9/f_W$ (125 ms)
Other than above			Setting prohibited

WTM3	Watch Flag Set Time Selection
0	Normal operating mode (flag set at $f_W/2^{14}$)
1	Fast feed operating mode (flag set at $f_W/2^5$)

WTM1	5-bit Counter Operation Control
0	Clear after operation stop
1	Start

WTM0	Watch Timer Operation Control
0	Operation stop (clear both prescaler and timer)
1	Operation enabled

Remarks 1. f_W: Watch timer clock frequency ($f_X/2^7$ or $f_X/2^{11}$)

2. f_X: Main system clock oscillation frequency

3. Figures in parentheses apply to operation with f_X = 8.38 MHz, f_W = 4.09 kHz.

9.4 Watch Timer Operations

9.4.1 Watch timer operation

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

The watch timer generates interrupt requests at a constant time interval.

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

For simultaneous operation of the interval timer, zero-second start can be set only for the watch timer by setting WTM1 to 0. However, since the 9-bit prescaler is not cleared the first overflow of the watch timer (INTWT) after zero-second start may include an error of up to $2^9 \times 1/f_W$.

9.4.2 Interval timer operation

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

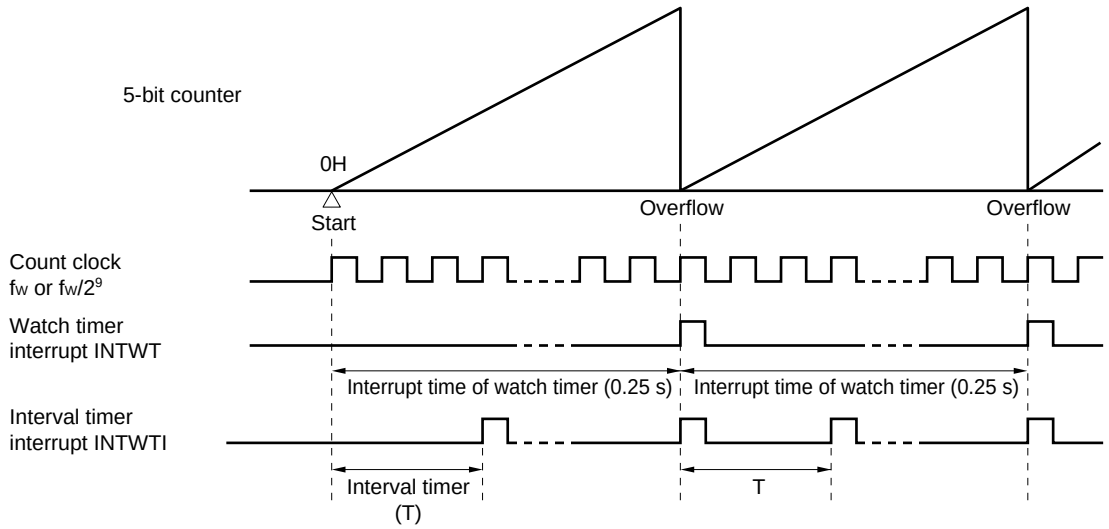
The interval time can be selected with bits 4 to 6 (WTM4 to WTM6) of the watch timer mode control register (WTM).

Table 9-3. Interval Timer Interval Time

WTM6	WTM5	WTM4	Interval Time	When Operated at f_W = 65.4 kHz	When Operated at f_W = 4.09 kHz
0	0	0	$2^4 \times 1/f_W$	244 μs	3.91 ms
0	0	1	$2^5 \times 1/f_W$	489 μs	7.81 ms
0	1	0	$2^6 \times 1/f_W$	978 μs	15.6 ms
0	1	1	$2^7 \times 1/f_W$	1.96 ms	31.3 ms
1	0	0	$2^8 \times 1/f_W$	3.91 ms	62.5 ms
1	0	1	$2^9 \times 1/f_W$	7.82 ms	125 ms
Other than above			Setting prohibited		

Remark f_W: Watch timer clock frequency

Figure 9-3. Watch Timer/Interval Timer Operation Timing



Remark f_W: Watch timer clock frequency

( ): f_W = 4.09 kHz (f_X = 8.38 MHz)

[MEMO]

CHAPTER 10 WATCHDOG TIMER

10.1 Watchdog Timer Functions

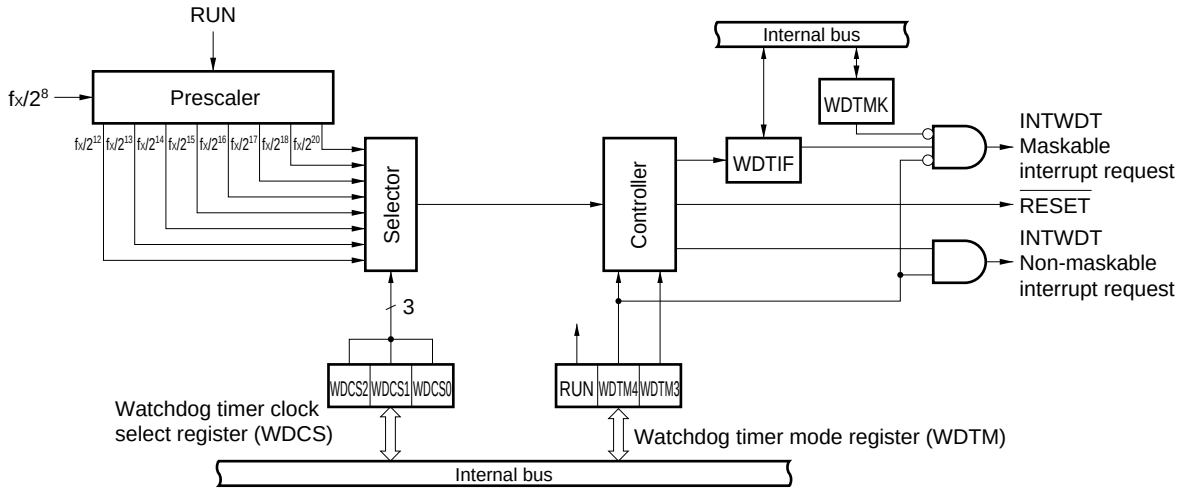
The watchdog timer has the following functions.

- Watchdog timer
- Interval timer

Caution Select the watchdog timer mode or the interval timer mode with the watchdog timer mode register (WDTM).

Figure 10-1 shows the watchdog timer block diagram.

Figure 10-1. Watchdog Timer Block Diagram



(1) Watchdog timer mode

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

Table 10-1. Watchdog Timer Runaway Detection Time

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

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses apply to operation with f_X = 8.38 MHz.

(2) Interval timer mode

Interrupt requests are generated at preset time intervals.

Table 10-2. Interval Time

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

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses apply to operation with f_X = 8.38 MHz.

10.2 Watchdog Timer Configuration

The watchdog timer consists of the following hardware.

Table 10-3. Watchdog Timer Configuration

Item	Configuration
Control register	Watchdog timer clock select register (WDCS) Watchdog timer mode register (WDTM)

10.3 Watchdog Timer Control Registers

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

- Watchdog timer clock select register (WDCS)
- Watchdog timer mode register (WDTM)

(1) Watchdog timer clock select register (WDCS)

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

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

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

Figure 10-2. Watchdog Timer Clock Select Register (WDCS) Format

Address: FF42H After Reset: 00H R/W

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

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

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

2. Bits 3 to 7 must be set to 0.

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses apply to operation with f_X = 8.38 MHz

(2) Watchdog timer mode register (WDTM)

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

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

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

Figure 10-3. Watchdog Timer Mode Register (WDTM) Format

Address: FFF9H After Reset: 00H R/W

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

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

WDTM4	WDTM3	Watchdog Timer Operation Mode Selection Note 2, and Reset by the Watchdog Timer and Timer Interrupt Control
0	×	Interval timer mode (Maskable interrupt request occurs upon generation of an overflow)
1	0	Watchdog timer mode 1 (Non-maskable interrupt request occurs upon generation of an overflow)
1	1	Watchdog timer mode 2 (Reset operation is activated upon generation of an overflow)

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

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

Cautions 1. If the watchdog timer is cleared by setting 1 for RUN, the actual overflow time will be up to $2^8/f_X$ (seconds) shorter than the time set by the watchdog timer clock select register (WDCS).

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

Remark ×: don't care

10.4 Watchdog Timer Operations

10.4.1 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 runaways.

A watchdog timer count clock (runaway detection time interval) can be selected by using bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock select register (WDCS).

Watchdog timer starts by setting bit 7 (RUN) of WDTM to 1. After the watchdog timer is started, set RUN to 1 within the set runaway time interval. The watchdog timer can be cleared and counting is started.

If RUN is not set to 1 and the runaway detection time is past, system reset or a non-maskable interrupt request is generated according to the WDTM bit 3 (WDTM3) value.

The watchdog timer is cleared if RUN is set to 1.

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 runaway detection time may be shorter than the set time by a maximum of $2^8/f_X$ [sec].

Table 10-4. Watchdog Timer Runaway Detection Time

WDCS2	WDCS1	WDCS0	Runaway Detection Time
0	0	0	$f_X/2^{12}$ (489 μs)
0	0	1	$f_X/2^{13}$ (978 μs)
0	1	0	$f_X/2^{14}$ (1.96 ms)
0	1	1	$f_X/2^{15}$ (3.91 ms)
1	0	0	$f_X/2^{16}$ (7.82 ms)
1	0	1	$f_X/2^{17}$ (15.6 ms)
1	1	0	$f_X/2^{18}$ (31.3 ms)
1	1	1	$f_X/2^{20}$ (125 ms)

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses apply to operation with f_X = 8.38 MHz.

10.4.2 Interval timer operation

The watchdog timer operates as an interval timer which generates interrupt request repeatedly at an interval of the preset count value when bit 3 (WDTM3) and bit 4 (WDTM4) of the watchdog timer mode register (WDTM) are set to 1 and 0, respectively.

When the watchdog timer operates as interval timer, the interrupt mask flag (WDTMK) and priority specify flag (WDTPR) are validated and the maskable interrupt request (INTWDT) can be generated. Among maskable interrupts, 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 $2^8/f_X$ [sec].

Table 10-5. Interval Timer Interval Time

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

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses apply to operation with f_X = 8.38 MHz.

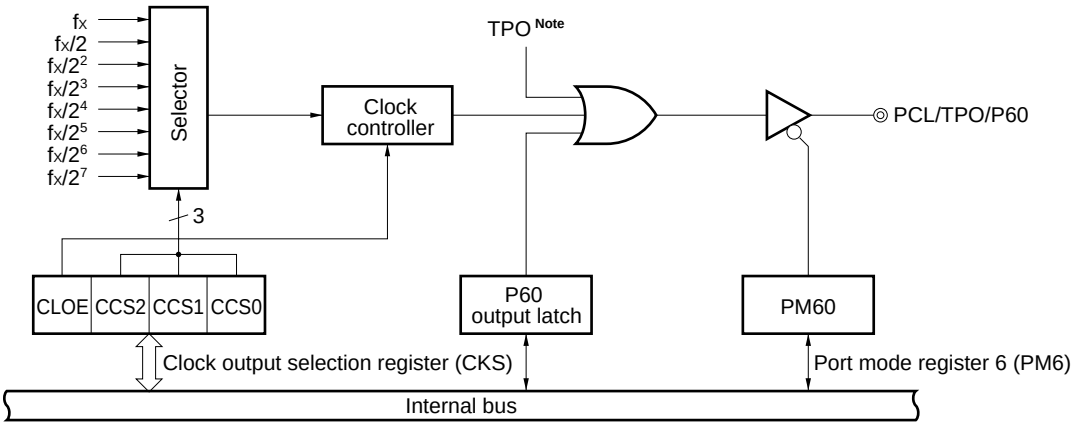
CHAPTER 11 CLOCK OUTPUT CONTROLLER

11.1 Clock Output Controller Functions

The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral LSIs. The clock selected with the clock output selection register (CKS) is output from the PCL/TPO/P60 pin.

Figure 11-1 shows the clock output controller block diagram.

Figure 11-1. Clock Output Controller Block Diagram



Note TPO: Prescaler output signal of 16-bit timer 0 TM0

11.2 Clock Output Controller Configuration

The clock output controller consists of the following hardware.

Table 11-1. Clock Output Controller Configuration

Item	Configuration
Control register	Clock output selection register (CKS) Port mode register 6 (PM6)

11.3 Clock Output Control Controller Registers

The following two types of registers are used to control the clock output controller.

- Clock output selection register (CKS)
- Port mode register 6 (PM6)

(1) Clock output selection register (CKS)

This register sets output clock.

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

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

Caution To enable PCL output, set CCS0 to CCS2, and then set CLOE to 1 by using a 1-bit memory manipulation instruction.

Figure 11-2. Clock Output Selection Register (CKS) Format

Address: FF40H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKS	0	0	0	CLOE	0	CCS2	CCS1	CCS0

CLOE	PCL Output Enable/Disable Specification
0	Operation disabled.
1	Operation enabled.

CCS2	CCS1	CCS0	PCL Output Clock Selection
0	0	0	f_X (8.38 MHz)
0	0	1	$f_X/2$ (4.19 MHz)
0	1	0	$f_X/2^2$ (2.09 MHz)
0	1	1	$f_X/2^3$ (1.04 MHz)
1	0	0	$f_X/2^4$ (524 kHz)
1	0	1	$f_X/2^5$ (262 kHz)
1	1	0	$f_X/2^6$ (131 kHz)
1	1	1	$f_X/2^7$ (65.5 kHz)

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

2. Bits 3 and 5 to 7 must be set to 0.

Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses apply to operation with f_X = 8.38 MHz.

(2) Port mode register 6 (PM6)

This register sets port 6 input/output in 1-bit units.

When using the P60/PCL/TPO pin for clock output, set PM60 and the output latch of P60 to 0.

PM6 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM6 to FFH.

Figure 11-3. Port Mode Register 6 (PM6) Format

Address: FF26H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	1	PM61	PM60

PM6n	P6n Pin Input/Output Mode Selection (n = 0, 1)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

11.4 Clock Output Controller Operation

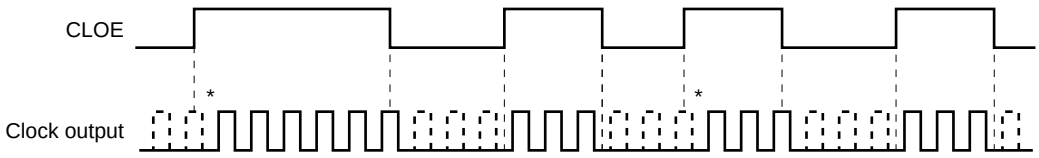
To output the clock pulse, follow the procedure described below.

<1> Select the clock pulse output frequency with bits 0 to 2 (CCS0 to CCS2) of the clock output selection register (CKS) (clock pulse output in disabled status).
<2> Set bit 0 (TPOE) of the 16-bit timer mode control register (TMC0) to 0 (prescaler signal output in disabled status).
<3> Set the P60 output latch to 0.
<4> Set bit 4 (PM60) of port mode register 6 to 0 (set to output mode).
<5> Set bit 4 (CLOE) of CKS to 1, and enable clock output.

Caution The clock output cannot be used if the output latch of P60 is set to 1.

Remark The clock output controller is designed not to output pulses with a small width during output enable/disable switching of the clock output. As shown in Figure 11-4, be sure to start output from the low period of the clock (marked with ***** in the figure below). When stopping output, do so after securing high level of the clock.

Figure 11-4. Remote Control Output Application Example



CHAPTER 12 A/D CONVERTER

12.1 A/D Converter Functions

The A/D converter is an 8-bit resolution converter that converts analog inputs into digital values. It can control up to 5 analog input channels (ANI0 to ANI4).

This A/D converter has the following functions:

(1) A/D conversion with 8-bit resolution

One channel of analog input is selected from ANI0 to ANI4, and A/D conversion is repeatedly executed with a resolution of 8 bits. Each time the conversion has been completed, interrupt request (INTAD) is generated.

(2) Power-fail detection function

This function is to detect a voltage drop in the battery of an automobile. The result of A/D conversion (value of the ADCR1 register) and the value of PFT register (PFT: power-fail compare threshold value register) are compared. If the condition for comparison is satisfied, INTAD is generated.

Figure 12-1. A/D Converter Block Diagram

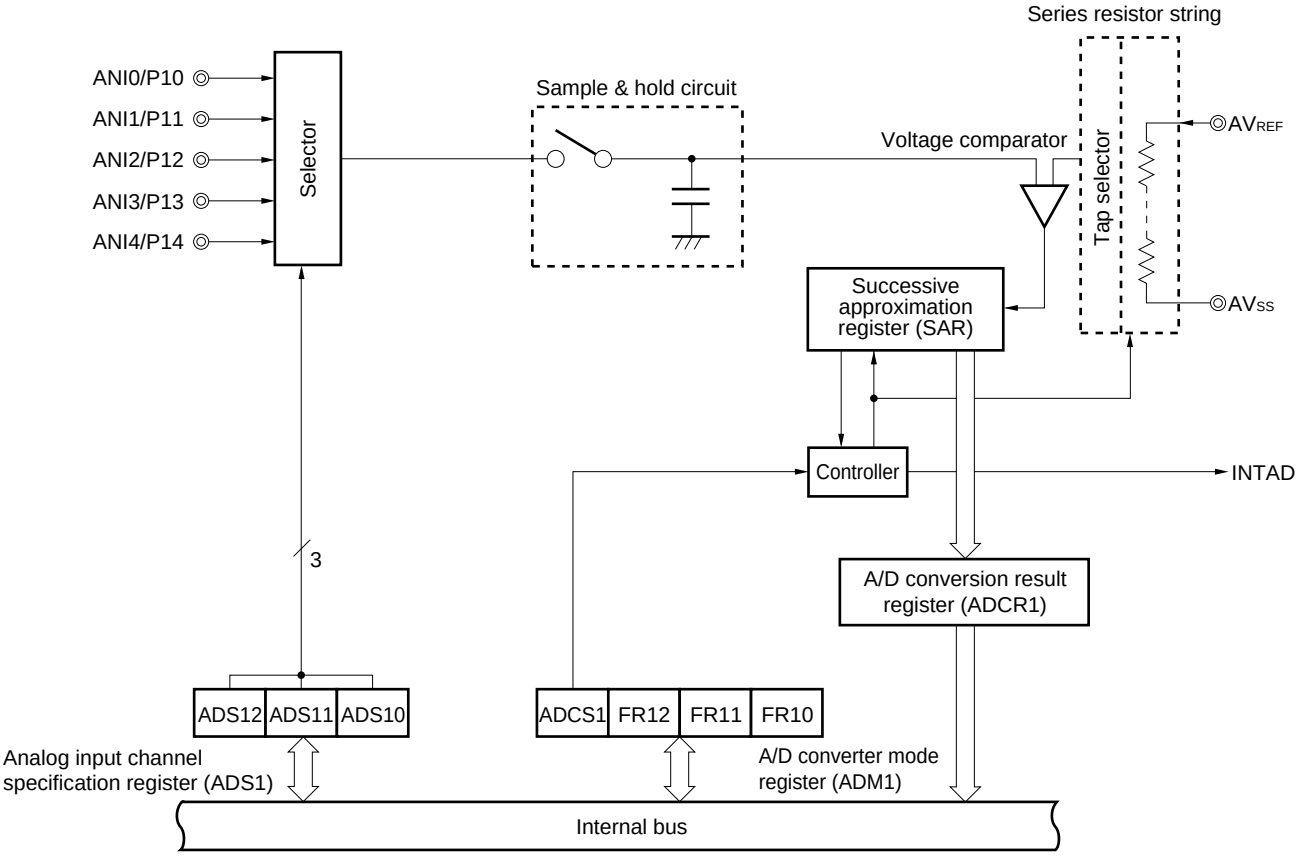
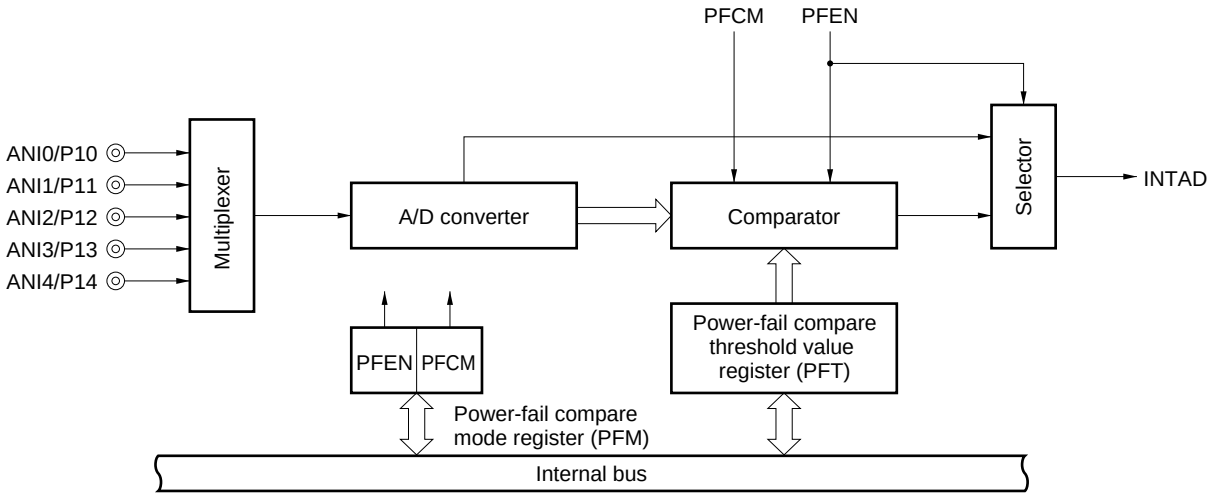


Figure 12-2. Power-Fail Detection Function Block Diagram



12.2 A/D Converter Configuration

A/D converter consists of the following hardware.

Table 12-1. A/D Converter Configuration

Item	Configuration
Analog input	5 channels (ANI0 to ANI4)
Register	Successive approximation register (SAR) A/D conversion result register (ADCR1)
Control register	A/D converter mode register (ADM1) Analog input channel specification register (ADS1) Power-fail compare mode register (PFM) Power-fail compare threshold value register (PFT)

(1) Successive approximation register (SAR)

This register compares the analog input voltage value to the voltage tap (compare voltage) value applied from the series resistor string, and holds the result from the most significant bit (MSB).

When up to the least significant bit (LSB) is set (end of A/D conversion), the SAR contents are transferred to the A/D conversion result register.

(2) A/D conversion result register (ADCR1)

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

ADCR1 is read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ADCR1 to 00H.

Caution When write operation is executed to A/D converter mode register (ADM1) and analog input channel specification register (ADS1), the contents of ADCR1 are undefined. Read the conversion result before write operation is executed to ADM1, ADS1. If a timing other than the above is used, the correct conversion result may not be read.

(3) Sample & hold circuit

The sample & hold circuit samples each analog input 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 in AV_{REF} to AV_{SS}, and generates a voltage to be compared to the analog input.

(6) ANI0 to ANI4 pins

These are five analog input pins to input analog signals to undergo A/D conversion to the A/D converter. ANI0 to ANI4 are alternate-function pins that can also be used for digital input.

Caution Keep the input voltage of ANI0 to ANI4 within the rated range. If a voltage outside the rated range is input, the conversion value of that channel is undefined, and the values of the other channels may be also affected.

(7) AV_{REF} pin (Shared with AV_{DD} pin)

This pin inputs the A/D converter reference voltage.

This pin also functions as an analog power supply pin. Supply power to this pin when the A/D converter is used. It converts signals input to ANI0 to ANI4 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 the standby mode.

(8) AV_{SS} pin

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

12.3 A/D Converter Control Registers

The following four types of registers are used to control A/D converter.

- A/D converter mode register (ADM1)
- Analog input channel specification register (ADS1)
- Power-fail compare mode register (PFM)
- Power-fail compare threshold value register (PFT)

(1) A/D converter mode register (ADM1)

This register sets the conversion time for analog input to be A/D converted, conversion start/stop and external trigger.

ADM1 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ADM1 to 00H.

Figure 12-3. A/D Converter Mode Register (ADM1) Format

Address: FF80H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADM1	ADCS1	0	FR12	FR11	FR10	0	0	0

ADCS1	A/D Conversion Operation Control
0	Conversion operation stop.
1	Conversion operation enabled.

FR12	FR11	FR10	Conversion Time Selection Note 1
0	0	0	144/fx (17.2 μs)
0	0	1	120/fx (14.3 μs)
0	1	0	96/fx (**Note 2**)
1	0	0	288/fx (34.4 μs)
1	0	1	240/fx (28.6 μs)
1	1	0	192/fx (22.9 μs)
Other than above			Setting prohibited

Notes 1. Set so that the A/D conversion time is 14 μs or more.

2. Setting prohibited because the A/D conversion time will be less than 14 μs.

Cautions 1. Bits 0 to 2 and 6 must be set to 0.

2. When rewriting FR10 to FR12 to other data, stop the A/D conversion operation beforehand.

Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses apply to operation with fx = 8.38 MHz.

(2) Analog input channel specification register (ADS1)

This register specifies the analog voltage input port for A/D conversion.

ADS1 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ADS1 to 00H.

Figure 12-4. Analog Input Channel Specification Register (ADS1) Format

Address: FF81H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS1	0	0	0	0	0	ADS12	ADS11	ADS10

ADS12	ADS11	ADS10	Analog Input Channel Specification
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
Other than above			Setting prohibited

Caution Bits 3 to 7 must be set to 0.

(3) Power-fail compare mode register (PFM)

The power-fail compare mode register (PFM) controls a comparison operation.

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

Figure 12-5. Power-Fail Compare Mode Register (PFM) Format

Address: FF82H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PFM	PFEN	PFCM	0	0	0	0	0	0

PFEN	Power-Fail Comparison Enable
0	Power-fail comparison disabled (used as normal A/D converter)
1	Power-fail comparison enabled (used to detect power failure)

PFCM		Power-Fail Compare Mode Selection
0	ADCR1 ≥ PFT	Generates interrupt request signal INTAD
	ADCR1 < PFT	Does not generate interrupt request signal INTAD
1	ADCR1 ≥ PFT	Does not generate interrupt request signal INTAD
	ADCR1 < PFT	Generates interrupt request signal INTAD

Caution Bits 0 to 5 must be set to 0.

(4) Power-fail compare threshold value register (PFT)

The power-fail compare threshold value register (PFT) sets a threshold value against which the result of A/D conversion is to be compared.

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

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

Figure 12-6. Power-Fail Compare Threshold Value Register (PFT) Format

Address: FF83H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PFT	PFT7	PFT6	PFT5	PFT4	PFT3	PFT2	PFT1	PFT0

12.4 A/D Converter Operations

12.4.1 Basic operations of A/D converter

<1> Select one channel for A/D conversion with the analog input channel specification register (ADS1).

<2> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<3> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the input analog voltage is held until the A/D conversion operation is ended.

<4> Set bit 7 of the successive approximation register (SAR) so that the tap selector sets the series resistor string voltage tap to (1/2) AV_{REF}.

<5> The voltage difference between the series resistor string voltage tap and analog input is compared with the voltage comparator. If the analog input is greater than (1/2) AV_{REF}, the MSB of SAR remains set. If the analog input is smaller than (1/2) AV_{REF}, the MSB is reset.

<6> Next, bit 6 of SAR is automatically set, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 7, as described below.

- Bit 7 = 1: (3/4) AV_{REF}
- Bit 7 = 0: (1/4) AV_{REF}

The voltage tap and analog input voltage are compared and bit 6 of SAR is manipulated as follows.

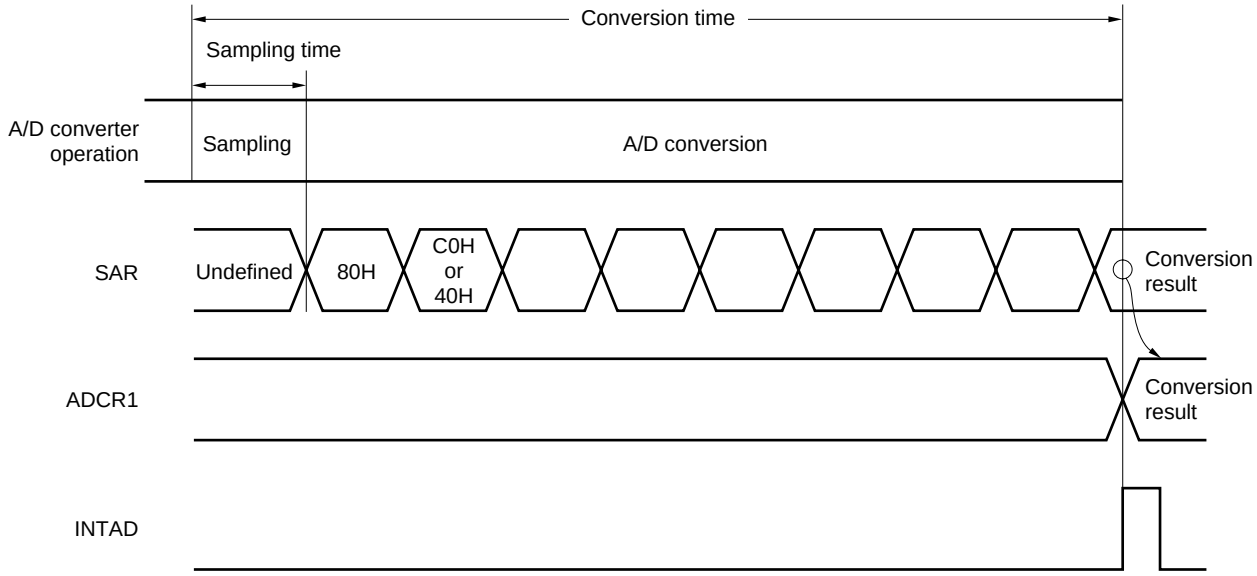
- Analog input voltage ≥ Voltage tap: Bit 6 = 1
- Analog input voltage < Voltage tap: Bit 6 = 0

<7> Comparison is continued in this way up to bit 0 of SAR.

<8> Upon completion of the comparison of 8 bits, an effective digital result value remains in SAR, and the result value is transferred to and latched in the A/D conversion result register (ADCR1).
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.
The occurrence of INTAD can be controlled by setting bit 6 (PFCM) of the power-fail compare mode register (PFM).

Caution The first A/D conversion value is undefined immediately after an A/D conversion operation has been started.

Figure 12-7. Basic Operation of 8-Bit A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS1) of the A/D converter mode register (ADM1) is reset (to 0) by software.

If a write operation to the ADM1 and analog input channel specification register (ADS1) is performed during an A/D conversion operation, the conversion operation is initialized, and if the ADCS1 bit is set (to 1), conversion starts again from the beginning.

$\overline{\text{RESET}}$ input clears the A/D conversion result register (ADCR1) to 00H.

12.4.2 Input voltage and conversion results

The relation between the analog input voltage input to the analog input pins (ANI0 to ANI4) and the A/D conversion result (stored in the A/D conversion result register (ADCR1)) is shown by the following expression.

$$ADCR1 = INT\left(\frac{V_{IN}}{AV_{REF}} \times 256 + 0.5\right)$$

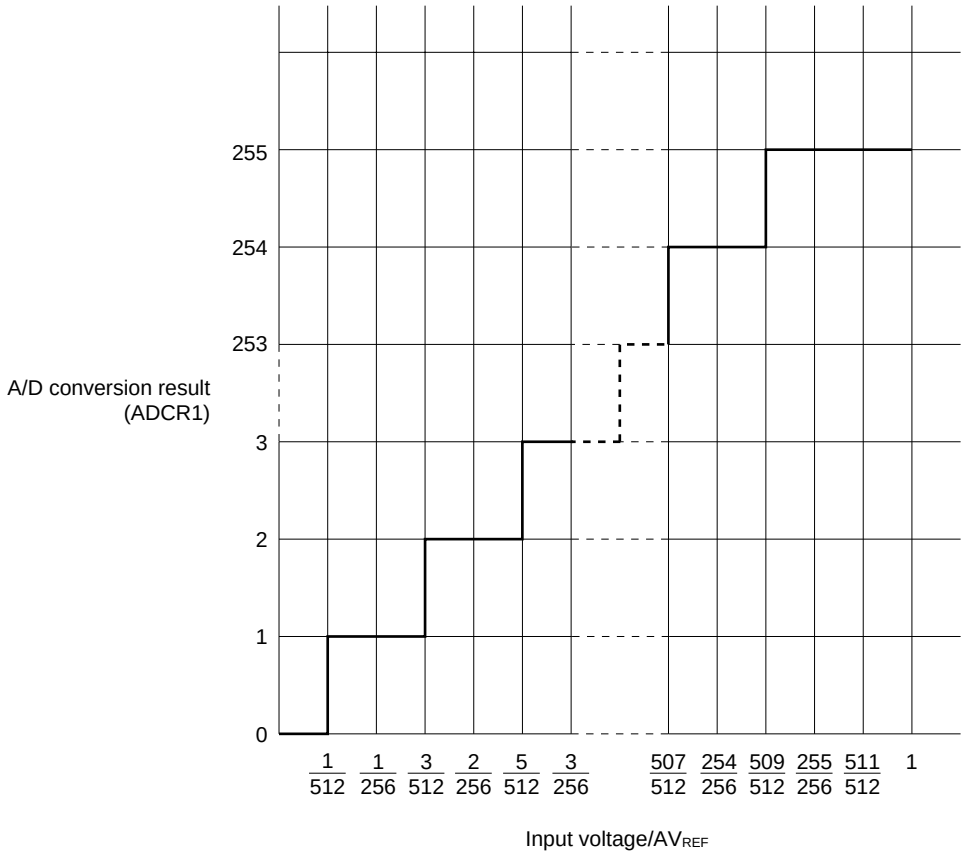
or

$$(ADCR1 - 0.5) \times \frac{AV_{REF}}{256} - V_{IN} < (ADCR1 + 0.5) \times \frac{AV_{REF}}{256}$$

where, INT( ): Function which returns integer part of value in parentheses
V_{IN}: Analog input voltage
AV_{REF}: AV_{REF} pin voltage
ADCR1: A/D conversion result register (ADCR1) value

Figure 12-8 shows the relation between the analog input voltage and the A/D conversion result.

Figure 12-8. Relation between Analog Input Voltage and A/D Conversion Result



12.4.3 A/D converter operation mode

The operation mode of the A/D converter is the select mode. One analog input channel is selected from among ANI0 to ANI4 with the analog input channel specification register (ADS1) and A/D conversion is performed.

The following two types of functions can be selected by setting the PFEN flag of the PFM register.

(1) Normal 8-bit A/D converter (PFEN = 0)
(2) Power-fail detection function (PFEN = 1)

(1) A/D conversion (when PFEN = 0)

When bit 7 (ADCS1) of the A/D converter mode register (ADM1) is set to 1 and bit 7 of the power-fail compare mode register (PFM) is set to 0, A/D conversion of the voltage applied to the analog input pin specified with the analog input channel specification register (ADS1) starts.

Upon the end of the A/D conversion, the conversion result is stored in the A/D conversion result register (ADCR1), and the interrupt request signal (INTAD) is generated. After one A/D conversion operation is started and ended, the next conversion operation is immediately started. A/D conversion operations are repeated until new data is written to ADS1.

If ADS1 is rewritten during A/D conversion operation, the A/D conversion operation under execution is stopped, and A/D conversion of a newly selected analog input channel is started.

If data with ADCS1 set to 0 is written to ADM1 during A/D conversion operation, the A/D conversion operation stops immediately.

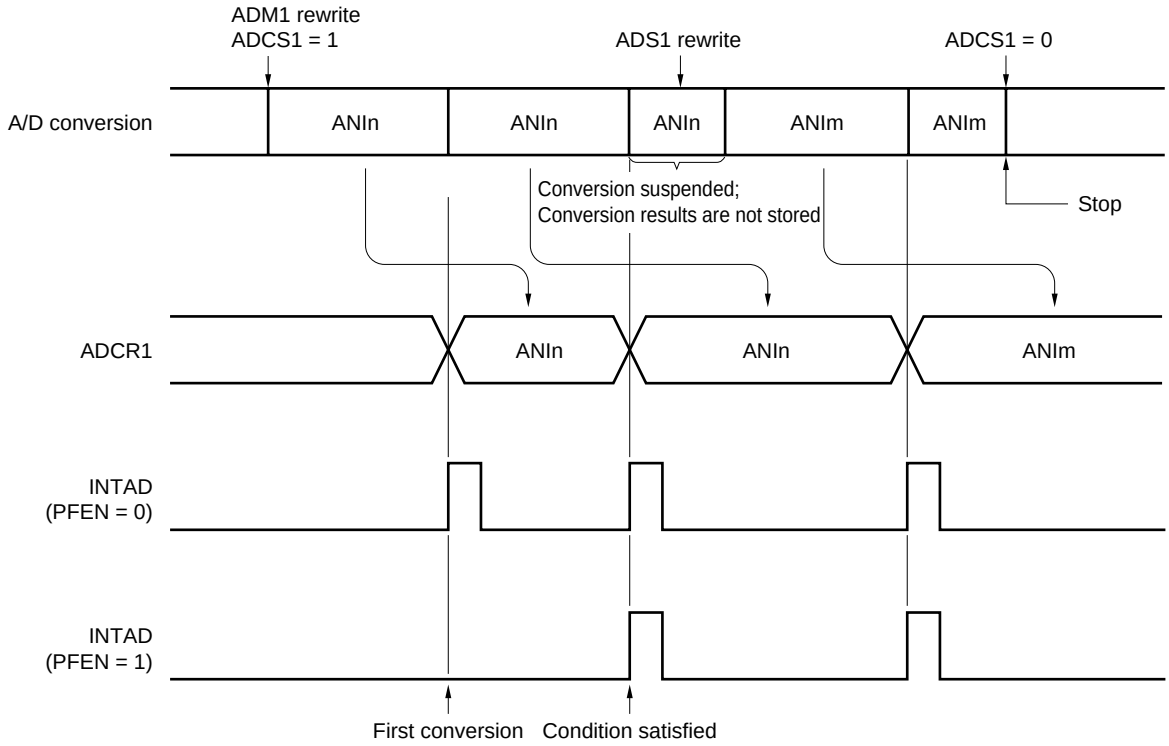
(2) Power-fail detection function (when PFEN = 1)

When bit 7 (ADCS1) of the A/D converter mode register (ADM1) and bit 7 (PFEN) of the power-fail compare mode register (PFM) are set to 1, A/D conversion of the voltage applied to the analog input pin specified with the analog input channel specification register (ADS1) starts.

Upon the end of the A/D conversion, the conversion result is stored in the A/D conversion result register (ADCR1), compared with the value of the power-fail compare threshold value register (PFT), and INTAD is generated under the condition specified by the PFCM flag of the PFM register.

Caution When executing power-fail comparison, the interrupt request signal (INTAD) is not generated on completion of the first conversion after ADCS1 has been set to 1. INTAD is valid from completion of the second conversion.

Figure 12-9. A/D Conversion



Remarks 1. n = 0, 1, ..., 4

2. m = 0, 1, ..., 4

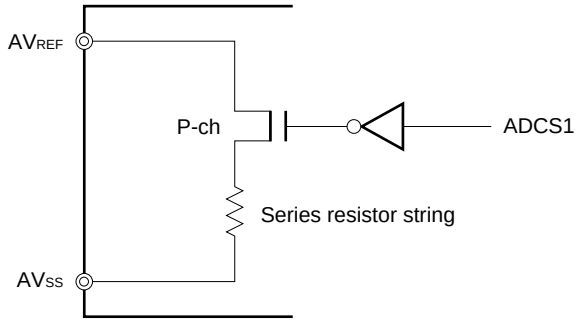
12.5 A/D Converter Cautions

(1) Current consumption in standby mode

A/D converter stops operating in the standby mode. At this time, current consumption can be reduced by setting bit 7 (ADCS1) of the A/D converter mode register (ADM1) to 0 to stop conversion.

Figure 12-10 shows how to reduce the current consumption in the standby mode.

Figure 12-10. Example of Method of Reducing Current Consumption in Standby Mode



(2) Input range of ANI0 to ANI4

Keep the input voltage of ANI0 to ANI4 within the rated range. If a voltage outside the rated range is input, the conversion value of that channel is undefined, and the values of the other channels may also be affected.

(3) Contending operations

<1> Contention between A/D conversion result register (ADCR1) write and ADCR1 read by instruction upon the end of conversion

ADCR1 read is given priority. After the read operation, the new conversion result is written to ADCR1.

<2> Contention between ADCR1 write and A/D converter mode register (ADM1) write or analog input channel specification register (ADS1) write upon the end of conversion

ADM1 or ADS1 write is given priority. ADCR1 write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

<3> If the A/D converter mode register (ADM1) or analog input channel specification register (ADS1) is written as soon as the A/D conversion end interrupt (INTAD) has occurred, the contents of the A/D conversion result register (ADCR1) will be undefined.

After an A/D conversion operation has been completed, therefore, read ADCR1 before writing data to ADM or ADS1.

(4) Starting conversion

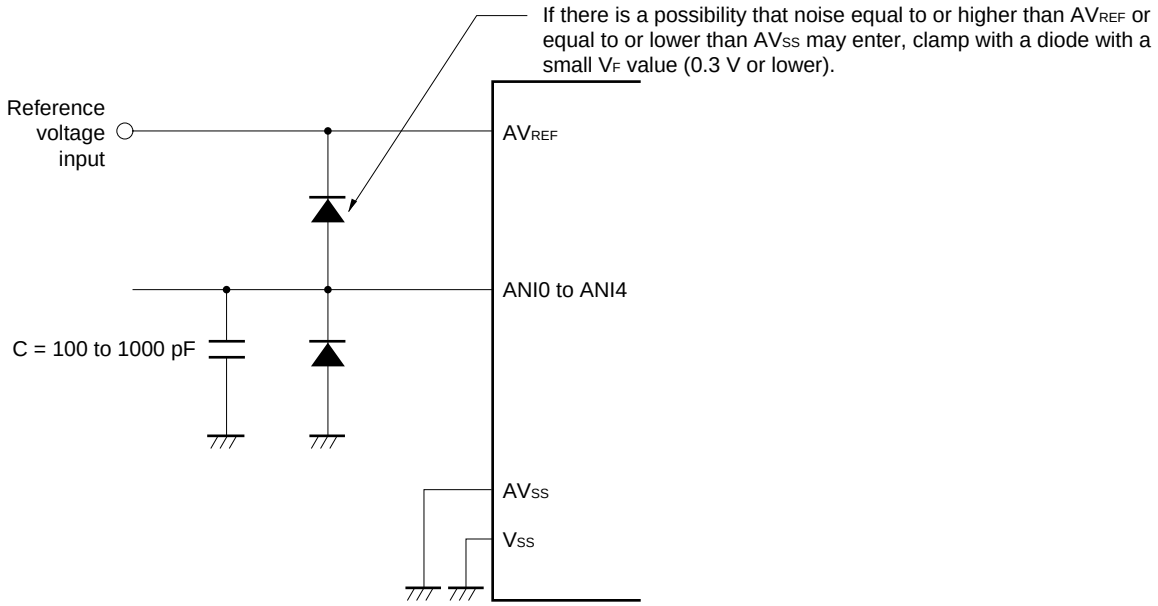
The first A/D conversion result after A/D conversion has been started by setting bit 7 (ADCS1) of the A/D conversion mode register (ADM1) to “1” may differ from the expected value.

Therefore, do not use the first conversion result immediately after A/D conversion has been started.

(5) Noise countermeasures

To maintain 8-bit resolution, attention must be paid to noise input to pin AV_{REF} and pins ANI0 to ANI4. Because the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally as shown in Figure 12-11 to reduce noise.

Figure 12-11. Analog Input Pin Connection



(6) ANI0 to ANI4

The analog input pins (ANI0 to ANI4) also function as input port pins (P10 to P14).

When A/D conversion is performed with any of pins ANI0 to ANI4 selected, do not execute a port input instruction while conversion is in progress, as this may reduce the conversion resolution.

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

(7) AV_{REF} pin input impedance

A series resistor string of approximately 21 kΩ is connected between the AV_{REF} pin and the AV_{SS} pin.

Therefore, if the output impedance of the reference voltage 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.

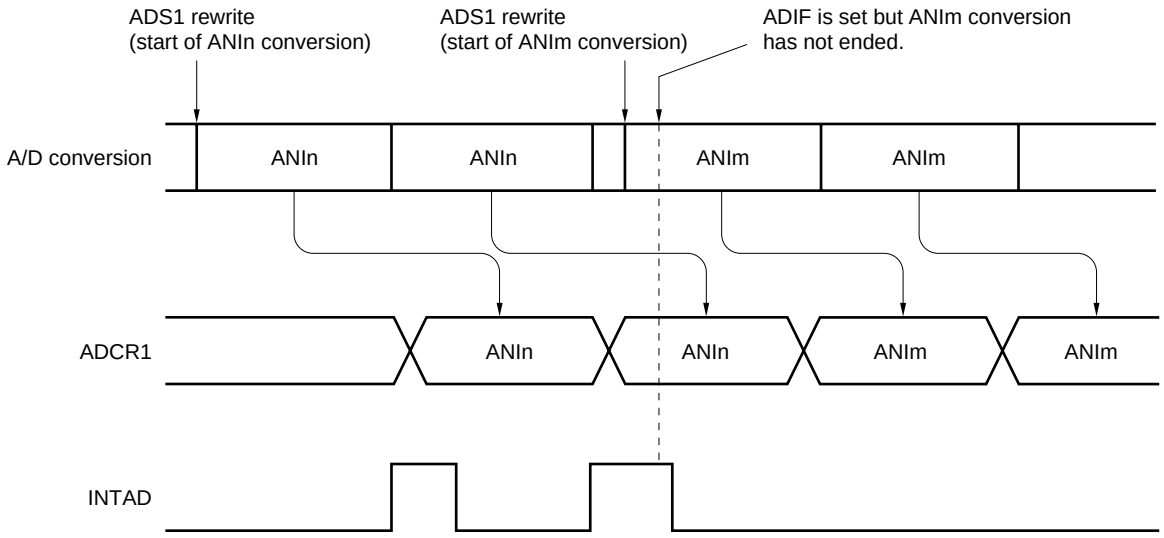
(8) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS1) 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 conversion end interrupt request flag for the pre-change analog input may be set just before the ADS1 rewrite, and when ADIF is read immediately after the ADS1 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 ADIF before the A/D conversion operation is resumed.

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



Remarks 1. n = 0, 1, ..., 4

2. m = 0, 1, ..., 4

(9) Read of A/D conversion result register (ADCR1)

When write operation is executed to A/D converter mode register (ADM1) and analog input channel specification register (ADS1), the contents of ADCR1 are undefined. Read the conversion result before write operation is executed to ADM1, ADS1. If a timing other than the above is used, the correct conversion result may not be read.

12.6 Cautions on Emulation

(1) D/A converter mode register (DAM1)

To perform debugging with an in-circuit emulator (IE-78K0-NS), the D/A converter mode register (DAM1) must be set. DAM1 is a register used to set a probe board (IE-780852-NS-EM4).

DAM1 is used when the power-fail detection function is used. Unless DAM1 is set, the power-fail detection function cannot be used. DAM1 is a write-only register.

Because the IE-780852-NS-EM4 uses an external analog comparator and a D/A converter to implement part of the power-fail detection function, the reference voltage must be controlled. Therefore, set bit 0 (DACE) of DAM1 to 1 when using the power-fail detection function.

Figure 12-13. D/A Converter Mode Register (DAM1) Format

Address: FF89H After Reset: 00H W

Symbol	7	6	5	4	3	2	1	0
DAM1	0	0	0	0	0	0	0	DACE

DACE	Reference Voltage Control
0	Disabled
1	Enabled (when power-fail detection function is used)

Cautions 1. DAM1 is a special register that must be set when debugging is performed with an in-circuit emulator. Even if this register is used, the operation of the μPD780852 Subseries is not affected. However, delete the instruction that manipulates this register from the program at the final stage of debugging.

2. Bits 7 to 1 must be set to 0.

(2) A/D converter of IE-780852-NS-EM4

The A/D converter of the IE-780852-NS-EM4 may not satisfy the rating of the first A/D conversion value right after A/D conversion has been started.

The above applies only to the IE-780852-NS-EM4 and does not affect the operation of the μPD780852 Subseries.

CHAPTER 13 SERIAL INTERFACE UART

13.1 Serial Interface Functions

The serial interface UART has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed to reduce power consumption.

For details, see **13.4.1 Operation stop mode**.

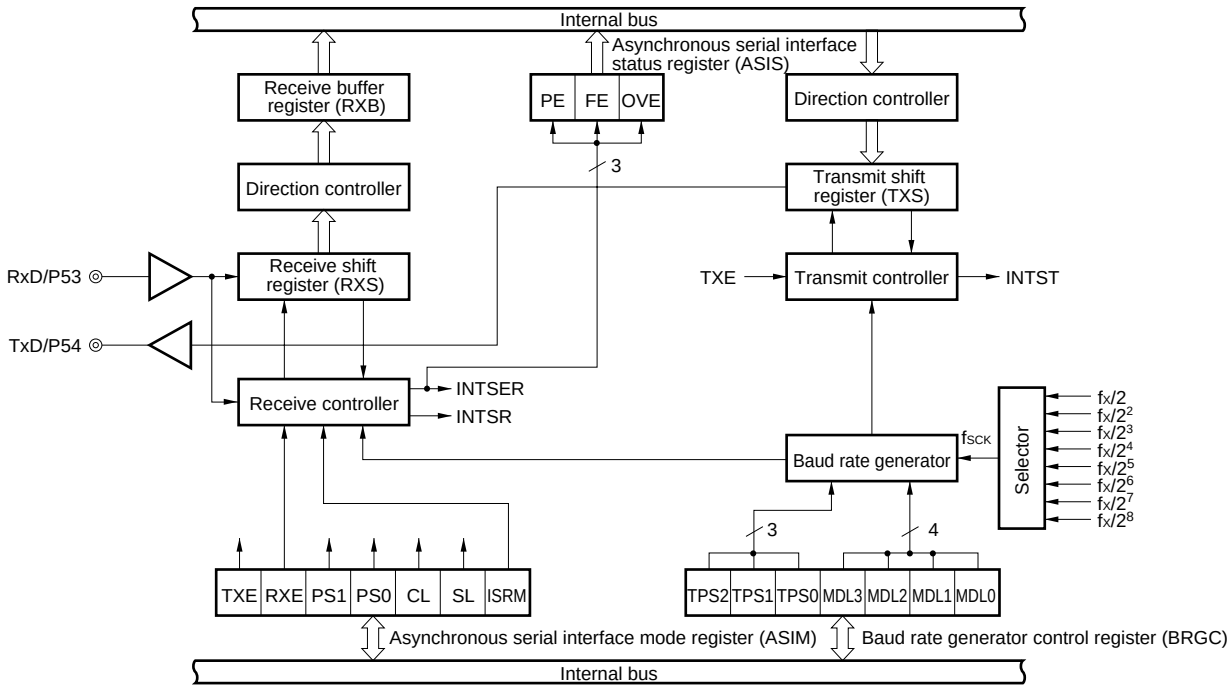
(2) Asynchronous serial interface (UART) mode

This mode enables full-duplex operation wherein one byte of data is transmitted and received after the start bit. The on-chip dedicated UART baud rate generator enables communications using a wide range of selectable baud rates.

For details, see **13.4.2 Asynchronous serial interface (UART) mode**.

Figure 13-1 shows the serial interface UART block diagram.

Figure 13-1. Serial Interface UART Block Diagram



13.2 Serial Interface Configuration

The serial interface UART consists of the following hardware.

Table 13-1. Serial Interface UART Configuration

Item	Configuration
Registers	Transmit shift register (TXS) Receive shift register (RXS) Receive buffer register (RXB)
Control registers	Asynchronous serial interface mode register (ASIM) Asynchronous serial interface status register (ASIS) Baud rate generator control register (BRGC)

(1) Transmit shift register (TXS)

This is the register for setting transmit data. Data written to TXS is transmitted as serial data.

When the data length is set as 7 bits, bits 0 to 6 of the data written to TXS are transmitted as transmit data. Writing data to TXS starts the transmit operation.

TXS is written with an 8-bit memory manipulation instruction. It cannot be read.

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

Caution Do not write to TXS during a transmit operation.
The same address is assigned to TXS and the receive buffer register (RXB). A read operation reads values from RXB.

(2) Receive shift register (RXS)

This register converts serial data input via the RxD pin to parallel data. When one byte of data is received at this register, the receive data is transferred to the receive buffer register (RXB).

RXS cannot be manipulated directly by a program.

(3) Receive buffer register (RXB)

This register is used to hold receive data. When one byte of data is received, one byte of new receive data is transferred from the receive shift register (RXS).

When the data length is set as 7 bits, receive data is transferred to bits 0 to 6 of RXB. In RXB, the MSB must be set to 0.

RXB is read with an 8-bit memory manipulation instruction. It cannot be written to.

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

Caution The same address is assigned to RXB and the transmit shift register (TXS). During a write operation, values are written to TXS.

(4) Transmit controller

The transmit controller controls transmit operations, such as adding a start bit, parity bit, and stop bit to data that is written to the transmit shift register (TXS), based on the values set to the asynchronous serial interface mode register (ASIM).

(5) Receive controller

The receive controller controls receive operations based on the values set to the asynchronous serial interface mode register (ASIM). During a receive operation, it performs error checking, such as for parity errors, and sets various values to the asynchronous serial interface status register (ASIS) according to the type of error that is detected.

13.3 Serial Interface Control Registers

The following three types of registers are used to control the serial interface UART.

- Asynchronous serial interface mode register (ASIM)
- Asynchronous serial interface status register (ASIS)
- Baud rate generator control register (BRGC)

(1) Asynchronous serial interface mode register (ASIM)

This is an 8-bit register that controls UART's serial transfer operations.

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

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

Figure 13-2 shows the format of ASIM.

Caution In UART mode, set the port mode register (PM5X) as follows. Besides that, set all output latches to 0.

- **When receiving**
  Set P53 (RxD) to the input mode (PM53 = 1)
- **When transmitting**
  Set P54 (TxD) to the output mode (PM54 = 0)
- **When transceiving**
  Set P53 to the input mode and P54 to the output mode

Figure 13-2. Asynchronous Serial Interface Mode Register (ASIM) Format

Address: FF85H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ASIM	TXE	RXE	PS1	PS0	CL	SL	ISRM	0

TXE	RXE	Operation Mode	RxD/P53 Pin Function	TxD/P54 Pin Function
0	0	Operation stop	Port function (P53)	Port function (P54)
0	1	UART mode (receive only)	Serial function (RxD)	Port function (P54)
1	0	UART mode (transmit only)	Port function (P53)	Serial function (TxD)
1	1	UART mode (transmit and receive)	Serial function (RxD)	Serial function (TxD)

PS1	PS0	Parity Bit Specification
0	0	No parity
0	1	Zero parity always added during transmission No parity detection during reception (parity errors do not occur)
1	0	Odd parity
1	1	Even parity

CL	Character Length Specification
0	7 bits
1	8 bits

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

ISRM	Receive Completion Interrupt Control When Error Occurs
0	Receive completion interrupt is issued when an error occurs
1	Receive completion interrupt is not issued when an error occurs

Cautions 1. Do not switch the operation mode until after the current serial transmit/receive operation has stopped.

2. Bit 0 must be set to 0.

(2) Asynchronous serial interface status register (ASIS)

When a receive error occurs during UART mode, this register indicates the type of error.

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

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

Figure 13-3. Asynchronous Serial Interface Status Register (ASIS) Format

Address: FF86H After Reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS	0	0	0	0	0	PE	FE	OVE

PE	Parity Error Flag
0	No parity error
1	Parity error (Transmit data parity does not match)

FE	Framing Error Flag
0	No framing error
1	Framing error **Note 1** (Stop bit not detected)

OVE	Overrun Error Flag
0	No overrun error
1	Overrun error **Note 2** (Next receive operation was completed before data was read from receive buffer register)

Notes 1. Even if a stop bit length of two bits has been set to bit 2 (SL) in the asynchronous serial interface mode register (ASIM), stop bit detection during a receive operation only applies to a stop bit length of 1 bit.

2. Be sure to read the contents of the receive buffer register (RXB) when an overrun error has occurred.

Until the contents of RXB are read, further overrun errors will occur when receiving data.

(3) Baud rate generator control register (BRGC)

This register sets the serial clock for UART.

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

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

Figure 13-4 shows the format of BRGC.

Figure 13-4. Baud Rate Generator Control Register (BRGC) Format

Address: FF87H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC	0	TPS2	TPS1	TPS0	MDL3	MDL2	MDL1	MDL0

(f_X = 8.38 MHz)

TPS2	TPS1	TPS0	Source Clock Selection for 5-bit Counter	n
0	0	0	$f_X/2$	1
0	0	1	$f_X/2^2$	2
0	1	0	$f_X/2^3$	3
0	1	1	$f_X/2^4$	4
1	0	0	$f_X/2^5$	5
1	0	1	$f_X/2^6$	6
1	1	0	$f_X/2^7$	7
1	1	1	$f_X/2^8$	8

MDL3	MDL2	MDL1	MDL0	Input Clock Selection for Baud Rate Generator	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	Setting prohibited	—

Caution Writing to BRGC during a communication operation may cause abnormal output from the baud rate generator and disable further communication operations. Therefore, do not write to BRGC during a communication operation.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{SCK}: Source clock for 5-bit counter

3. n: Value set via TPS0 to TPS2 ($1 \leq n \leq 8$)

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

13.4 Serial Interface Operations

This section explains the two modes of the serial interface UART.

13.4.1 Operation stop mode

This mode is used when serial transfers are not performed to reduce power consumption.

In the operation stop mode, P53/RxD and P54/TxD pins can be used as ordinary ports.

(1) Register settings

Operation stop mode is set with the asynchronous serial interface mode register (ASIM).

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

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

Address: FF85H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ASIM	TXE	RXE	PS1	PS0	CL	SL	ISRM	0

TXE	RXE	Operation Mode	RxD/P53 Pin Function	TxD/P54 Pin Function
0	0	Operation stop	Port function (P53)	Port function (P54)
0	1	UART mode (receive only)	Serial function (RxD)	Port function (P54)
1	0	UART mode (transmit only)	Port function (P53)	Serial function (TxD)
1	1	UART mode (transmit and receive)	Serial function (RxD)	Serial function (TxD)

Cautions 1. Do not switch the operation mode until after the current serial transmit/receive operation has stopped.

2. Bit 0 must be set to 0.

13.4.2 Asynchronous serial interface (UART) mode

This mode enables full-duplex operation wherein one byte of data is transmitted or received after the start bit.

The on-chip dedicated UART baud rate generator enables communications using a wide range of selectable baud rates.

The dedicated UART baud rate generator can also be used to generate a MIDI-standard baud rate (31.25 kbps).

(1) Register settings

UART mode is set with the asynchronous serial interface mode register (ASIM), asynchronous serial interface status register (ASIS), and the baud rate generator control register (BRGC).

(a) Asynchronous serial interface mode register (ASIM)

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

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

Caution In UART mode, set the port mode register (PM5X) as follows. Besides that, set all output latches to 0.

- **When receiving**
  Set P53 (RxD) to the input mode (PM53 = 1)
- **When transmitting**
  Set P54 (TxD) to the output mode (PM54 = 0)
- **When transceiving**
  Set P53 to the input mode and P54 to the output mode

Address: FF85H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ASIM	TXE	RXE	PS1	PS0	CL	SL	ISRM	0

TXE	RXE	Operation Mode	RxD/P53 Pin Function	TxD/P54 Pin Function
0	0	Operation stop	Port function (P53)	Port function (P54)
0	1	UART mode (receive only)	Serial function (RxD)	Port function (P54)
1	0	UART mode (transmit only)	Port function (P53)	Serial function (TxD)
1	1	UART mode (transmit and receive)	Serial function (RxD)	Serial function (TxD)

PS1	PS0	Parity Bit Specification
0	0	No parity
0	1	Zero parity always added during transmission No parity detection during reception (parity errors do not occur)
1	0	Odd parity
1	1	Even parity

CL	Character Length Specification
0	7 bits
1	8 bits

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

ISRM	Receive Completion Interrupt Control When Error Occurs
0	Receive completion interrupt is issued when an error occurs
1	Receive completion interrupt is not issued when an error occurs

Cautions 1. Do not switch the operation mode until after the current serial transmit/receive operation has stopped.

2. Bit 0 must be set to 0.

(b) Asynchronous serial interface status register (ASIS)

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

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

Address: FF86H After Reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS	0	0	0	0	0	PE	FE	OVE

PE	Parity Error Flag
0	No parity error
1	Parity error (Transmit data parity does not match)

FE	Framing Error Flag
0	No framing error
1	Framing error **Note 1** (Stop bit not detected)

OVE	Overrun Error Flag
0	No overrun error
1	Overrun error **Note 2** (Next receive operation was completed before data was read from receive buffer register)

Notes 1. Even if a stop bit length of two bits has been set to bit 2 (SL) in the asynchronous serial interface mode register (ASIM), stop bit detection during a receive operation only applies to a stop bit length of 1 bit.

2. Be sure to read the contents of the receive buffer register (RXB) when an overrun error has occurred.

Until the contents of RXB are read, further overrun errors will occur when receiving data.

(c) Baud rate generator control register (BRGC)

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

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

Address: FF87H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BRGC	0	TPS2	TPS1	TPS0	MDL3	MDL2	MDL1	MDL0

(f_X = 8.38 MHz)

TPS2	TPS1	TPS0	Source Clock Selection for 5-bit Counter	n
0	0	0	$f_X/2$	1
0	0	1	$f_X/2^2$	2
0	1	0	$f_X/2^3$	3
0	1	1	$f_X/2^4$	4
1	0	0	$f_X/2^5$	5
1	0	1	$f_X/2^6$	6
1	1	0	$f_X/2^7$	7
1	1	1	$f_X/2^8$	8

MDL3	MDL2	MDL1	MDL0	Input Clock Selection for Baud Rate Generator	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	Setting prohibited	—

Cautions 1. Writing to BRGC during a communication operation may cause abnormal output from the baud rate generator and disable further communication operations. Therefore, do not write to BRGC during a communication operation.

2. Bit 7 must be set to 0.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{SCK}: Source clock for 5-bit counter

3. n: Value set via TPS0 to TPS2 ($1 \leq n \leq 8$)

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

The transmit/receive clock that is used to generate the baud rate is obtained by dividing the main system clock.

- **Use of main system clock to generate a transmit/receive clock for baud rate**
  The main system clock is divided to generate the transmit/receive clock. The baud rate generated by the main system clock is determined according to the following formula.

$$[\text{Baud rate}] = \frac{f_X}{2^{n+1}(k+16)} \quad [\text{Hz}]$$

f_X: Main system clock oscillation frequency
n: Value set via TPS0 to TPS2 ($1 \leq n \leq 8$)
For details, see Table 13-2.
k: Value set via MDL0 to MDL3 ($0 \leq k \leq 14$)

Table 13-2 shows the relation between the 5-bit counter's source clock assigned to bits 4 to 6 (TPS0 to TPS2) of BRGC and the "n" value in the above formula.

Table 13-2. Relation between 5-Bit Counter's Source Clock and "n" Value

TPS2	TPS1	TPS0	5-bit Counter's Source Clock Selection	n
0	0	0	$f_X/2$	1
0	0	1	$f_X/2^2$	2
0	1	0	$f_X/2^3$	3
0	1	1	$f_X/2^4$	4
1	0	0	$f_X/2^5$	5
1	0	1	$f_X/2^6$	6
1	1	0	$f_X/2^7$	7
1	1	1	$f_X/2^8$	8

Remark f_X: Main system clock oscillation frequency (f_X = 8.38 MHz)

- **Error tolerance range for baud rates**

  The tolerance range for baud rates depends on the number of bits per frame and the counter's division rate [1/(16 + k)].

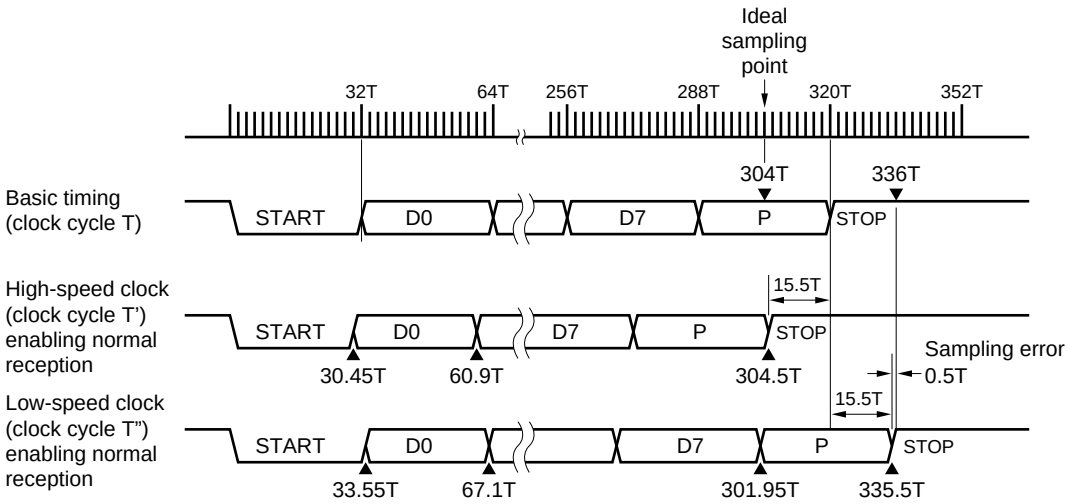
  Table 13-3 describes the relation between the main system clock and the baud rate and Figure 13-5 shows an example of a baud rate error tolerance range.

Table 13-3. Relation between Main System Clock and Baud Rate

Baud Rate (bps)	f_X = 8.386 MHz	
	BRGC Set Value	Error (%)
600	7BH	1.10
1,200	6BH	1.10
2,400	5BH	1.10
4,800	4BH	1.10
9,600	3BH	1.10
19,200	2BH	–1.3
31,250	21H	1.10
38,400	1BH	1.10
76,800	0BH	1.10
115,200	01H	1.03

Remark f_X: Main system clock oscillation frequency

Figure 13-5. Error Tolerance (When k = 0) Including Sampling Errors



Remark T: 5-bit counter's source clock cycle

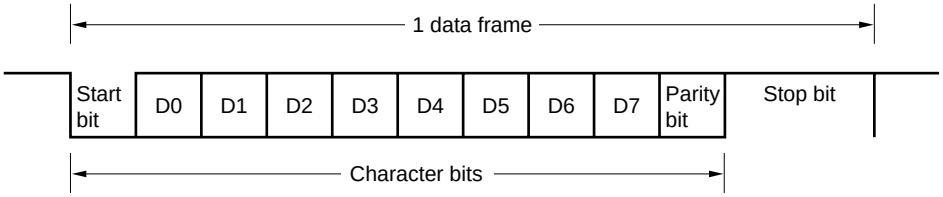
$$\text{Baud rate error tolerance (when k = 0)} = \frac{\pm 15.5}{320} \times 100 = 4.8438\ (\%)$$

(2) Communication operations

(a) Data format

Figure 13-6 shows the transmit/receive data format.

Figure 13-6. Format of Transmit/Receive Data in Asynchronous Serial Interface



One data frame consists of the following each bit.

- Start bit ..... 1 bit
- Character bits ... 7 bits or 8 bits
- Parity bit ..... Even parity, odd parity, zero parity, or no parity
- Stop bit(s) ..... 1 bit or 2 bits

The asynchronous serial interface mode register (ASIM) is used to set the character bit length, parity selection, and stop bit length within each data frame.

When “7 bits” is selected as the number of character bits, only the lower 7 bits (bits 0 to 6) are valid, so that during a transmission the highest bit (bit 7) is ignored and during reception the highest bit (bit 7) must be set to “0”.

The asynchronous serial interface mode register (ASIM) and the baud rate generator control register (BRGC) are used to set the serial transfer rate.

If a receive error occurs, information about the receive error can be recognized by reading the asynchronous serial interface status register (ASIS).

(b) Parity types and operations

The parity bit is used to detect bit errors in transfer data. Usually, the same type of parity bit is used by the transmitting and receiving sides. When odd parity or even parity is set, errors in the parity bit (the odd-number bit) can be detected. When zero parity or no parity is set, errors are not detected.

(i) Even parity

- During transmission
  The number of bits in transmit data that includes a parity bit is controlled so that there are an even number of “1” bits. The value of the parity bit is as follows.

  If the transmit data contains an odd number of “1” bits: the parity bit value is “1”
  If the transmit data contains an even number of “1” bits: the parity bit value is “0”

- During reception
  The number of “1” bits is counted among the receive data that include a parity bit, and a parity error occurs when the result is an odd number.

(ii) Odd parity

- During transmission
  The number of bits in transmit data that includes a parity bit is controlled so that there is an odd number of “1” bits. The value of the parity bit is as follows.

  If the transmit data contains an odd number of “1” bits: the parity bit value is “0”
  If the transmit data contains an even number of “1” bits: the parity bit value is “1”

- During reception
  The number of “1” bits is counted among the receive data that include a parity bit, and a parity error occurs when the result is an even number.

(iii) Zero parity

During transmission, the parity bit is set to “0” regardless of the transmit data.
During reception, the parity bit is not checked. Therefore, no parity errors will occur regardless of whether the parity bit is a “0” or a “1”.

(iv) No parity

No parity bit is added to the transmit data.
During reception, receive data is regarded as having no parity bit. Since there is no parity bit, no parity errors will occur.

(c) Transmission

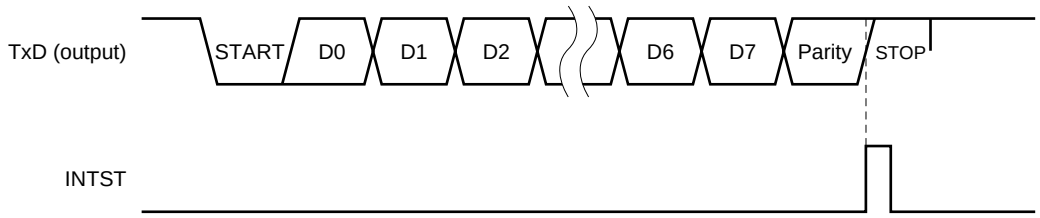
The transmit operation is started when transmit data is written to the transmit shift register (TXS). A start bit, parity bit, and stop bit(s) are automatically added to the data.

Starting the transmit operation shifts out the data in TXS, thereby emptying TXS, after which a transmit completion interrupt (INTST) is issued.

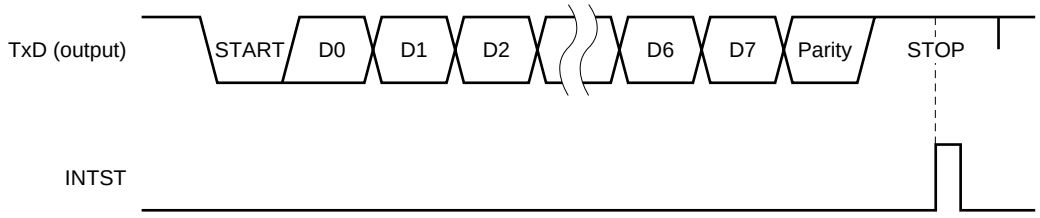
The timing of the transmit completion interrupt is shown in Figure 13-7.

Figure 13-7. Asynchronous Serial Interface Transmit Completion Interrupt Timing

(i) Stop bit length: 1 bit



(ii) Stop bit length: 2 bits



Caution Do not rewrite the asynchronous serial interface mode register (ASIM) during a transmit operation. Rewriting to the ASIM register during a transmit operation may disable further transmit operations (in such case, enter a $\overline{\text{RESET}}$ to restore normal operation). Whether or not a transmit operation is in progress can be determined by software using the transmit completion interrupt (INTST) or the interrupt request flag (STIF) that is set by INTST.

(d) Reception

The receive operation is enabled when “1” is set to bit 6 (RXE) of the asynchronous serial interface mode register (ASIM), and input via the RxD pin is sampled.

The serial clock specified by ASIM is used when sampling the RxD pin.

When the RxD pin goes low, the 5-bit counter of the baud rate generator begins counting and the start timing signal for data sampling is output when half of the specified baud rate time has elapsed. If sampling the RxD pin input with this start timing signal yields a low-level result, a start bit is recognized, after which the 5-bit counter is initialized and starts counting and data sampling begins. After the start bit is recognized, the character data, parity bit, and one-bit stop bit are detected, at which point reception of one data frame is completed.

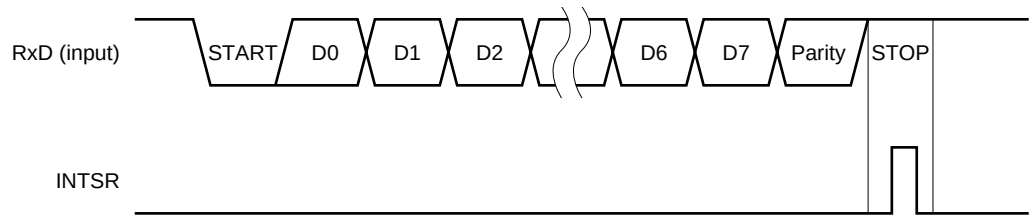
Once reception of one data frame is completed, the receive data in the shift register is transferred to the receive buffer register (RXB) and a receive completion interrupt (INTSR) occurs.

Even if an error has occurred, the receive data in which the error occurred is still transferred to RXB. INTSR occurs if bit 1 (ISRM) of ASIM is cleared to 0 on occurrence of an error. If the ISRM bit is set to 1, INTSR does not occur (see Figure 13-9).

If the RXE bit is reset (to “0”) during a receive operation, the receive operation is stopped immediately. At this time, the contents of RXB and ASIS do not change, nor does INTSR or INTSER occur.

Figure 13-8 shows the timing of the asynchronous serial interface receive completion interrupt.

Figure 13-8. Asynchronous Serial Interface Receive Completion Interrupt Timing



Caution Be sure to read the contents of the receive buffer register (RXB) even when a receive error has occurred. Overrun errors will occur during the next data receive operations and the receive error status will remain until the contents of RXB are read.

(e) Receive errors

Three types of errors can occur during a receive operation: parity error, framing error, or overrun error. If, as the result of data reception, an error flag is set to the asynchronous serial interface status register (ASIS), a receive error interrupt (INTSER) will occur. Receive error interrupts are generated before receive interrupt requests (INTSR). Table 13-4 lists the causes behind receive errors.

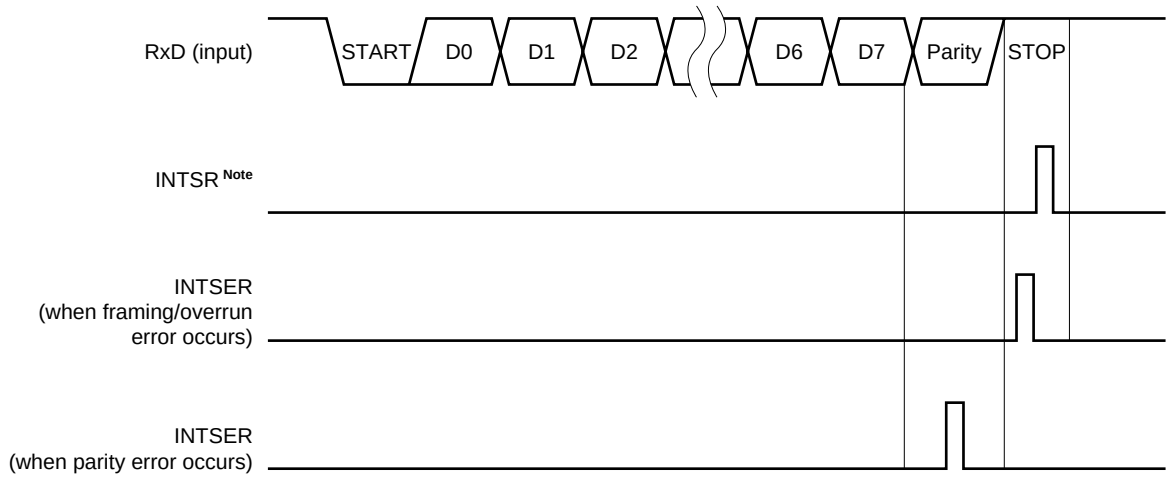
As part of receive error interrupt servicing (INTSER), the contents of ASIS can be read to determine which type of error occurred during the receive operation (see Table 13-4 and Figure 13-9).

The contents of ASIS are reset (to "0") when the receive buffer register (RXB) is read or when the next data is received (if the next data contains an error, another error flag will be set).

Table 13-4. Causes of Receive Errors

Receive Error	Cause	ASIS Value
Parity error	Parity specified during transmission does not match parity of receive data	04H
Framing error	Stop bit was not detected	02H
Overrun error	Reception of the next data was completed before data was read from the receive buffer register	01H

Figure 13-9. Receive Error Timing



Note If a reception error occurs when ISRM bit is set to 1, INTSR does not occur.

Cautions 1. The contents of asynchronous serial interface status register (ASIS) are reset (to "0") when the receive buffer register (RXB) is read or when the next data is received. To obtain information about the error, be sure to read the contents of ASIS before reading RXB.

2. Be sure to read the contents of the receive buffer register (RXB) even when a receive error has occurred. Overrun errors will occur during the next data receive operations and the receive error status will remain until the contents of RXB are read.

[MEMO]

CHAPTER 14 SERIAL INTERFACE SIO2

14.1 Serial Interface Functions

The serial interface SIO2 has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. For details, see **14.4.1 Operation stop mode**.

(2) 3-wire serial I/O mode (fixed as MSB first)

This is an 8-bit data transfer mode using three lines: a serial clock line ($\overline{SCK2}$), a serial output line (SO2), and a serial input line (SI2).

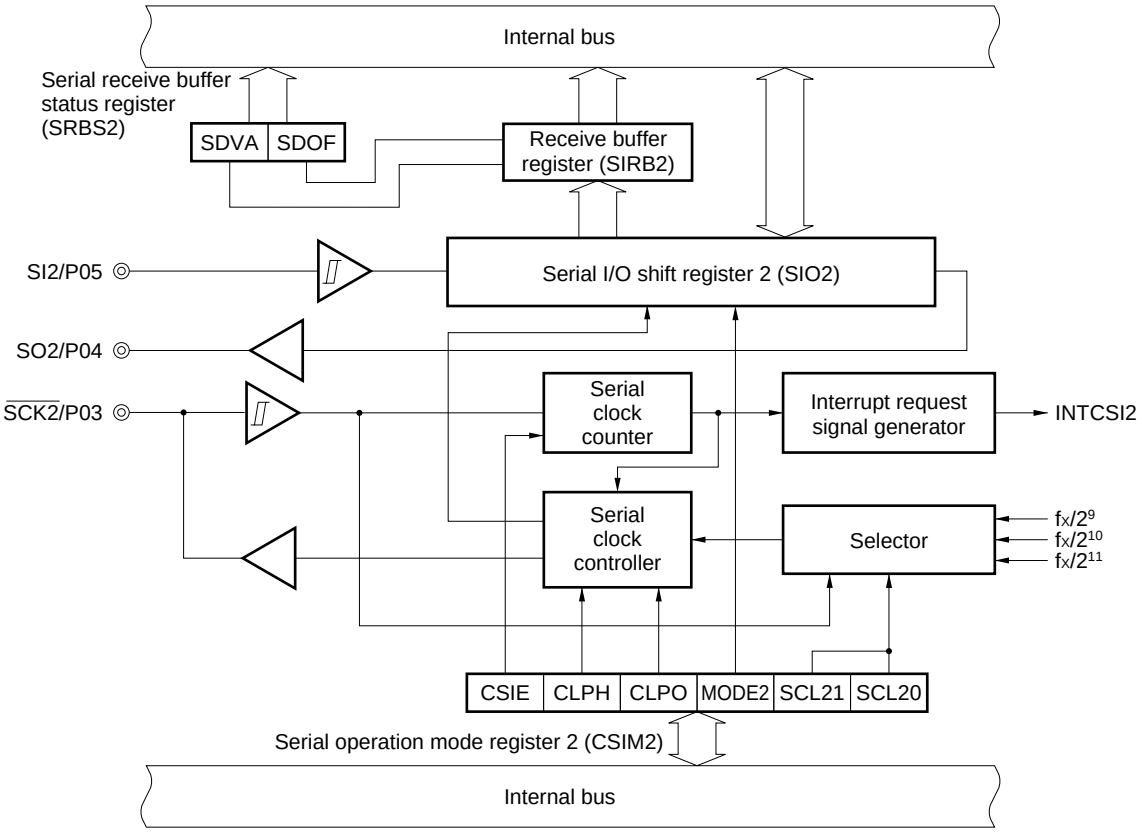
Since simultaneous transmit and receive operations are enabled in the 3-wire serial I/O mode, the processing time for data transfers is reduced.

The first bit in the 8-bit data in serial transfer is fixed as the MSB. This data contains a 1-byte receive buffer, and can be received successively. The serial clock and the data phase/polarity can be selected.

The 3-wire serial I/O mode is useful for connection to a peripheral I/O device that includes a clocked serial interface, a display controller, etc.

Figure 14-1 shows the serial interface SIO2 block diagram.

Figure 14-1. Serial Interface SIO2 Block Diagram



14.2 Serial Interface Configuration

The serial interface SIO2 consists of the following hardware.

Table 14-1. Serial Interface SIO2 Configuration

Item	Configuration
Registers	Serial I/O shift register 2 (SIO2) Serial receive data buffer register (SIRB2)
Control registers	Serial operation mode register 2 (CSIM2) Serial receive data buffer status register (SRBS2) Port mode register 0 (PM0)

(1) Serial I/O shift register 2 (SIO2)

This is an 8-bit register that performs parallel-serial conversion and serial transmit/receive (shift operations) in synchronization with the serial clock.

SIO2 is set with an 8-bit memory manipulation instruction.

A transmit/receive operation is started by writing or reading data to or from SIO2 when bit 7 (CSIE2) of the serial operation mode register 2 (CSIM2) is 1.

When the received data is completely stored in SIO2, if SDVA (bit 1 of the receive data buffer status register (SRBS2)) is 0, the contents of SIO2 are immediately transferred to the receive data buffer register (SIRB2). If SDVA is 1, the received data is held in SIO2.

$\overline{\text{RESET}}$ input clears SIO2 to 00H.

Cautions 1. Do not access (read/write) SIO2 during a transmit/receive operation (shift operation).

2. When a transmit/receive operation starts (writing to SIO2), do not access (read/write) SIO2 before a transmit completion interrupt (INTCSI2) occurs in the transmit/receive mode (MODE2 = 1).

3. If the external clock mode (CLPH = 1) is selected in the slave mode (SCL20 = 0, SCL21 = 0), do not read the data of SIO2 directly. The value of SIO2 may not coincide with the value transferred to SIRB2. To obtain the accurate value, read the data of SIRB2.

(2) Serial receive data buffer register (SIRB2)

This is an 8-bit register that stores the data transferred from the serial I/O shift register 2 (SIO2).

The contents of SIO2 are immediately transferred to SIRB2 when SDVA (bit 1 of the receive data buffer status register (SRBS2)) = 0. When SDVA = 1, the contents of SIO2 are not transferred to SIRB2, and the receive data is held by SIO2.

The status of SIRB2 can be checked by using the serial receive data buffer status register (SRBS2). If an overflow occurs, the value of SIRB2 does not change after SRBS2 has been read, until transfer of the new data has been completed.

SIRB2 can be read with an 8-bit memory manipulation instruction. It cannot be written to.

$\overline{\text{RESET}}$ input makes SIRB2 to undefined.

14.3 Serial Interface Control Registers

The following three types of registers are used to control the serial interface SIO2.

- Serial operation mode register 2 (CSIM2)
- Serial receive data buffer status register (SRBS2)
- Port mode register 0 (PM0)

(1) Serial operation mode register 2 (CSIM2)

This register is used to set the SIO2 interface's serial clock, operation mode, and operation enable/disable.

CSIM2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM2 to 00H.

Figure 14-2. Serial Operation Mode Register 2 (CSIM2) Format

Address: FF98H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM2	CSIE2	0	0	CLPH	CLPO	MODE2	SCL21	SCL20

CSIE2	SIO2 Operation Enable/Disable Specification		
	Shift Register Operation	Serial Counter	Port
0	Operation disabled	Clear	Port function
1	Operation enabled	Counter operation enabled	Serial function + port function

CLPH	SO2 Output Timing Selection
0	Transfer starts at the first active edge of $\overline{\text{SCK2}}$.
1	Transfer starts when $\overline{\text{SCK2}}$ is written.

CLPO	Serial Clock Active Level Selection
0	$\overline{\text{SCK2}}$ is high while serial transfer is stopped.
1	$\overline{\text{SCK2}}$ is low while serial transfer is stopped.

MODE2	SIO2 Operation Mode Setting		
	Operation Mode	Transfer Start Trigger	SO2/P04 Pin
0	Transmit/receive mode	Writing to SIO2	SO output
1	Receive-only mode	Reading from SIO2	Port function

SCL21	SCL20	Serial Transfer Operation Clock Selection	
		Operation Clock (Transfer Frequency)	Master/Slave
0	0	External clock input to $\overline{\text{SCK2}}$	Slave mode
0	1	$f_X/2^9$	Master mode
1	0	$f_X/2^{10}$	Master mode
1	1	$f_X/2^{11}$	Master mode

Cautions 1. Bits 5 and 6 must be set to 0.

2. While a serial transfer operation is enabled (CSIE2 = 1), be sure to stop the serial transfer operation once before changing the values of bits other than CSIE2 to different data.

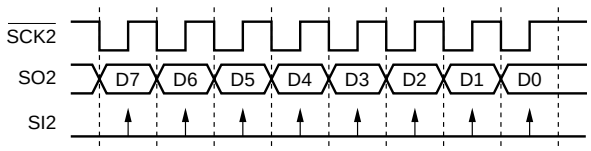
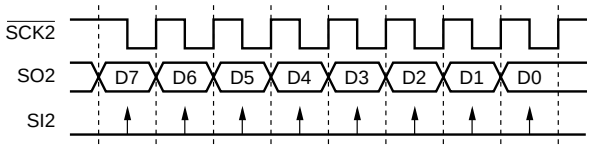
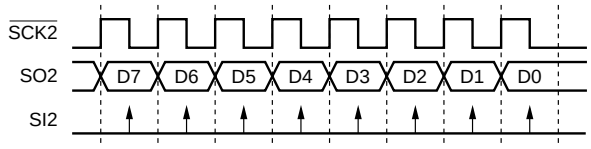
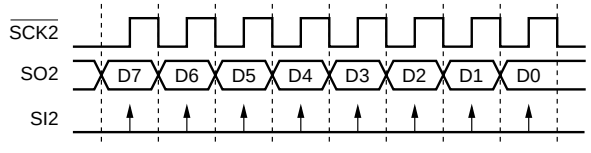
3. When operation is disabled (CSIE2 = 0) during a serial transfer operation, the operation will be stopped immediately. At this time, even if operation is enabled again (CSIE2 = 1) after it was once stopped, the operation will not start. To resume operation, set operation enable (CSIE2 = 1) and then execute an access that will be the start trigger of each transfer operation mode.

4. Changing CSIE2 and other bits at the same time is prohibited. After clearing CSIE2 to 0, change the other bits.

Remark fx: Main system clock oscillation frequency

The following shows the relationships between the CLPO and CLPH settings, and the serial transfer clock, data output, and input data capture timing.

Figure 14-3. Serial Transfer Operation Timing According to CLPO and CLPH Settings

CLPO	CLPH	Serial Transfer Operation Clock Selection
0	0	
0	1	
1	0	
1	1	

Remarks 1. $\overline{\text{SCK2}}$: Serial transfer clock

2. SO2: Data output timing

3. SI2: Input data capture timing

(2) Serial receive data buffer status register (SRBS2)

This register is used to indicate the status of serial receive data buffer register (SIRB2).

SRBS2 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears SRBS2 to 00H.

Figure 14-4. Serial Receive Data Buffer Status Register (SRBS2) Format

Address: FF9AH After Reset: 00H R

Symbol	7	6	5	4	3	2	1	0
SRBS2	0	0	0	0	0	0	SDVA	SDOF

SDVA	Receive Data Status Check
0	All data in SIRB2 has been read. This bit is cleared to 0 when SIRB2 has been read.
1	Data in SIRB2 has not been read. This bit is set to 1 when all receive data has been transferred from SIO2 to SIRB2.

SDOF	Overflow Check When Serial Data Is Transferred
0	No overflow error. All data in SIRB2 has been read. This bit is cleared to 0 when SIRB2 has been read.
1	Overflow error occurs. This bit is set to 1 if receive data is set in SIRB2 and if the next reception operation has been completed before that data is read (if data is transferred to SIO2).

Cautions 1. When an overflow error occurs, receive data in SIO2 will not be transferred to SIRB2 even if the next receive operation for SIO2 is complete.

2. When an overflow error occurs, be sure to read SRBS2 (clear SDOF), and read SIRB2 (clear SDVA). If the receive operation is resumed without reading SIRB2 (clearing SDVA) after SDOF clear, SDOF is set even if the next receive operation ends normally.

3. Even if an overflow error has occurred, new receive data can be received by SIO2. At this time, a transmit completion interrupt (INTCSI2) occurs.

(a) Serial data valid flag (SDVA)

This flag indicates that the serial receive data buffer register (SIRB2) has not been completely read. It is set to 1 when the receive data has been completely transferred from the serial I/O shift register 2 (SIO2) to SIRB2.

SDVA is cleared to 0 when SIRB2 has been read. If SIRB2 is accessed for read, SDVA remains cleared (to 0) until the next receive data is transferred from SIO2 to SIRB2.

(b) Overflow flag (SDOF)

This flag indicates whether an overflow error occurs on the serial receive data buffer register (SIRB2). It is automatically set to 1 to prevent a loss of receive data if data that has not yet been read remains in SIRB2 (SDVA = 1) and if the next data has been transferred to SIO2.

(3) Port mode register 0 (PM0)

This register is used to specify the input/output of port 0 in 1-bit units.

When using the P03/$\overline{SCK2}$, P04/SO2, and P05/SI2 pins in the 3-wire serial I/O mode, set PM03 to PM05 as shown in Table 14-2 below. Set the output latches of P03 to P05 to 0.

PM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input sets PM0 to FFH.

Figure 14-5. Port Mode Register 0 (PM0) Format

Address: FF20H After Reset: FFH R/W

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

PM0n	P0n Pin Input/Output Mode Selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark n = 0 to 7

Table 14-2. Relation between Operation Modes and Settings of PM03 to PM05

Operation Mode			PM0 Settings Note 2		
SIO2 Operation	Serial Operation Mode	Master/Slave Note 1	PM03	PM04	PM05
Disabled (CSIE2 = 0)	—	—	×	×	×
Enabled (CSIE2 = 1)	Receive-only mode (MODE2 = 0)	Master	0	×	1
		Slave	1	×	1
	Transmit/receive mode (MODE2 = 1)	Master	0	0	1
		Slave	1	0	1

Notes 1. Master/slave can be selected by setting bits 0 and 1 (SCL20 and SCL21) of the serial operation mode register 2 (CSIM2).

2. 0: Output mode
1: Input mode
×: don't care (can be used as an ordinary port pin)

14.4 Serial Interface Operations

This section explains the two modes of the serial interface SIO2.

14.4.1 Operation stop mode

This mode is used when serial transfers are not performed to reduce power consumption.

In addition, in this mode, the P03/$\overline{SCK2}$, P04/SO2, and P05/SI2 pins can be used as normal I/O port pins.

(1) Register setting

The operation stop mode is set with the serial operation mode register 2 (CSIM2).

CSIM2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input clears CSIM2 to 00H.

Address: FF98H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM2	CSIE2	0	0	CLPH	CLPO	MODE2	SCL21	SCL20

CSIE2	SIO2 Operation Enable/Disable Specification		
	Shift Register Operation	Serial Counter	Port
0	Operation disabled	Clear	Port function
1	Operation enabled	Counter operation enabled	Serial function + port function

14.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is useful when connecting to devices such as peripheral I/Os and display controllers, which incorporate a clocked serial interface.

This mode executes data transfer via three lines: a serial clock line ($\overline{SCK2}$), a serial output line (SO2), and a serial input line (SI2).

(1) Register settings

The 3-wire serial I/O mode is set with the serial operation mode register 2 (CSIM2).

CSIM2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input clears CSIM2 to 00H.

Address: FF98H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM2	CSIE2	0	0	CLPH	CLPO	MODE2	SCL21	SCL20

CSIE2	SIO2 Operation Enable/Disable Specification		
	Shift Register Operation	Serial Counter	Port
0	Operation disabled	Clear	Port function
1	Operation enabled	Counter operation enabled	Serial function + port function

CLPH	SO2 Output Timing Sleleciton
0	Transfer starts at the first active edge of $\overline{SCK2}$.
1	Transfer starts when $\overline{SCK2}$ is written.

CLPO	Serial Clock Active Level Selection
0	$\overline{SCK2}$ is high while serial transfer is stopped.
1	$\overline{SCK2}$ is low while serial transfer is stopped.

MODE2	SIO2 Operation Mode Setting		
	Operation Mode	Transfer Start Trigger	SO2/P04 Pin
0	Transmit/receive mode	Writing to SIO2	SO output
1	Receive-only mode	Reading from SIO2	Port function

SCL21	SCL20	Serial Transfer Operation Clock Selection	
		Operation Clock (Transfer Frequency)	Master/Slave
0	0	External clock input to $\overline{SCK2}$	Slave mode
0	1	$f_X/2^9$	Master mode
1	0	$f_X/2^{10}$	Master mode
1	1	$f_X/2^{11}$	Master mode

Cautions 1. Bits 5 and 6 must be set to 0.

2. While a serial transfer operation is enabled (CSIE2 = 1), be sure to stop the serial transfer operation once before changing the values of bits other than CSIE2 to different data.

3. When operation is disabled (CSIE2 = 0) during a serial transfer operation, the operation will be sopped immediately. At this time, even if operation is enabled again (CSIE2 = 1) after it was once stopped, the operation will not start. To resume operation, set operation enable (CSIE2 = 1) and then execute an access that will be the start trigger of each transfer operation mode.

4. Changing CSIE2 and other bits at the same time is prohibited. After clearing CSIE2 to 0, change the other bits.

Remark fx: Main system clock oscillation frequency

(2) Communication operations

Data is transmitted/received in 8-bit units. 8-bit data is transmitted/received bit by bit in synchronization with the serial clock.

Two transfer modes are provided for the 3-wire serial I/O mode: transmit/receive mode and receive-only mode.

After setting each operation mode using the serial operation mode register (CSIM2), transmit/receive operation is started by performing an operation that will be the start trigger.

(3) Transfer start

A serial transfer starts when the following conditions have been satisfied.

- **Receive-only mode**
  When CSIE2 = 1 and MODE2 = 1, transfer starts when writing to SIO2.

- **Transmit/receive mode**
  When CSIE2 = 1 and MODE2 = 0, transfer starts when reading from SIO2.

Caution After data has been written to SIO2, transfer will not start even if the CSIE2 bit value is set to “1”.

Completion of an 8-bit transfer automatically stops the serial transfer operation and sets a serial transfer completion flag.

(4) Operation mode

(a) Master mode (with internal clock $\overline{\text{SCK2}}$)

Serial interface SIO2 operates in the master mode (with internal clock $\overline{\text{SCK2}}$) if bits 1 and 0 (SCL21 and SCL20) of the serial operation mode register 2 (CSIM2) are set to (0, 1), (1, 0), or (1, 1).

Transfer is started when data has been read from or written to the serial I/O shift register 2 (SIO2). The serial data is output from the SO2 pin in synchronization with the serial clock. Select the operating clock for serial transfer by using SCL21 and SCL20.

Transfer is completed and a transmit completion interrupt (INTCSI2) occurs when all the 8 bits of the serial data have been completely transferred.

(b) Slave mode (with external clock)

Serial interface SIO2 operates in the slave mode (with an external clock) if bits 1 and 0 (SCL21 and SCL20) of the serial operation mode register 2 (CSIM2) are set to (0, 0). In the slave mode, the $\overline{\text{SCK2}}$ pin operates as an external serial clock input pin.

Serial data is transferred to SIO2 in synchronization with the externally input serial clock. After the serial data has been received by SIO2, it is transferred to the serial receive data buffer register (SIRB2). At the same time, the SDVA flag is set to 1 and a transmit completion interrupt (INTCSI2) occurs.

Caution To prevent the occurrence of an overflow error, read the value of SIRB2 before the next serial data is transferred to SIO2.

Table 14-3 below shows the status of the P03/$\overline{\text{SCK2}}$, P04/SO2, and P05/SI2 pins in each operation mode.

Table 14-3. Operation Mode and Pin Status

Operation Mode			Pin		
SIO2 Operation	Serial Operation Mode	Master/Slave Note	P03/$\overline{\text{SCK2}}$	P04/SO2	P05/SI2
Disabled (CSIE2 = 0)	—	—	Port function	Port function	Port function
Enabled (CSIE2 = 1)	Receive-only mode (MODE2 = 1)	Master	Serial function	Port function	Hi-Z
		Slave	Hi-Z	Port function	Hi-Z
	Transmit/receive mode (MODE2 = 0)	Master	Serial function	Serial function	Hi-Z
		Slave	Hi-Z	Serial function	Hi-Z

Note Master/slave can be selected by setting bits 0 and 1 (CSK1 and SCL20) of the serial operation mode register 2 (CSIM2).

(5) Transfer format

A simultaneous transmit/receive operation can be performed when the receive data is transferred from the serial I/O shift register 2 (SIO2) to the receive data buffer register (SIRB2).

(a) Clock phase and polarity

The phase and polarity of the serial clock can be selected from four combinations by setting bits 3 and 4 (CLPO and CLPH) of the serial operation mode register 2 (CSIM2).

Select a clock polarity by using CLPO. An active-high or active-low clock can be selected.

The clock phase is set by CLPH. The output timing of SO2 can be selected.

For the setting of CLPO and CLPH, serial transfer clock, data output, and the capture timing of input data, see Figure 14-3.

(b) Transfer format when CLPH = 0

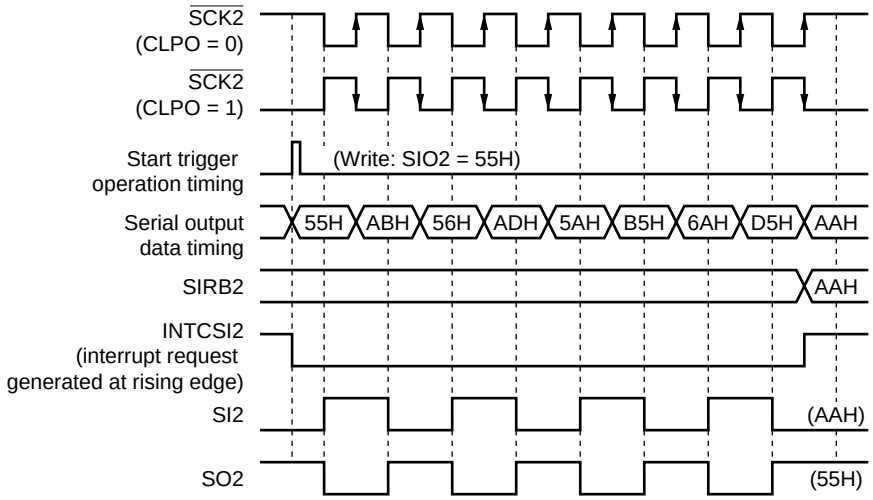
Figure 14-6 shows the operation timing when CLPH = 0. Two waves of $\overline{SCK2}$, when CLPO = 0 and when CLPO = 1, are shown in the figure.

Data is transmitted or received in 8-bit units. Each bit of data is transmitted or received in synchronization with the serial clock.

SIO2 is shifted at the falling edge of $\overline{SCK2}$ if CLPH = 0 and CLPO = 0. If CLPH = 0 and CLPO = 1, SIO2 is shifted at the rising edge of $\overline{SCK2}$. The transmit data is held by the SO2 latch and output from the SO2 pin. The receive data input to the SI2 pin is latched to SIO2 at the rising edge of $\overline{SCK2}$ (if CLPH = 0 and CLPO = 0).

Completion of an 8-bit transfer automatically stops operation of SIO2 and sets a serial transfer completion flag.

Figure 14-6. Operation Timing When CLPH Is Set to 0
(Serial output data: 55H, serial input data: AAH)



(c) Transfer format when CLPH = 1

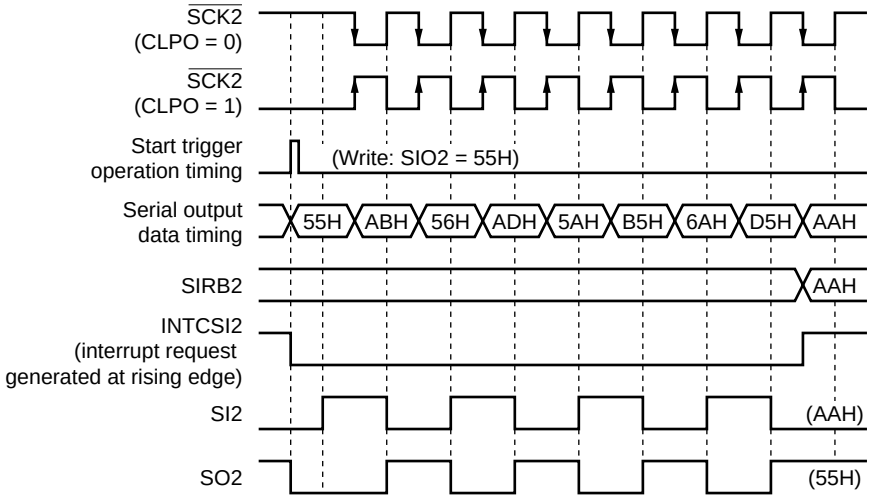
Figure 14-7 shows the operation timing when CLPH = 1. Two waves of $\overline{SCK2}$, when CLPO = 1 and when CLPO = 0, are shown in the figure.

Data is transmitted or received in 8-bit units. Each bit of data is transmitted or received in synchronization with the serial clock.

SIO2 is shifted at the falling edge of $\overline{SCK2}$ if CLPH = 1 and CLPO = 0. If CLPH = 1 and CLPO = 1, SIO2 is shifted at the rising edge of $\overline{SCK2}$. The transmit data is held by the SO2 latch and output from the SO2 pin. The receive data input to the SI2 pin is latched to SIO2 at the falling edge of $\overline{SCK2}$ (if CLPH = 1 and CLOP = 0).

Completion of an 8-bit transfer automatically stops operation of SIO2 and sets a serial transfer completion flag.

Figure 14-7. Operation Timing When CLPH Is Set to 1 (Serial output data: 55H, serial input data: AAH)



(6) Hardware detection in error status

Serial interface SIO2 has a function for checking whether an overflow error has occurred.

While serial data being transferred to the serial receive data buffer register (SIRB2) has not yet been read, the overflow flag (SDOF) is set to 1 if the capture strobe of the LSB of the serial data to be transferred next has occurred.

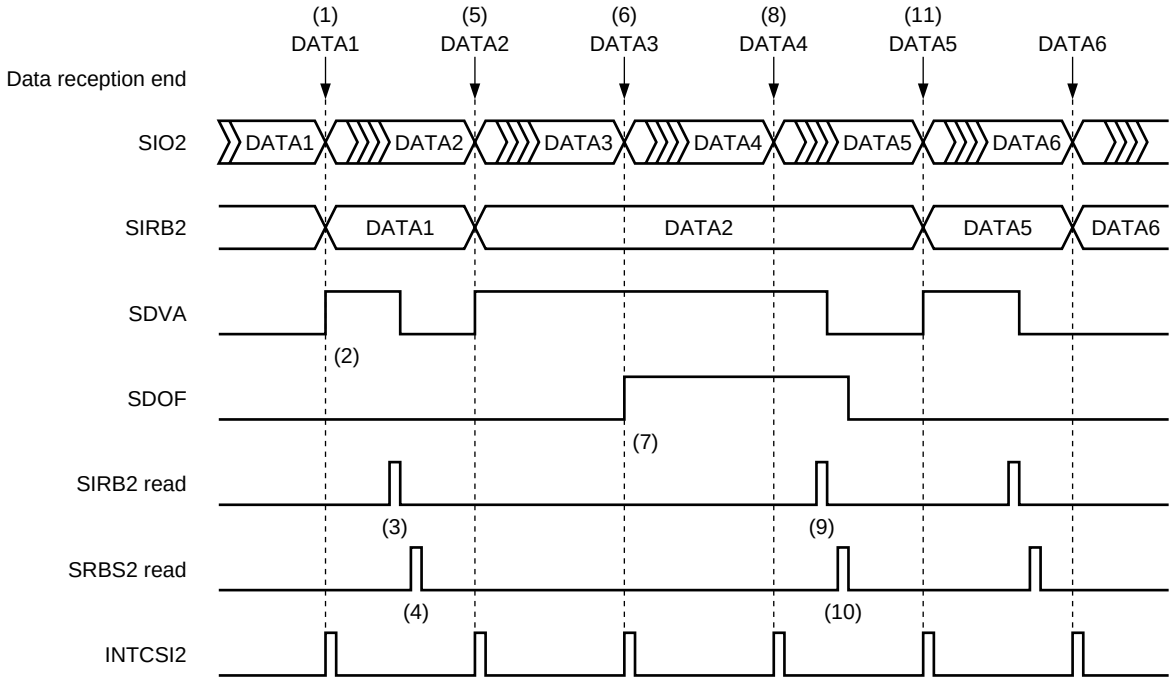
If an overflow has occurred, the received serial data is not transferred to SIRB2. Therefore, data that has already been received and transferred to SIRB2 can be read. In this way, the loss of receive data is prevented even if an overflow error occurs.

SDOF is cleared by reading the serial receive data buffer status register (SRBS2).

If SIRB2 is read to monitor the status of serial interface SIO2, always monitor SRBS2 to check whether an interrupt request (overflow error) has occurred.

Figure 14-8 shows the status of each register (SIO2, SIRB2, SRBS2) during a receive operation.

Figure 14-8. Receive Operation



(1) After DATA1 (receive data) is received completely, it is transferred to SIRB2.
(2) SDVA is set since the receive data is transferred to SIRB2.
(3) SDVA is automatically cleared to 0 by SIRB2 read.
(4) SRBS2 is read and the status is checked (overflow error check).
(5) Like (1) above, DATA2 (receive data) is transferred to SIRB2 after it is received completely.
(6) Even though DATA3 (receive data) reception is complete, it is held in SIO2 without being transferred to SIRB2 since the previous receive data (DATA2) has not been read from SIRB2.
(7) SDOF is set to 1 since SDVA has been set to 1 and DATA3 reception for SIO2 is complete.
(8) DATA3 (receive data) is discarded and DATA4 (receive data) reception ends.
Like (6) above, DATA4 is not transferred to SIRB2 and is held in SIO2.
(9) Like (3) above, SDVA is automatically cleared to 0 by SIRB2 read.
(10) SDOF is automatically cleared to 0 by SRBS2 read.
(11) Like (1) above, after DATA5 (receive data) is received completely, DATA5 is transferred to SIRB2.

(7) Operation in standby mode

(a) Operation in HALT mode

Even after a HALT instruction has been executed, serial interface SIO2 continues to operate.

In the HALT mode, the CPU cannot access the registers of serial interface SIO2.

If it is not necessary to use serial interface SIO2 in the HALT mode, the power consumption can be reduced by stopping the operation of the serial interface SIO2 before the HALT instruction is executed.

(b) Operation in STOP mode

Serial interface SIO2 can operate in the STOP mode if the slave mode (in which an external clock is used) is selected.

CHAPTER 15 SERIAL INTERFACE SIO3

15.1 Serial Interface Functions

The serial interface SIO3 has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. For details, see **15.4.1 Operation stop mode**.

(2) 3-wire serial I/O mode (fixed as MSB first)

This is an 8-bit data transfer mode using three lines: a serial clock line ($\overline{\text{SCK3}}$), serial output line (SO3), and serial input line (SI3).

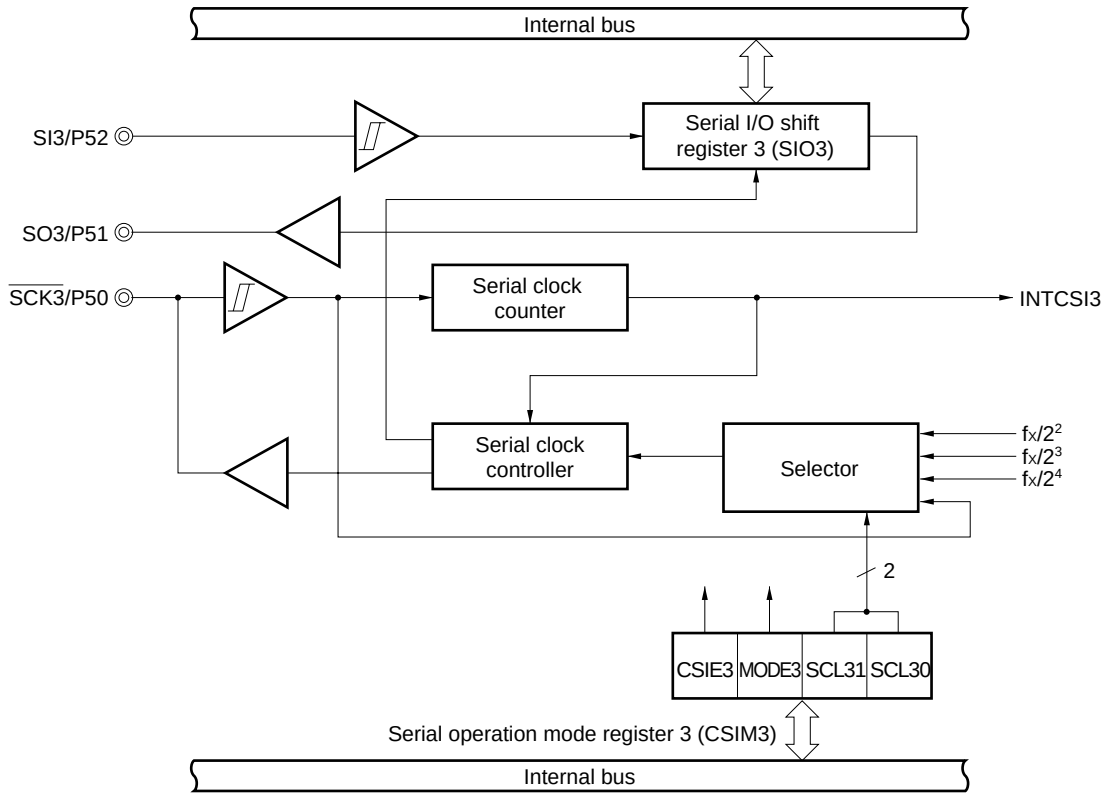
Since simultaneous transmit and receive operations are enabled in 3-wire serial I/O mode, the processing time for data transfers is reduced.

The first bit in the 8-bit data in serial transfers is fixed as the MSB.

3-wire serial I/O mode is useful for connection to a peripheral I/O device that includes a clocked serial interface, a display controller, etc. For details, see **15.4.2 3-wire serial I/O mode**.

Figure 15-1 shows the serial interface SIO3 block diagram.

Figure 15-1. Serial Interface SIO3 Block Diagram



15.2 Serial Interface Configuration

The serial interface SIO3 consists of the following hardware.

Table 15-1. Serial Interface SIO3 Configuration

Item	Configuration
Register	Serial I/O shift register 3 (SIO3)
Control register	Serial operation mode register 3 (CSIM3)

(1) Serial I/O shift register 3 (SIO3)

This is an 8-bit register that performs parallel-serial conversion and serial transmit/receive (shift operations) synchronized with the serial clock.

SIO3 is set with an 8-bit memory manipulation instruction.

When “1” is set to bit 7 (CSIE3) of the serial operation mode register 3 (CSIM3), a serial operation can be started by writing data to or reading data from SIO3.

When transmitting, data written to SIO3 is output via the serial output (SO3).

When receiving, data is read from the serial input (SI3) and written to SIO3.

$\overline{\text{RESET}}$ input clears SIO3 to 00H.

Caution Do not access SIO3 during a transfer operation unless the access is triggered by a transfer start (Read is disabled when MODE3 = 0 and write is disabled when MODE3 = 1).

15.3 Serial Interface Control Register

The serial operation mode register 3 (CSIM3) is used to control the serial interface SIO3.
This register is used to set the SIO3's serial clock, operation mode, and operation enable/disable.
CSIM3 is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears CSIM3 to 00H.

Caution In the 3-wire serial I/O mode, set the port mode register (PM5X) as follows. Besides that, set all output latches to 0.

- **When serial clock output (Master transmit or master receive)**
  Set P50 ($\overline{\text{SCK3}}$) to the output mode (PM50 = 0)
- **When serial clock input (Slave transmit or slave receive)**
  Set P50 to the input mode (PM50 = 1)
- **When transmit or transmit/receive mode**
  Set P51 (SO3) to the output mode (PM51 = 0)
- **When receive mode**
  Set P52 (SI3) to the input mode (PM52 = 1)

Figure 15-2. Serial Operation Mode Register 3 (CSIM3) Format

Address: FF84H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM3	CSIE3	0	0	0	0	MODE3	SCL31	SCL30

CSIE3	SIO3 Operation Enable/Disable Specification Shift Register Operation
0	Operation disabled
1	Operation enabled

MODE3	Transfer Operation Mode Flag Operation Mode
0	Transmit or transmit/receive mode
1	Receive-only mode

SCL31	SCL30	Clock Selection
0	0	External clock input
0	1	$f_X/2^2$
1	0	$f_X/2^3$
1	1	$f_X/2^4$

Caution Bits 3 to 6 must be set to 0.

Remark f_X: Main system clock oscillation frequency

15.4 Serial Interface Operations

This section explains the two modes of the serial interface SIO3.

15.4.1 Operation stop mode

This mode is used when serial transfers are not performed to reduce power consumption.

In the operation stop mode, the P50/$\overline{\text{SCK3}}$, P51/SO3, and P52/SI3 pins can be used as normal I/O port pins.

(1) Register settings

Operation stop mode is set with the serial operation mode register 3 (CSIM3).

CSIM3 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM3 to 00H.

Address: FF84H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM3	CSIE3	0	0	0	0	MODE3	SCL31	SCL30

CSIE3	SIO3 Operation Enable/Disable Specification Shift Register Operation
0	Operation disabled
1	Operation enabled

Caution Bits 3 to 6 must be set to 0.

15.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is useful when connecting a peripheral I/O device that includes a clocked serial interface, a display controller, etc.

This mode executes data transfers via three lines: a serial clock line ($\overline{SCK3}$), serial output line (SO3), and serial input line (SI3).

(1) Register settings

3-wire serial I/O mode is set with the serial operation mode register 3 (CSIM3).

CSIM3 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input clears CSIM3 to 00H.

Caution In the 3-wire serial I/O mode, set the port mode register (PM5X) as follows. Besides that, set all output latches to 0.

- **When serial clock output (Master transmit or master receive)**
  Set P50 ($\overline{SCK3}$) to the output mode (PM50 = 0)
- **When serial clock input (Slave transmit or slave receive)**
  Set P50 to the input mode (PM50 = 1)
- **When transmit or transmit/receive mode**
  Set P51 (SO3) to the output mode (PM51 = 0)
- **When receive mode**
  Set P52 (SI3) to the input mode (PM52 = 1)

Address: FF84H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIM3	CSIE3	0	0	0	0	MODE3	SCL31	SCL30

CSIE3	SIO3 Operation Enable/Disable Specification Shift Register Operation
0	Operation disabled
1	Operation enabled

MODE3	Transfer Operation Mode Flag Operation Mode
0	Transmit or transmit/receive mode
1	Receive-only mode

SCL31	SCL30	Clock Selection
0	0	External clock input
0	1	$f_X/2^2$
1	0	$f_X/2^3$
1	1	$f_X/2^4$

Caution Bits 3 to 6 must be set to 0.

Remark f_X: Main system clock oscillation frequency

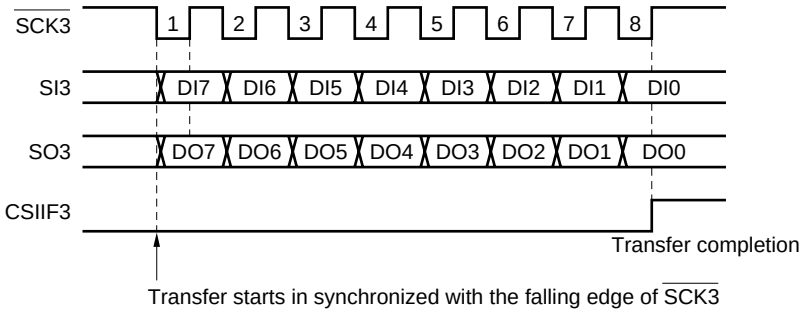
(2) Communication operations

In the 3-wire serial I/O mode, data is transmitted and received in 8-bit units. Each bit of data is transmitted or received in synchronization with the serial clock.

The serial I/O shift register 3 (SIO3) is shifted in synchronization with the falling edge of the serial clock. Transmission data is held in the SO3 latch and is output from the SO3 pin. Data that is received via the SI3 pin in synchronization with the rising edge of the serial clock is latched to SIO3.

Completion of an 8-bit transfer automatically stops operation of SIO3 and sets an interrupt request flag (CSIIF3).

Figure 15-3. 3-Wire Serial I/O Mode Timing



(3) Transfer start

A serial transfer starts when the following two conditions have been satisfied and transfer data has been set to (or read from) serial I/O shift register 3 (SIO3).

- SIO3 operation control bit (CSIE3) = 1
- After an 8-bit serial transfer, the internal serial clock is either stopped or $\overline{\text{SCK3}}$ is set to high level.
- Transmit or transmit/receive mode
  When CSIE3 = 1 and MODE3 = 0, transfer starts when writing to SIO3.
- Receive-only mode
  When CSIE3 = 1 and MODE3 = 1, transfer starts when reading from SIO3.

Caution After data has been written to SIO3, transfer will not start even if the CSIE3 bit value is set to "1".

Completion of an 8-bit transfer automatically stops the serial transfer operation and sets an interrupt request flag (CSIIF3).

CHAPTER 16 LCD CONTROLLER/DRIVER

16.1 LCD Controller/Driver Functions

The functions of the LCD controller/driver incorporated in the μPD780852 Subseries are shown below.

(1) Automatic output of segment signals and common signals is possible by automatic reading of the display data memory.
(2) Display mode
  • 1/4 duty (1/3 bias)
(3) Any of four frame frequencies can be selected in each display mode.
(4) Maximum of 20 segment signal outputs (S0 to S19); 4 common signal outputs (COM0 to COM3).
Fifteen of the segment signal outputs can be switched to input/output ports in units of 2 (P81/S19 to P87/S13, P90/S12 to P97/S5).

The maximum number of displayable pixels is shown in Table 16-1.

Table 16-1. Maximum Number of Display Pixels

Bias Method	Time Division	Common Signals Used	Maximum Number of Display Pixels
1/3	4	COM0 to COM3	80 (20 segments × 4 commons) **Note**

Note 10 digits on 8. type LCD panel with 2 segments/digit

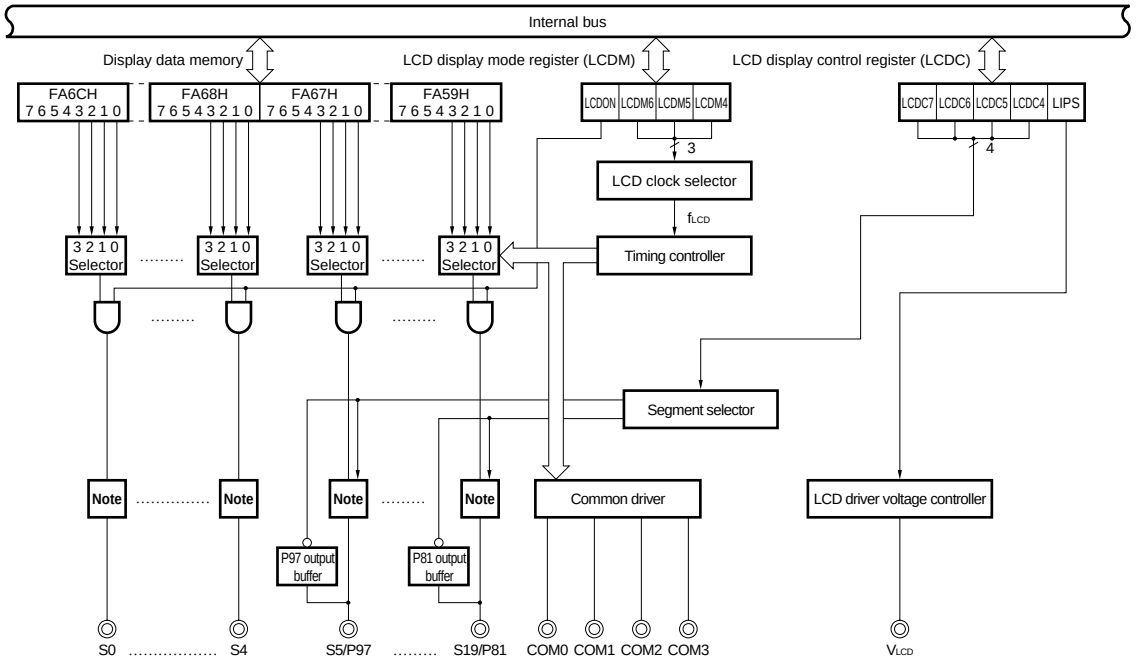
16.2 LCD Controller/Driver Configuration

The LCD controller/driver consists of the following hardware.

Table 16-2. LCD Controller/Driver Configuration

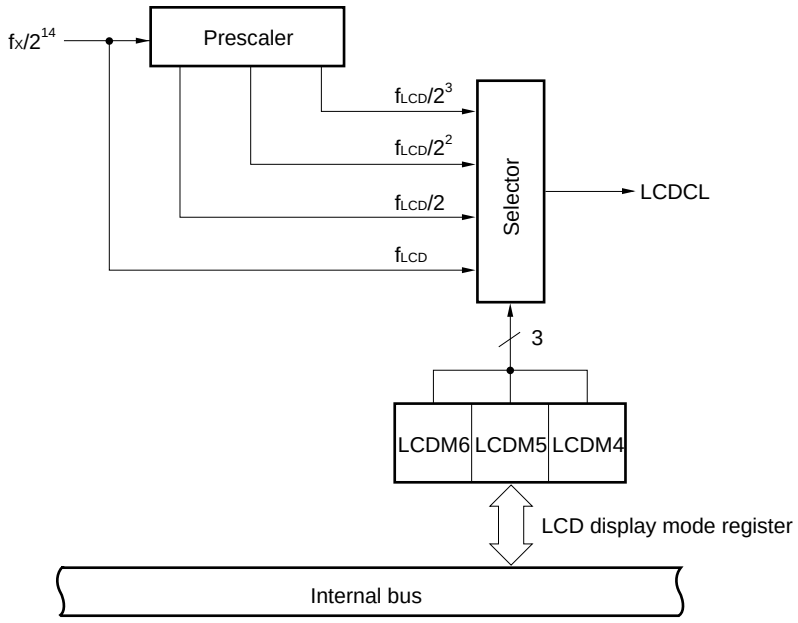
Item	Configuration
Display outputs	Segment signals: 20 Dedicated segment signals: 5 Segment signal/input or output port alternate function: 14 Segment signal/input or output port/16-bit timer prescaler output alternate function: 1 Common signals: 4 (COM0 to COM3)
Control registers	LCD display mode register (LCDM) LCD display control register (LCDC)

Figure 16-1. LCD Controller/Driver Block Diagram



Note Segment driver

Figure 16-2. LCD Clock Selector Block Diagram



Remarks 1. LCDCL: LCD clock

2. f_{LCD}: LCD clock frequency

16.3 LCD Controller/Driver Control Registers

The following two types of registers are used to control the LCD controller/driver.

- LCD display mode register (LCDM)
- LCD display control register (LCDC)

(1) LCD display mode register (LCDM)

This register sets display operation enabling/disabling, the LCD clock, and frame frequency.

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

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

Figure 16-3. LCD Display Mode Register (LCDM) Format

Address: FFB0H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDM	LCDON	LCDM6	LCDM5	LCDM4	0	0	0	0

LCDON	LCD Display Enable/Disable
0	Display off (all segment outputs are non-select signal outputs)
1	Display on

LCDM6	LCDM5	LCDM4	LCD Clock Selection (f_X = 8.38 MHz)
0	0	0	$f_X/2^{17}$ (64 Hz)
0	0	1	$f_X/2^{16}$ (128 Hz)
0	1	0	$f_X/2^{15}$ (256 Hz)
0	1	1	$f_X/2^{14}$ (512 Hz)
Other than above			Setting prohibited

Remark f_X: Main system clock oscillation frequency

(2) LCD display control register (LCDC)

This register sets cutoff of the current flowing to split resistors for LCD drive voltage generation and switchover between segment output and input/output port functions.

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

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

Figure 16-4. LCD Display Control Register (LCDC) Format

Address: FFB2H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDC	LCDC7	LCDC6	LCDC5	LCDC4	0	0	0	LIPS

LCDC7	LCDC6	LCDC5	LCDC4	P81/S19 to P97/S5 Pin Functions	
				Port Pins	Segment Pins
0	0	0	0	P81 to P97	None
0	0	0	1	P81 to P95	S5, S6
0	0	1	0	P81 to P93	S5 to S8
0	0	1	1	P81 to P91	S5 to S10
0	1	0	0	P81 to P87	S5 to S12
0	1	0	1	P81 to P85	S5 to S14
0	1	1	0	P81 to P83	S5 to S16
0	1	1	1	P81	S5 to S18
1	0	0	0	None	S5 to S19
Other than above				Setting prohibited	

LIPS	LCD Driving Power Supply Selection
0	Does not supply power to LCD.
1	Supplies power to LCD from V_{DD} pin.

Cautions 1. Pins which perform segment output cannot be used as output port pins even if 0 is set in the port mode register.

2. If a pin which performs segment output is read as a port, its value will be 0.

★ **3. If a pin set as a segment output pin is not used, leave that pin open.**

16.4 LCD Controller/Driver Settings

LCD controller/driver settings should be performed as shown below.

<1> Set the initial value in the display data memory (FA59H to FA6CH).
<2> Set the pins to be used as segment outputs in the LCD display control register (LCDC).
<3> Set the LCD clock in the LCD display mode register (LCDM).

Next, set data in the display data memory according to the display contents.

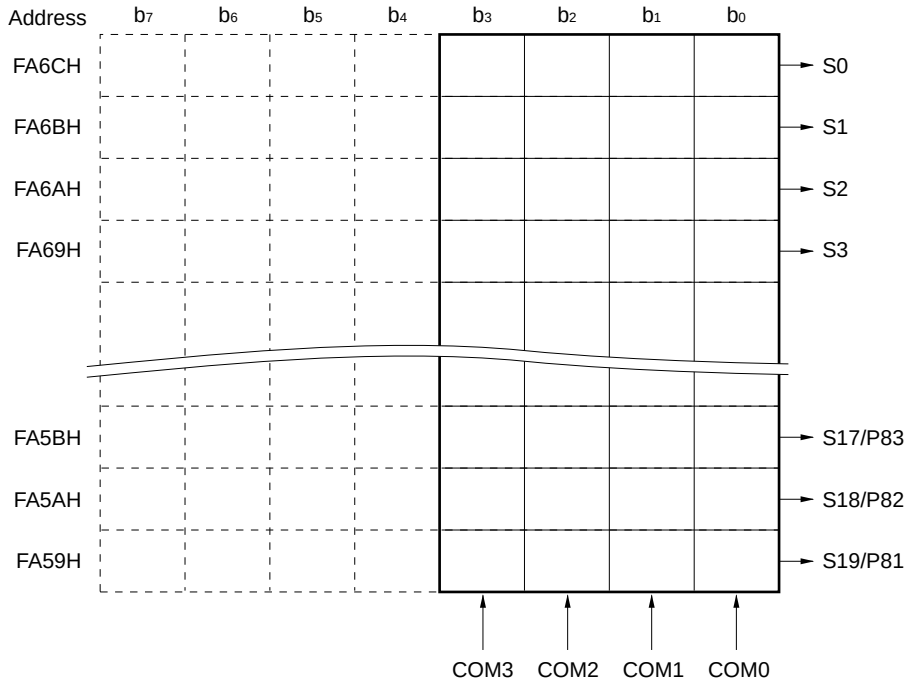
16.5 LCD Display Data Memory

The LCD display data memory is mapped onto addresses FA59H to FA6CH. The data stored in the LCD display data memory can be displayed on an LCD panel by the LCD controller/driver.

Figure 16-5 shows the relation between the LCD display data memory contents and the segment outputs/common outputs.

Any area not used for display can be used as normal RAM.

Figure 16-5. Relation between LCD Display Data Memory Contents and Segment/Common Outputs



Caution The higher 4 bits of the LCD display data memory do not incorporate memory. Be sure to set them to 0.

16.6 Common Signals and Segment Signals

An individual pixel on an LCD panel lights when the potential difference of the corresponding common signal and segment signal reaches or exceeds a given voltage (the LCD drive voltage V_{LCD}). The light goes off when the potential difference becomes V_{LCD} or lower.

As an LCD panel deteriorates if a DC voltage is applied in the common signals and segment signals, it is driven by AC voltage.

(1) Common signals

For common signals, the selection timing order is as shown in Table 16-3, and operations are repeated with these as the cycle.

Table 16-3. COM Signals

COM signal / Time division	COM0	COM1	COM2	COM3
4-time division				

(2) Segment signals

Segment signals correspond to a 20-byte LCD display data memory (FA59H to FA6CH). Each display data memory bit 0, bit 1, bit 2, and bit 3 is read in synchronization with the COM0, COM1, COM2 and COM3 timings respectively, and if the value of the bit is 1, it is converted to the selection voltage. If the value of the bit is 0, it is converted to the non-selection voltage and output to a segment pin (S0 to S19) (S18 to S5 have an alternate function as input/output port pins).

Consequently, it is necessary to check what combination of front surface electrodes (corresponding to the segment signals) and rear surface electrodes (corresponding to the common signals) of the LCD panel to be used form the display pattern, and then write bit data corresponding on a one-to-one basis with the pattern to be displayed.

Bits 4 to 7 are fixed at 0.

(3) Common signal and segment signal output waveforms

The voltages shown in Table 16-4 are output in the common signals and segment signals.

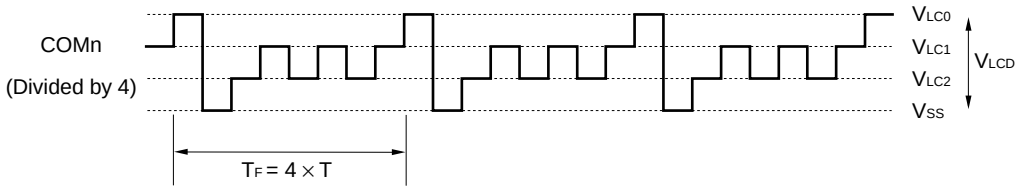
The $\pm V_{LCD}$ ON voltage is only produced when the common signal and segment signal are both at the selection voltage; other combinations produce the OFF voltage.

Table 16-4. LCD Drive Voltage

Segment Signal / Common Signal		Selection Signal Level	Non-Selection Signal Level
		V_{SS1}, V_{LC0}	V_{LC1}, V_{LC2}
Selection signal level	V_{LC0}, V_{SS1}	$-V_{LCD}$, $+V_{LCD}$	$-1/3V_{LCD}$, $+1/3V_{LCD}$
Non-selection signal level	V_{LC2}, V_{LC1}	$-1/3V_{LCD}$, $+1/3V_{LCD}$	$-1/3V_{LCD}$, $+1/3V_{LCD}$

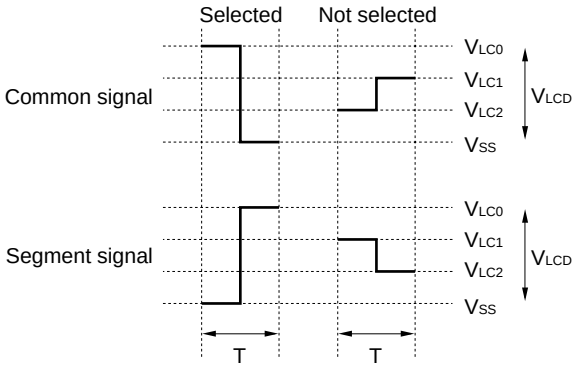
Figure 16-6 shows the common signal waveform, and Figure 16-7 shows the common signal and segment signal voltages and phases.

Figure 16-6. Common Signal Waveform



T: One LCDCL cycle
TF: Frame frequency

Figure 16-7. Common Signal and Segment Signal Voltages and Phases



T: One LCDCL cycle

16.7 Supplying LCD Drive Voltage V_{LC0}, V_{LC1}, and V_{LC2}

The μPD780852 Subseries have a split resistor to create an LCD drive voltage, and the drive voltage is fixed to 1/3 bias.

To supply various LCD drive voltages, internal V_{DD} or external V_{LCD} supply voltage can be selected.

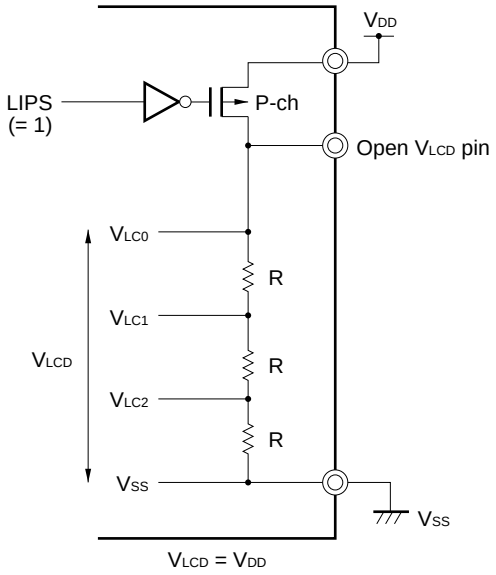
Table 16-5. LCD Drive Voltage

Bias Method / LCD Drive Voltage	1/3 Bias Method
V_{LC0}	V_{LCD}
V_{LC1}	$2/3V_{LCD}$
V_{LC2}	$1/3V_{LCD}$

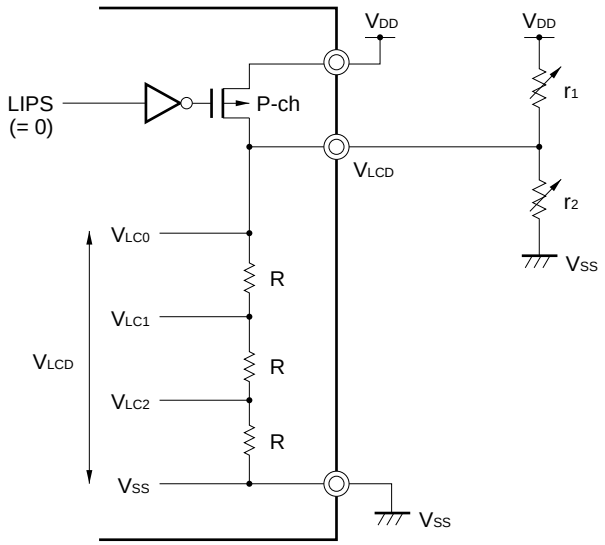
Figure 16-8 shows an example of supplying an LCD drive voltage from an internal source according to Table 16-5. By using variable resistors r_1 and r_2, a non-stepwise LCD drive voltage can be supplied.

Figure 16-8. Example of Connection of LCD Drive Power Supply

(a) To supply LCD drive voltage from V_{DD}



(b) To supply LCD drive voltage from external source



$$V_{LCD} = \frac{3R \bullet r_2}{3R \bullet r_2 + 3R \bullet r_1 + r_1 \bullet r_2} \times V_{DD}$$

16.8 Display Mode

16.8.1 4-time-division display example

Figure 16-10 shows the connection of a 4-time-division type 10-digit LCD panel with the display pattern shown in Figure 16-9 with the μPD780852 Subseries segment signals (S0 to S19) and common signals (COM0 to COM3). The display example is "1234567890," and the display data memory contents (addresses FA59H to FA6CH) correspond to this.

An explanation is given here taking the example of the 5th digit "6" ( ). In accordance with the display pattern in Figure 16-9, selection and non-selection voltages must be output to pins S8 and S9 as shown in Table 16-6 at the COM0 to COM3 common signal timings.

Table 16-6. Selection and Non-Selection Voltages (COM0 to COM3)

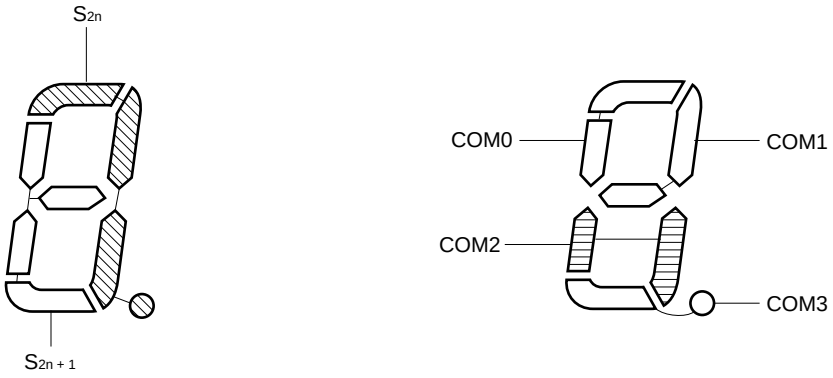
Segment / Common	S8	S9
COM0	S	S
COM1	NS	S
COM2	S	S
COM3	NS	S

S: Selection, NS: Non-selection

From this, it can be seen that 0101 must be prepared in the display data memory (address FA64H) corresponding to S8.

Examples of the LCD drive waveforms between S8 and the COM0 and COM1 signals are shown in Figure 16-11 (for the sake of simplicity, waveforms for COM2 and COM3 have been omitted). When S8 is at the selection voltage at the COM0 selection timing, it can be seen that the $+V_{LCD}/-V_{LCD}$ AC square wave, which is the LCD illumination (ON) level, is generated.

Figure 16-9. 4-Time-Division LCD Display Pattern and Electrode Connections



n = 0 to 9

Figure 16-10. 4-Time-Division LCD Panel Connection Example

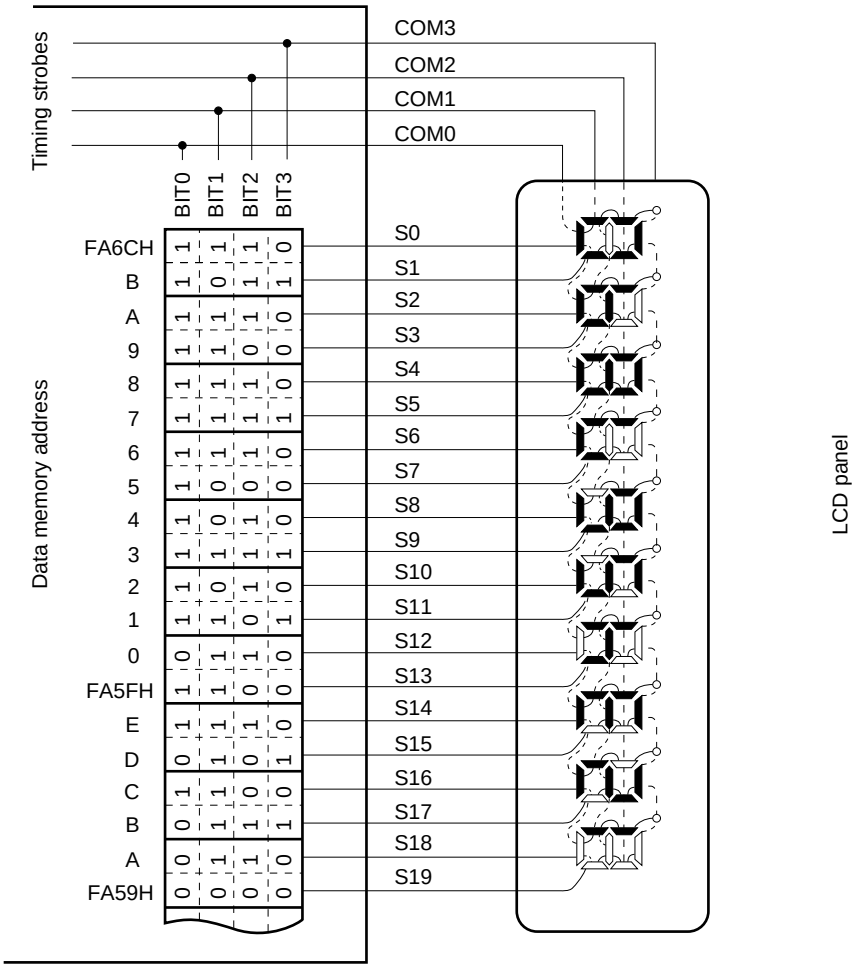
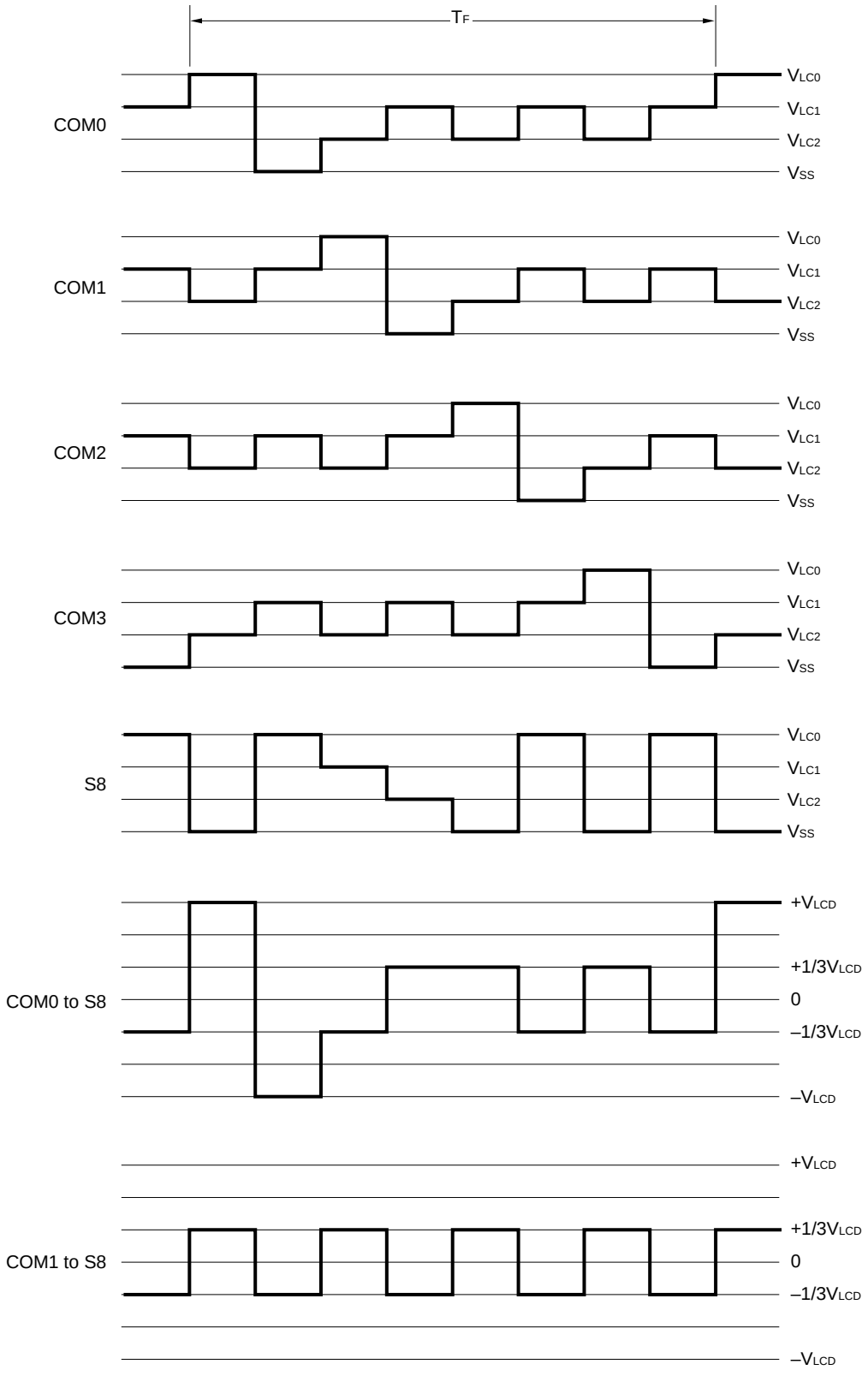


Figure 16-11. 4-Time-Division LCD Drive Waveform Examples (1/3 Bias Method)



16.9 Cautions on Emulation

(1) LCD timer control register (LCDTM)

To perform debugging with an in-circuit emulator (IE-78K0-NS), the LCD timer control register (LCDTM) must be set. LCDTM is a register used to set a probe board (IE-780852-NS-EM4).

LCDTM is a write-only register that controls supply of the LCD clock. Unless LCDTM is set, the LCD controller/driver does not operate. Therefore, set bit 1 (TMC21) of LCDTM to 1 when using the LCD controller/driver.

Figure 16-12. LCD Timer Control Register (LCDTM) Format

Address: FF4AH After Reset: 00H W

Symbol	7	6	5	4	3	2	1	0
LCDTM	0	0	0	0	0	0	TMC21	0

TMC21	LCD Clock Supply Control
0	LCD controller/driver stop mode (supply of LCD clock is stopped)
1	LCD controller/driver operation mode (supply of LCD clock is enabled)

Cautions 1. LCDTM is a special register that must be set when debugging is performed with an in-circuit emulator. Even if this register is used, the operation of the μPD780852 Subseries is not affected. However, delete the instruction that manipulates this register from the program at the final stage of debugging.

2. Bits 7 to 2 and 0 must be set to 0.

[MEMO]

CHAPTER 17 SOUND GENERATOR

17.1 Sound Generator Function

The sound generator has the function to sound the buzzer from an external speaker, and the following signal are output.

- **Basic cycle output signal**
  The signal is a buzzer signal with a variable frequency. By setting bits 0 to 2 (SGCL0 to SGCL2) of the sound generator control register (SGCR), the signal in a range of 0.12 to 4.0 kHz can be output (when f_X = 8.38 MHz). The amplitude of the 7-bit-resolution PWM signal can be varied to enable control of the buzzer sound volume.

  Figure 17-1 shows the sound generator block diagram and Figure 17-2 shows the concept of basic cycle output signal SGO.

Figure 17-1. Sound Generator Block Diagram

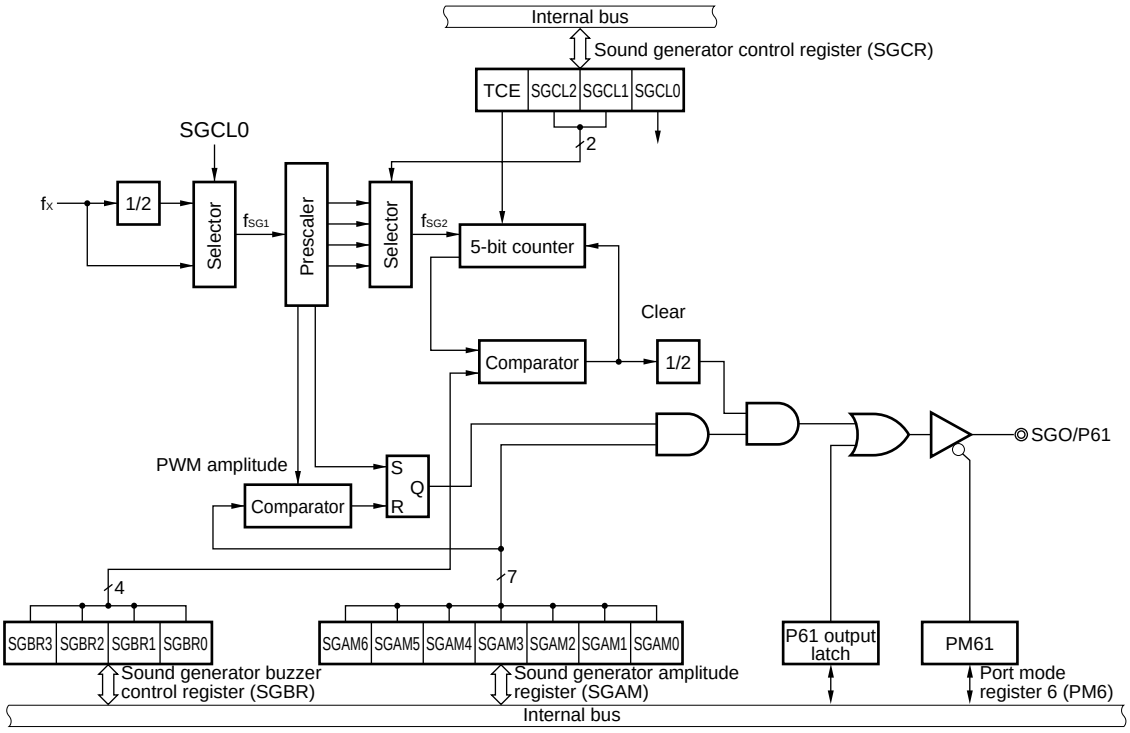
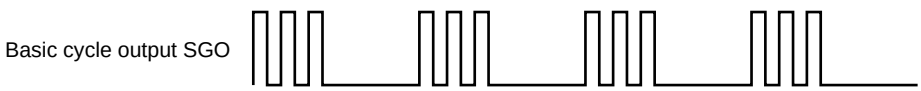


Figure 17-2. Concept of Basic Cycle Output Signal SGO



17.2 Sound Generator Configuration

The sound generator consists of the following hardware.

Table 17-1. Sound Generator Configuration

Item	Configuration
Counter	8 bits × 1, 5 bits × 1
SG output	SGO
Control register	Sound generator control register (SGCR) Sound generator buzzer control register (SGBR) Sound generator amplitude register (SGAM)

17.3 Sound Generator Control Registers

The following three types of registers are used to control the sound generator.

- Sound generator control register (SGCR)
- Sound generator buzzer control register (SGBR)
- Sound generator amplitude register (SGAM)

(1) Sound generator control register (SGCR)

SGCR is a register which sets up the following three types.

- Controls sound generator output
- Selects sound generator input frequency f_{SG1}
- Selects 5-bit counter input frequency f_{SG2}

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

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

Figure 17-3 shows the SGCR format.

Figure 17-3. Sound Generator Control Register (SGCR) Format

Address: FF94H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SGCR	TCE	0	0	0	1	SGCL2	SGCL1	SGCL0

TCE	Sound Generator Operation Selection
0	Timer operation stopped SGO for low-level output
1	Sound generator operation SGO for output

SGCL2	SGCL1	5-Bit Counter Input Frequency f_{SG2} Selection
0	0	$f_{SG2} = f_{SG1}/2^5$
0	1	$f_{SG2} = f_{SG1}/2^6$
1	0	$f_{SG2} = f_{SG1}/2^7$
1	1	$f_{SG2} = f_{SG1}/2^8$

SGCL0	Sound Generator Input Frequency f_{SG1} Selection
0	$f_{SG1} = f_X/2$
1	$f_{SG1} = f_X$

Cautions 1. Before setting the TCE bit, set all the other bits.

2. When rewriting SGCR to other data, stop the timer operation (TCE = 0) beforehand.

3. Bits 4 to 6 must be set to 0.

4. Bit 3 must be set to 1.

The maximum and minimum values of the buzzer output frequency are as follows.

Table 17-2. Maximum Value and Minimum Value of Buzzer Output Frequency

SGCL2	SGCL1	SGCL0	Maximum and Minimum Values of Buzzer Output				
			f_{SG2}	f_X = 8.00 MHz		f_X = 8.38 MHz	
				Max. (kHz)	Min. (kHz)	Max. (kHz)	Min. (kHz)
0	0	0	$f_X/2^6$	3.68	1.95	3.85	2.05
0	0	1	$f_X/2^5$	3.68	1.95	3.85	2.05
0	1	0	$f_X/2^7$	1.84	0.98	1.93	1.02
0	1	1	$f_X/2^6$	1.84	0.98	1.93	1.02
1	0	0	$f_X/2^8$	0.92	0.49	0.96	0.51
1	0	1	$f_X/2^7$	0.92	0.49	0.96	0.51
1	1	0	$f_X/2^9$	0.46	0.24	0.48	0.26
1	1	1	$f_X/2^8$	0.46	0.24	0.48	0.26

(2) Sound generator buzzer control register (SGBR)

SGBR is a register that sets the basic frequency of the sound generator output signal.

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

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

Figure 17-4 shows the SGBR format.

Figure 17-4. Sound Generator Buzzer Control Register (SGBR) Format

Address: FF95H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SGBR	0	0	0	0	SGBR3	SGBR2	SGBR1	SGBR0

SGBR3	SGBR2	SGBR1	SGBR0	Buzzer Output Frequency (kHz) Note	
				f_X = 8 MHz	f_X = 8.38 MHz
0	0	0	0	3.677	3.851
0	0	0	1	3.472	3.637
0	0	1	0	3.290	3.446
0	0	1	1	3.125	3.273
0	1	0	0	2.976	3.117
0	1	0	1	2.841	2.976
0	1	1	0	2.717	2.847
0	1	1	1	2.604	2.728
1	0	0	0	2.500	2.619
1	0	0	1	2.404	2.518
1	0	1	0	2.315	2.425
1	0	1	1	2.232	2.339
1	1	0	0	2.155	2.258
1	1	0	1	2.083	2.182
1	1	1	0	2.016	2.112
1	1	1	1	1.953	2.046

Note Output frequency where SGCL0, SGCL1, and SGCL2 are all 0s

Cautions 1. When rewriting SGBR to other data, stop the timer operation (TCE = 0) beforehand.

2. Bits 4 to 7 must be set to 0.

(3) Sound generator amplitude register (SGAM)

SGAM is a register that sets the amplitude of the sound generator output signal.

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

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

Figure 17-5 shows the SGAM format.

Figure 17-5. Sound Generator Amplitude Register (SGAM) Format

Address: FFC1H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SGAM	0	SGAM6	SGAM5	SGAM4	SGAM3	SGAM2	SGAM1	SGAM0

SGAM6	SGAM5	SGAM4	SGAM3	SGAM2	SGAM1	SGAM0	Amplitude
0	0	0	0	0	0	0	0/128
0	0	0	0	0	0	1	2/128
0	0	0	0	0	1	0	3/128
0	0	0	0	0	1	1	4/128
0	0	0	0	1	0	0	5/128
0	0	0	0	1	0	1	6/128
0	0	0	0	1	1	0	7/128
0	0	0	0	1	1	1	8/128
0	0	0	1	0	0	0	9/128
0	0	0	1	0	0	1	10/128
0	0	0	1	0	1	0	11/128
0	0	0	1	0	1	1	12/128
0	0	0	1	1	0	0	13/128
0	0	0	1	1	0	1	14/128
0	0	0	1	1	1	0	15/128
0	0	0	1	1	1	1	16/128
0	0	1	0	0	0	0	17/128
0	0	1	0	0	0	1	18/128
0	0	1	0	0	1	0	19/128
0	0	1	0	0	1	1	20/128
0	0	1	0	1	0	0	21/128
0	0	1	0	1	0	1	22/128
0	0	1	0	1	1	0	23/128
0	0	1	0	1	1	1	24/128
0	0	1	1	0	0	0	25/128
0	0	1	1	0	0	1	26/128
0	0	1	1	0	1	0	27/128
0	0	1	1	0	1	1	28/128
0	0	1	1	1	0	0	29/128
0	0	1	1	1	0	1	30/128
0	0	1	1	1	1	0	31/128
			⋮				⋮
1	1	1	1	1	1	1	128/128

Cautions 1. When rewriting SGAM to other data, stop the timer operation beforehand. However, note that a high level may be output for one period due to rewrite timing.

2. Bit 7 must be set to 0.

17.4 Sound Generator Operations

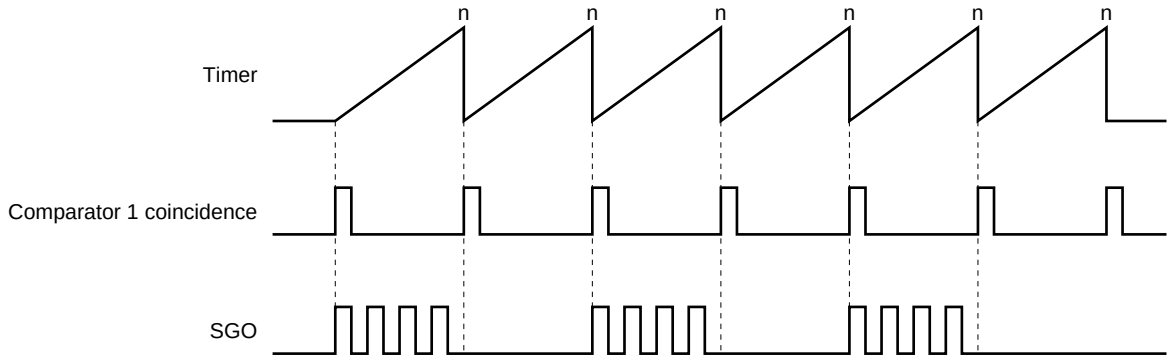
17.4.1 To output basic cycle signal SGO

The basic cycle signal is output from the SGO pin if the bit 7 (TCE) of the sound generator control register (SGCR) is set to “1”.

The basic cycle signal of the frequency set by SGCL0 to SGCL2 and SGBR0 to SGBR3 is output.

The amplitude of the basic cycle signal can be changed by changing the set value of the sound generator amplitude register (SGAM).

Figure 17-6. Sound Generator Output Operation Timing



[MEMO]

CHAPTER 18 METER CONTROLLER/DRIVER

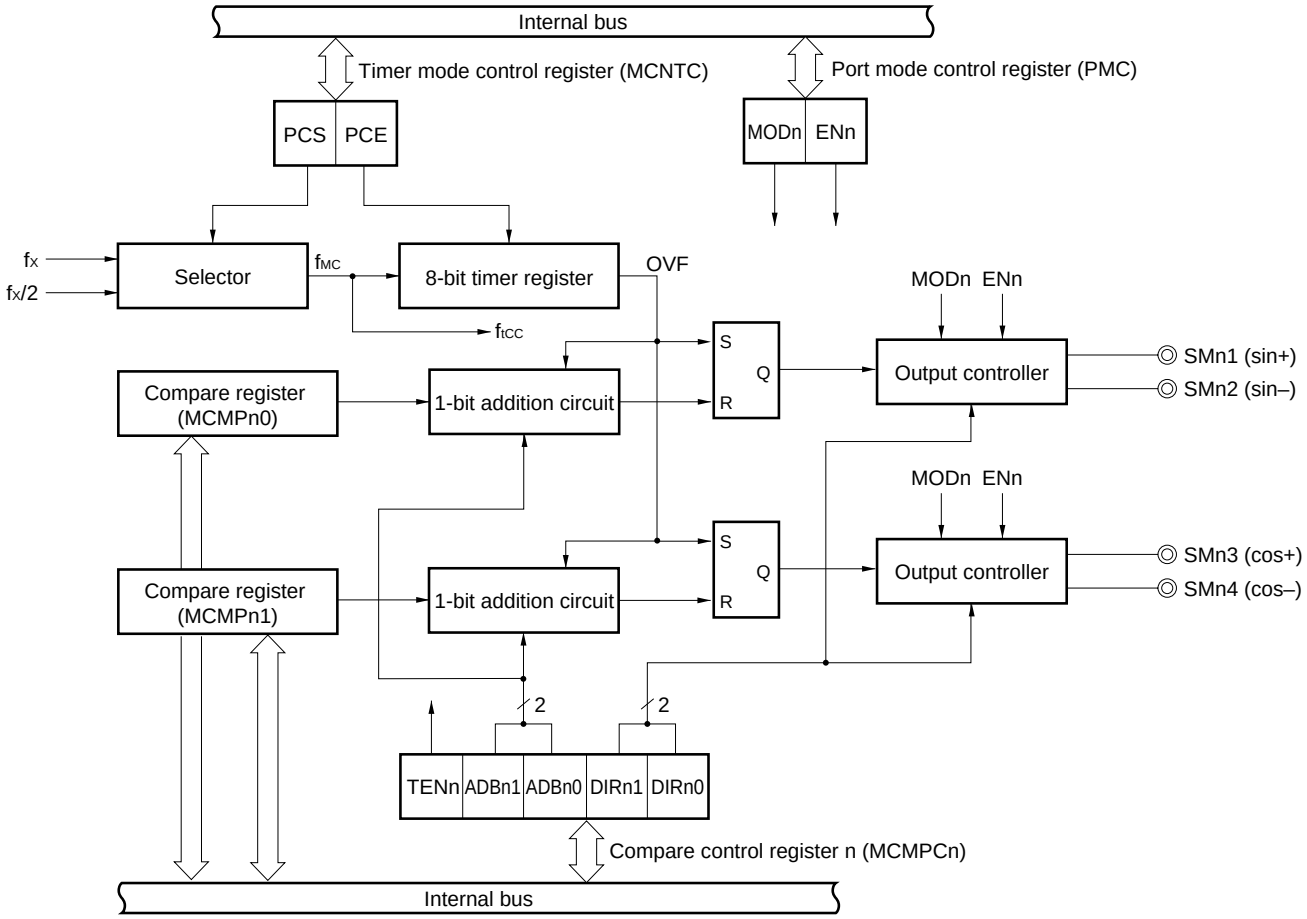
18.1 Meter Controller/Driver Functions

The meter controller/driver is a function to drive a stepping motor for external meter control or cross coil.

- Can set pulse width with a precision of 8 bits
- Can set pulse width with a precision of 8 + 1 bits with 1-bit addition function
- Can drive up to four 360˚ type meters

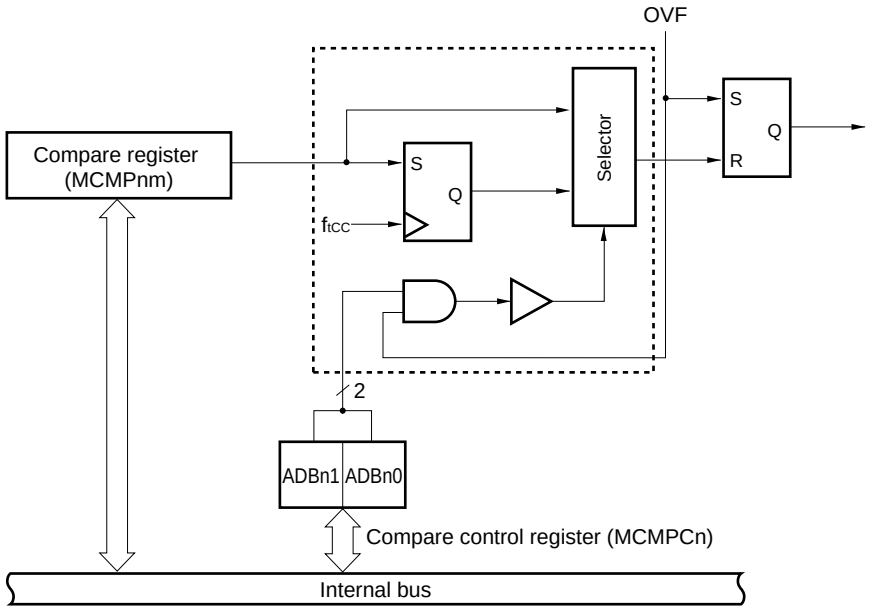
Figure 18-1 shows the block diagram of the meter controller/driver and Figure 18-2 shows 1-bit addition circuit block diagram.

Figure 18-1. Meter Controller/Driver Block Diagram



Remark n = 1 to 4

Figure 18-2. 1-Bit Addition Circuit Block Diagram



Remark n = 1 to 4, m = 0, 1

18.2 Meter Controller/Driver Configuration

The meter controller/driver consists of the following hardware.

Table 18-1. Meter Controller/Driver Configuration

Item	Configuration
Timer	Free-running up counter (MCNT): 1 channel
Register	Compare register (MCMPn1, MCMPn0): 8 channels
Control registers	Timer mode control register (MCNTC) Compare control register n (MCMPCn) Port mode control register (PMC)
Pulse controller	1-bit addition circuit/output controller

Remark n = 1 to 4

(1) Free-running up counter (MCNT)

MCNT is an 8-bit free-running up counter, and is a register that executes increment at the rising edge of input clock.

A PWM pulse with a resolution of 8 bits can be output. The duty factor can be set in a range of 0% to 100%. The count value is cleared in the following cases.

- $\overline{\text{RESET}}$ input
- Stop counter (PCE = 0)

Cautions 1. MCNT executes counting operation from 01H to FFH repeatedly. However, it counts from 00H upon operation start.

2. The PWM output is not output until the first overflow of MCNT.

(2) Compare register n0 (MCMPn0)

MCMPn0 is an 8-bit register that can rewrite compare values through specification of bit 4 (TENn) of the compare control register n (MCMPCn).

$\overline{\text{RESET}}$ input clears this register to 00H and clears hardware to 0.

MCMPn0 is a register that supports read/write only for 8-bit access instructions. MCMPn0 continuously compares its value with the MCNT value. When the above two values match, a match signal of the sin side of meter n is generated.

(3) Compare register n1 (MCMPn1)

MCMPn1 is an 8-bit register that can rewrite compare values through specification of bit 4 (TENn) of the compare control register n (MCMPCn).

$\overline{\text{RESET}}$ input clears this register to 00H and clears hardware to 0.

MCMPn1 is a register that supports read/write only for 8-bit access instructions. MCMPn1 continuously compares its value with the MCNT value. When the above two values match, a match signal of the cos side of meter n is generated.

(4) 1-bit addition circuit

The 1-bit addition circuit repeats 1-bit addition/non-addition to PWM output alternately upon MCNT overflow output, and enables the state of PWM output between current compare value and the next compare value. This circuit is controlled by bits 2 and 3 (ADBn0 and ADBn1) of the MCMPCn register.

(5) Output controller

The output controller consists of a P-ch and N-ch drivers and can drive a meter in H bridge configuration by connecting a coil.

When a meter is driven in half bridge configuration, the unused pins can be used as normal output port pins.

The relation of the duty factor of the PWM signal output from the SMnm pin is indicated by the following expression (n = 1 to 4, m = 0, 1).

$$\text{PWM (duty)} = \frac{\text{Set value of MCMPnm} \times \text{cycle of MCNT count clock}}{255 \times \text{cycle of MCNT count clock}} \times 100\%$$

$$= \frac{\text{Set value of MCMPnm}}{255} \times 100\%$$

Cautions 1. MCMPn0 and MCMPn1 cannot be read or written by a 16-bit access instruction.

2. MCMPn0 and MCMPn1 are in master-slave configuration, and MCNT is compared with a slave register. The PWM pulse is not output until the first overflow occurs after the counting operation has been started because the compare data is not transferred to the slave.

18.3 Meter Controller/Driver Control Registers

The following three types of registers are used to control the meter controller/driver.

- Timer mode control register (MCNTC)
- Compare control register n (MCMPCn)
- Port mode control register (PMC)

Remark n = 1 to 4

(1) Timer mode control register (MCNTC)

MCNTC is an 8-bit register that controls the operation of the free-running up counter (MCNT).

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

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

Figure 18-3 shows the MCNTC format.

Figure 18-3. Timer Mode Control Register (MCNTC) Format

Address: FF69H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MCNTC	0	0	PCS	PCE	0	0	0	0

PCS	Timer Counter Clock (f_{MC}) Selection
0	f_X
1	$f_X/2$

PCE	Timer Operation Control
0	Operation stopped (timer value is cleared)
1	Operation enabled

Cautions 1. When rewriting MCNTC to other data, stop the timer operation (PCE = 0) beforehand.

2. Bits 0 to 3, 6, and 7 must be set to 0.

(2) Compare control register n (MCMPCn)

MCMPCn is an 8-bit register that controls the operation of the compare register and output direction of the PWM pin.

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

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

Figure 18-4 shows the MCMPCn format.

Figure 18-4. Compare Control Register n (MCMPCn) Format

Address: FF6BH to FF6EH After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MCMPCn	0	0	0	TENn	ADBn1	ADBn0	DIRn1	DIRn0

TENn Note	Transfer Enable Control Bit by Register from Master to Slave
0	Disables data transfer from master to slave. New data can be written.
1	Transfers data from master to slave when MCNT overflows. New data cannot be written.

ADBn1	1-Bit Addition Circuit Control (cos side of meter n)
0	No 1-bit addition to PWM output
1	1-bit addition to PWM output

ADBn0	1-Bit Addition Circuit Control (sin side of meter n)
0	No 1-bit addition to PWM output
1	1-bit addition to PWM output

DIRn1	DIRn0	Output Pin Control			
		SMn1	SMn2	SMn3	SMn4
0	0	PWM	0	PWM	0
0	1	PWM	0	0	PWM
1	0	0	PWM	0	PWM
1	1	0	PWM	PWM	0

Note TENn functions as a control bit and status flag.
As soon as the timer overflows and PWM data is output, TENn is cleared to “0” by hardware.

Caution Bits 5 to 7 must be set to 0.

Remark n = 1 to 4

(3) Port mode control register (PMC)

PMC is an 8-bit register that specifies PWM/port output.

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

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

Figure 18-5 shows the PMC format.

Figure 18-5. Port Mode Control Register (PMC) Format

Address: FF6AH After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PMC	MOD4	MOD3	MOD2	MOD1	EN4	EN3	EN2	EN1

MODn	Meter n Full/Half Bridge Selection
0	Meter n output is full bridge.
1	Meter n output is half bridge.

ENn	Meter n Port/PWM Mode Selection
0	Meter n output is in port mode.
1	Meter n output is in PWM mode.

Remark n = 1 to 4

The relation among the ENn and MODn of the PMC register, DIRn1 and DIRn0 of the MCMPCn register, and output pins is shown below.

ENn	MODn	DIRn1	DIRn0	SMn1 (sin+)	SMn2 (sin–)	SMn3 (cos+)	SMn4 (cos–)	Mode
0	×	×	×	Port	Port	Port	Port	Port mode
1	0	0	0	PWM	0	PWM	0	PWM mode full bridge
1	0	0	1	PWM	0	0	PWM	
1	0	1	0	0	PWM	0	PWM	
1	0	1	1	0	PWM	PWM	0	
1	1	0	0	PWM	Port	PWM	Port	PWM mode half bridge
1	1	0	1	PWM	Port	Port	PWM	
1	1	1	0	Port	PWM	Port	PWM	
1	1	1	1	Port	PWM	PWM	Port	

DIRn1 and DIRn0 mean the quadrant of sin and cos. DIRn1 and DIRn0 = 00 through 11 correspond to quadrants 1 through 4, respectively. The PWM signal is output to the specific pin of the + and – polarities of sin and cos of each quadrant.

When ENn = 0, all the output pins are used as port pins regardless of MODn, DIRn1, and DIRn0.

When ENn = 1 and MODn = 0, the full bridge mode is set, and 0 is output to a pin that does not output a PWM signal.

When ENn = 1 and MODn = 1, the half bridge mode is set, and the pin that does not output a PWM signal is used as a port pin.

Caution The output polarity of the PWM output changes when MCNT overflows.

18.4 Meter Controller/Driver Operations

18.4.1 Basic operation of free-running up counter (MCNT)

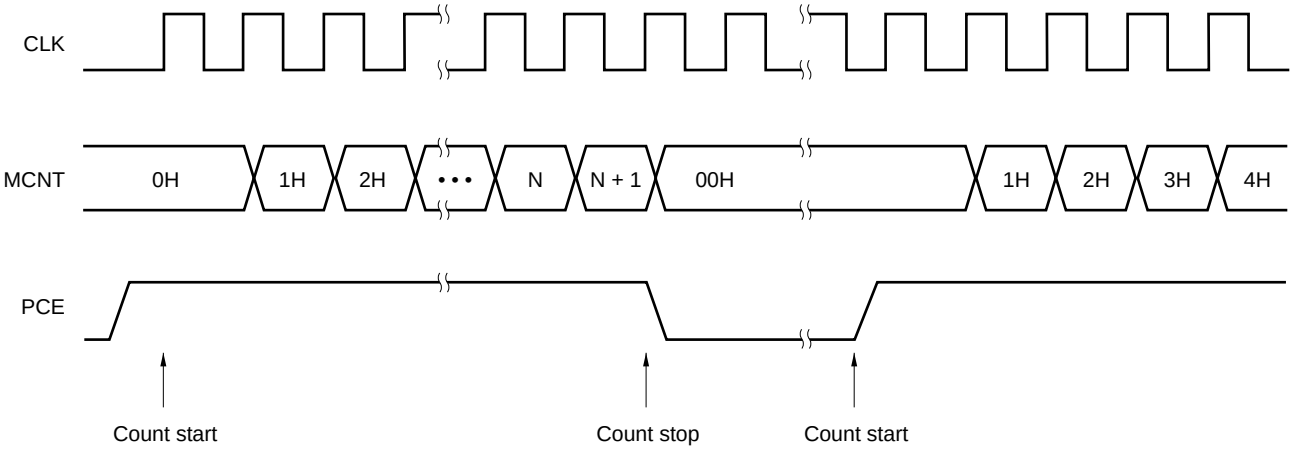
The free-running up counter (MCNT) is counted up by the count clock selected by the PCS bit of the timer mode control register (MCNTC).

$\overline{\text{RESET}}$ input clears the value of MCNT.

The counting operation is enabled or disabled by the PCE bit of MCNTC.

Figure 18-6 shows the timing from count start to restart.

Figure 18-6. Restart Timing after Count Stop (Count Start→Count Stop→Count Start)



Remark N = 00H to FFH

18.4.2 To update PWM data

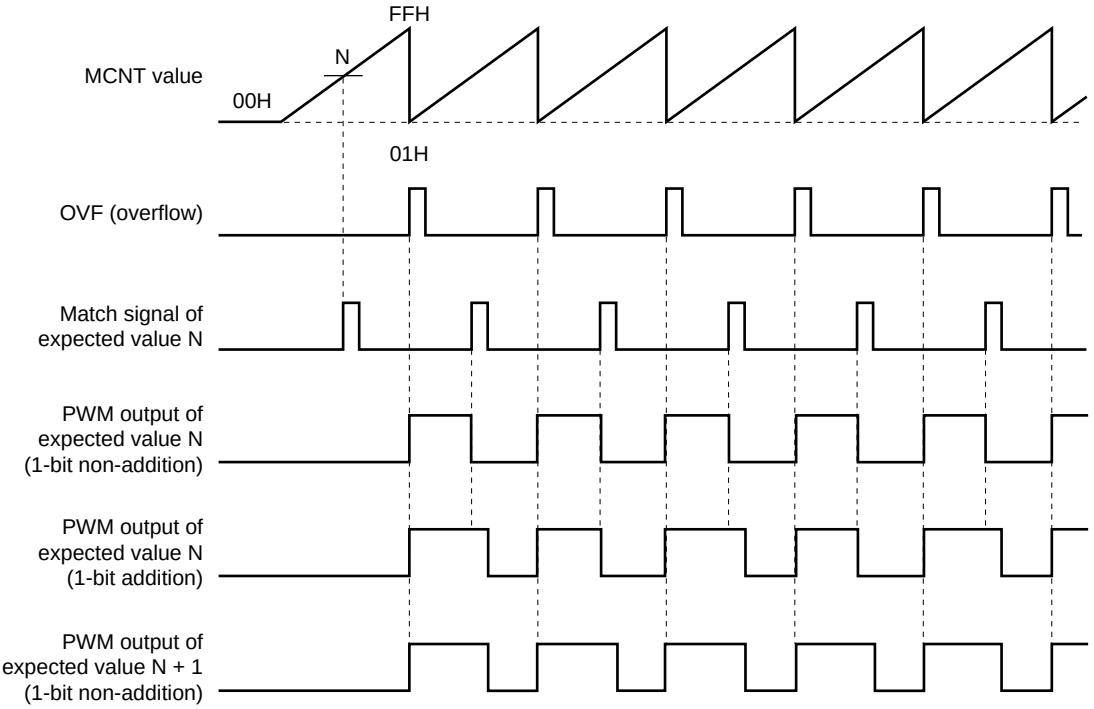
Confirm that bit 4 (TENn) of MCMPCn is 0, and then set 8-bit PWM data to MCMPn1 and MCMPn0, and bits 2 and 3 (ADBn1 and ADBn0) of MCMPCn, and at the same time, set TENn to 1.

The data will be automatically transferred to the slave latch when the timer overflows, and the PWM data becomes valid. At the same time, TENn is automatically cleared to 0.

Remark n = 1 to 4

18.4.3 1-bit addition circuit operation

Figure 18-7. Timing in 1-Bit Addition Circuit Operation

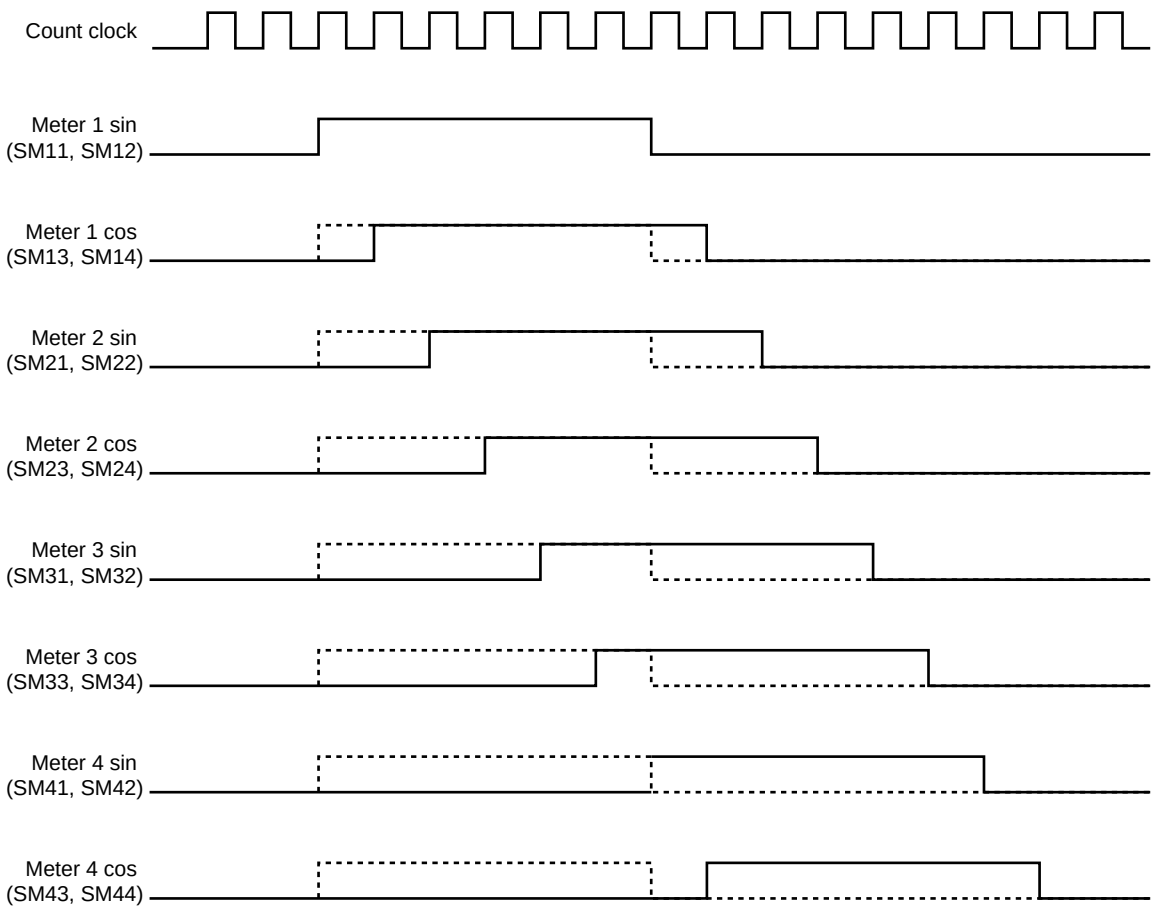


The 1-bit addition mode repeats 1-bit addition/non-addition to PWM output alternately upon MCNT overflow output, and enables the state of PWM output between current compare value N and the next compare value N + 1. In this mode, 1-bit addition to the PWM output is set by setting ADBn of the MCMPCn register to 1, and 1-bit non-addition (normal output) is set by setting ADBn to 0.

Remark n = 1 to 4

18.4.4 PWM output operation (output with 1 clock shifted)

Figure 18-8. Timing of Output with 1 Clock Shifted



If the wave of sin and cos of meters 1 to 4 rises and falls internally as indicated by the broken line, the SM11 to SM44 pins always shift the count clock by 1 clock and output signals, in order to prevent V_{DD}/GND from fluctuating.

[MEMO]

CHAPTER 19 INTERRUPT FUNCTIONS

19.1 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 state. It does not undergo priority control and is given top priority over all other interrupt requests.

A standby release signal is generated.

One interrupt request from the watchdog timer is incorporated as a non-maskable interrupt.

(2) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into a high interrupt priority group and a low interrupt priority group by setting the priority specify flag registers (PR0L, PR0H, PR1L).

High priority interrupts can be issued even if there are low priority interrupts. If two or more interrupts with the same priority are simultaneously generated, each interrupt has a predetermined priority (see **Table 19-1**).

A standby release signal is generated.

Three external interrupt requests and sixteen internal interrupt requests are incorporated as maskable interrupts.

(3) Software interrupt

This is a vectored interrupt to be generated by executing the BRK instruction. It is acknowledged even in the interrupt disabled state. The software interrupt does not undergo interrupt priority control.

19.2 Interrupt Sources and Configuration

A total of 21 interrupt sources exist among non-maskable, maskable, and software interrupts (see **Table 19-1**).

Table 19-1. Interrupt Source List

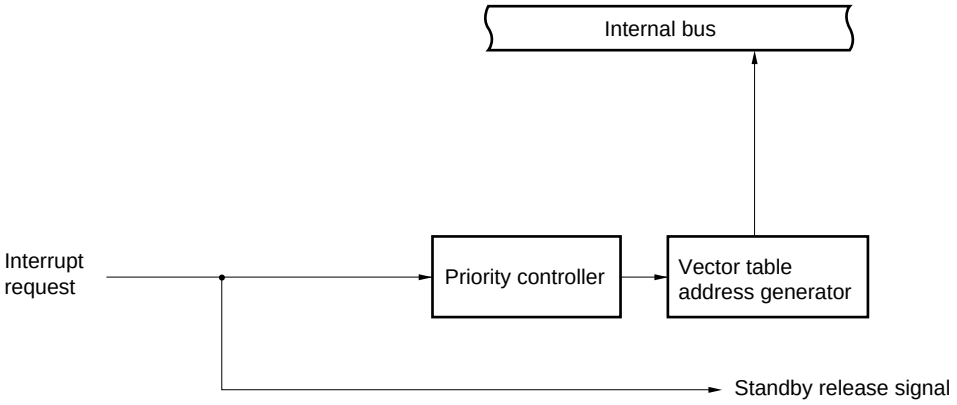
Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2
		Name	Trigger			
Non-maskable	—	INTWDT	Watchdog timer overflow (with non-maskable interrupt selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (with interval timer selected)			(B)
	1	INTAD	End of A/D conversion		0006H	
	2	INTOVF	16-bit timer overflow		0008H	
	3	INTTM00	TI00 valid edge detection		000AH	(C)
	4	INTTM01	TI01 valid edge detection		000CH	
	5	INTTM02	TI02 valid edge detection		000EH	
	6	INTP0	Pin input edge detection	External	0010H	(D)
	7	INTP1			0012H	
	8	INTP2			0014H	
	9	INTCSI3	End of serial interface SIO3 transfer	Internal	0016H	(B)
	10	INTSER	Generation of serial interface UART receive error		0018H	
	11	INTSR	End of serial interface UART reception		001AH	
	12	INTST	End of serial interface UART transmission		001CH	
	13	INTTM1	Generation of 8-bit timer register and capture register (CR1) match signal		001EH	
	14	INTTM2	Generation of 8-bit timer register and capture register (CR2) match signal		0020H	
	15	INTTM3	Generation of 8-bit timer register and capture register (CR3) match signal		0022H	
	16	INTCSI2	End of serial interface SIO2 transfer		0024H	
	17	INTWTI	Watch timer overflow		0026H	
	18	INTWT	Reference time interval signal from watch timer		0028H	
Software	—	BRK	BRK instruction execution	—	003EH	(E)

Notes 1. The default priority is the priority applicable when two or more maskable interrupt requests are generated simultaneously. 0 is the highest priority, and 18 is the lowest.

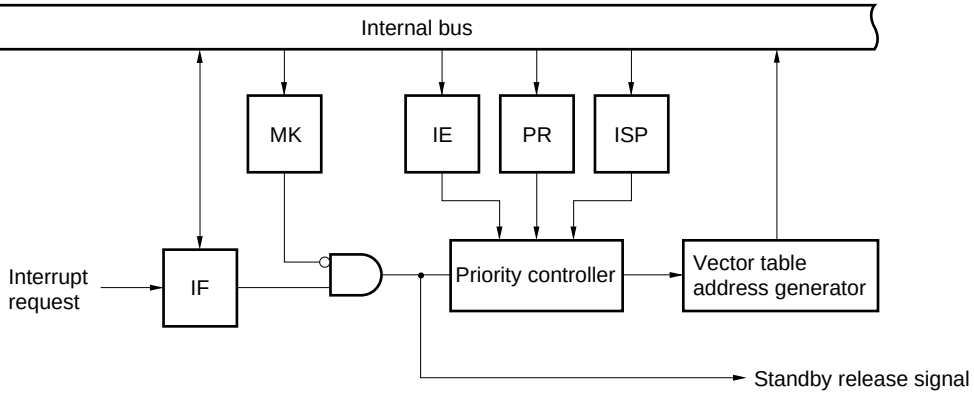
2. Basic configuration types (A) to (E) correspond to (A) to (E) in Figure 19-1.

Figure 19-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt (16-bit timer capture input)

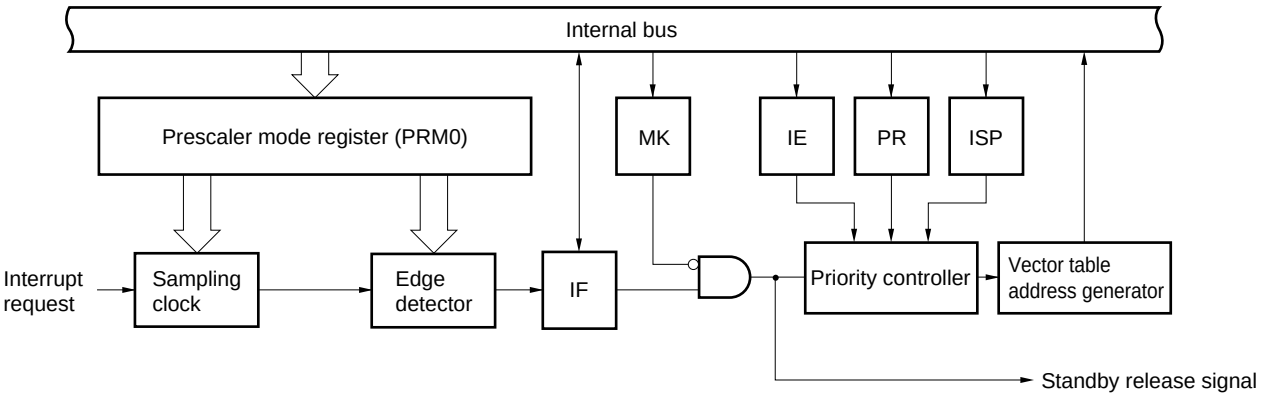
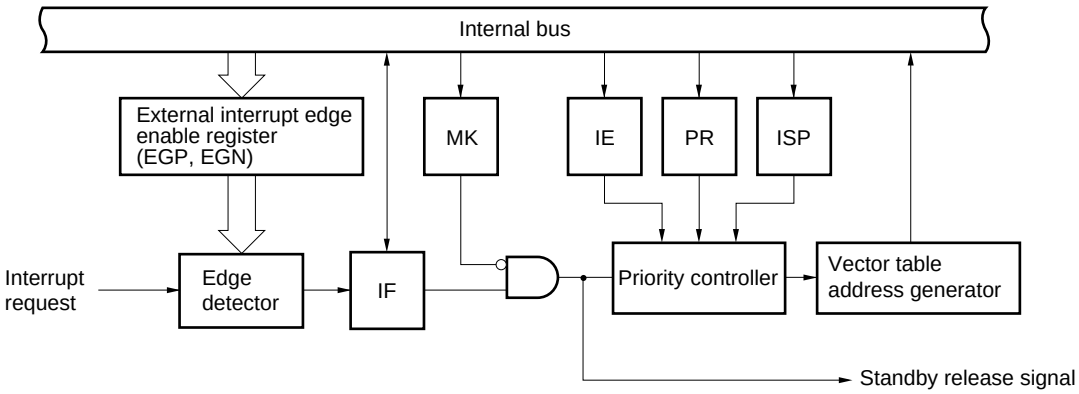
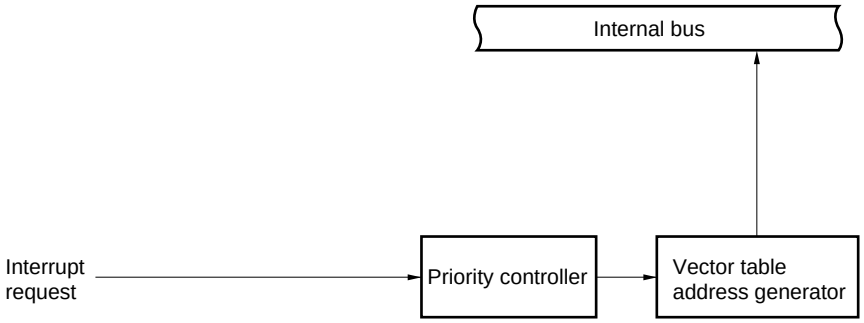


Figure 19-1. Basic Configuration of Interrupt Function (2/2)

(D) External maskable interrupt (except 16-bit timer capture input)



(E) Software interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
ISP: In-service priority flag
MK: Interrupt mask flag
PR: Priority specify flag

19.3 Interrupt Function Control Registers

The following six types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H, IF1L)
- Interrupt mask flag register (MK0L, MK0H, MK1L)
- Priority specify flag register (PR0L, PR0H, PR1L)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)
- Prescaler mode register (PRM0)
- Program status word (PSW)

Table 19-2 gives a list of interrupt request flags, interrupt mask flags, and priority specify flags corresponding to interrupt request sources.

Table 19-2. Flags Corresponding to Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specify Flag	
		Register		Register		Register
INTWDT	WDTIF	IF0L	WDTMK	MK0L	WDTPR	PR0L
INTAD	ADIF		ADMK		ADPR	
INTOVF	OVFIF		OVFMK		OVFPR	
INTTM00	TMIF00		TMMK00		TMPR00	
INTTM01	TMIF01		TMMK01		TMPR01	
INTTM02	TMIF02		TMMK02		TMPR02	
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2	IF0H	PMK2	MK0H	PPR2	PR0H
INTCSI3	CSIIF3		CSIMK3		CSIPR3	
INTSER	SERIF		SERMK		SERPR	
INTSR	SRIF		SRMK		SRPR	
INTST	STIF		STMK		STPR	
INTTM1	TMIF1		TMMK1		TMPR1	
INTTM2	TMIF2		TMMK2		TMPR2	
INTTM3	TMIF3		TMMK3		TMPR3	
INTCSI2	CSIIF2	IF1L	CSIMK2	MK1L	CSIPR2	PR1L
INTWTI	WTIIF		WTIMK		WTIPR	
INTWT	WTIF		WTMK		WTPR	

Remark The WDTIF, WDTMK, and WDTPR flags are interrupt control flags when the watchdog timer is used as an interval timer.

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

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

IF0L, IF0H, and IF1L are set with a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H are combined to form 16-bit register IF0, they are set with a 16-bit memory manipulation instruction.

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

Figure 19-2. Interrupt Request Flag Register (IF0L, IF0H, IF1L) Format

Address: FFE0H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF0L	PIF1	PIF0	TMIF02	TMIF01	TMIF00	OVFIF	ADIF	WDTIF

Address: FFE1H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF0H	TMIF3	TMIF2	TMIF1	STIF	SRIF	SERIF	CSIIF3	PIF2

Address: FFE2H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF1L	0	0	0	0	0	WTIF	WTIIF	CSIIF2

XXIFX	Interrupt Request Flag
0	No interrupt request signal is generated
1	Interrupt request signal is generated, interrupt request status

Cautions 1. The WDTIF flag is R/W enabled only when the watchdog timer is used as the interval timer. If watchdog timer mode 1 is used, set the WDTIF flag to 0.

2. Bits 3 to 7 of IF1L must be set to 0.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt service.

MK0L, MK0H, and MK1L are set with a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H are combined to form a 16-bit register MK0, they are set with a 16-bit memory manipulation instruction.

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

Figure 19-3. Interrupt Mask Flag Register (MK0L, MK0H, MK1L) Format

Address: FFE4H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK0L	PMK1	PMK0	TMMK02	TMMK01	TMMK00	OVFMK	ADMK	WDTMK

Address: FFE5H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK0H	TMMK3	TMMK2	TMMK1	STMK	SRMK	SERMK	CSIMK3	PMK2

Address: FFE6H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK1L	1	1	1	1	1	WTMK	WTIMK	CSIMK2

XXMKX	Interrupt Servicing Control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Cautions 1. If the watchdog timer is used in watchdog timer mode 1, the contents of the WDTMK flag become undefined when read.

2. Because port 0 pins have an alternate function as external interrupt request input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, 1 should be set in the interrupt mask flag before using the output mode.

3. Bits 3 to 7 of MK1L must be set to 1.

(3) Priority specify flag registers (PR0L, PR0H, PR1L)

The priority specify flag registers are used to set the corresponding maskable interrupt priority orders. PR0L, PR0H, and PR1L are set with a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H are combined to form 16-bit register PR0, they are set with a 16-bit memory manipulation instruction.

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

Figure 19-4. Priority Specify Flag Register (PR0L, PR0H, PR1L) Format

Address: FFE8H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR0L	PPR1	PPR0	TMPR02	TMPR01	TMPR00	OVFPR	ADPR	WDTPR

Address: FFE9H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR0H	TMPR3	TMPR2	TMPR1	STPR	SRPR	SERPR	CSIPR3	PPR2

Address: FFEAH After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR1L	1	1	1	1	1	WTPR	WTIPR	CSIPR2

XXPRX	Priority Level Selection
0	High priority level
1	Low priority level

Cautions 1. When the watchdog timer is used in the watchdog timer mode 1, set 1 in the WDTPR flag.

2. Bits 3 to 7 of PR1L must be set to 1.

(4) External interrupt rising edge enable register (EGP) and external interrupt falling edge enable register (EGN)

These registers specify the valid edge for INTP0 to INTP2.

EGP and EGN are set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 19-5. External Interrupt Rising Edge Enable Register (EGP) and External Interrupt Falling Edge Enable Register (EGN) Format

Address: FF48H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP	0	0	0	0	0	EGP2	EGP1	EGP0

Address: FF49H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	0	0	0	0	0	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn Pin Valid Edge (n = 0 to 2)
0	0	Interrupt disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

(5) Prescaler mode register (PRM0)

This register specifies the valid edge for TI00/P40 to TI02/P42 pins input.

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

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

Figure 19-6. Prescaler Mode Register (PRM0) Format

Address: FF70H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM0	ES21	ES20	ES11	ES10	ES01	ES00	PRM01	PRM00

ES21	ES20	TI02 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Interrupt disabled
1	1	Both rising and falling edges

ES11	ES10	TI01 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Interrupt disabled
1	1	Both rising and falling edges

ES01	ES00	TI00 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Interrupt disabled
1	1	Both rising and falling edges

Caution Set the valid edge of the TI00/P40 to TI02/P42 pins after setting bit 2 (TMC02) of 16-bit timer mode control register 0 (TMC0) to 0 to stop the timer operation.

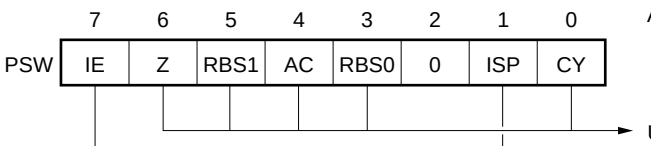
(6) Program status word (PSW)

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

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

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

Figure 19-7. Program Status Word Format



ISP	Priority of Interrupt Currently Being Serviced
0	High-priority interrupt servicing (low-priority interrupt disabled)
1	Interrupt request not acknowledged, or low-priority interrupt servicing. (all maskable interrupts enabled)

IE	Interrupt Request Acknowledge Enable/Disable
0	Disabled
1	Enabled

19.4 Interrupt Servicing Operations

19.4.1 Non-maskable interrupt request acknowledge operation

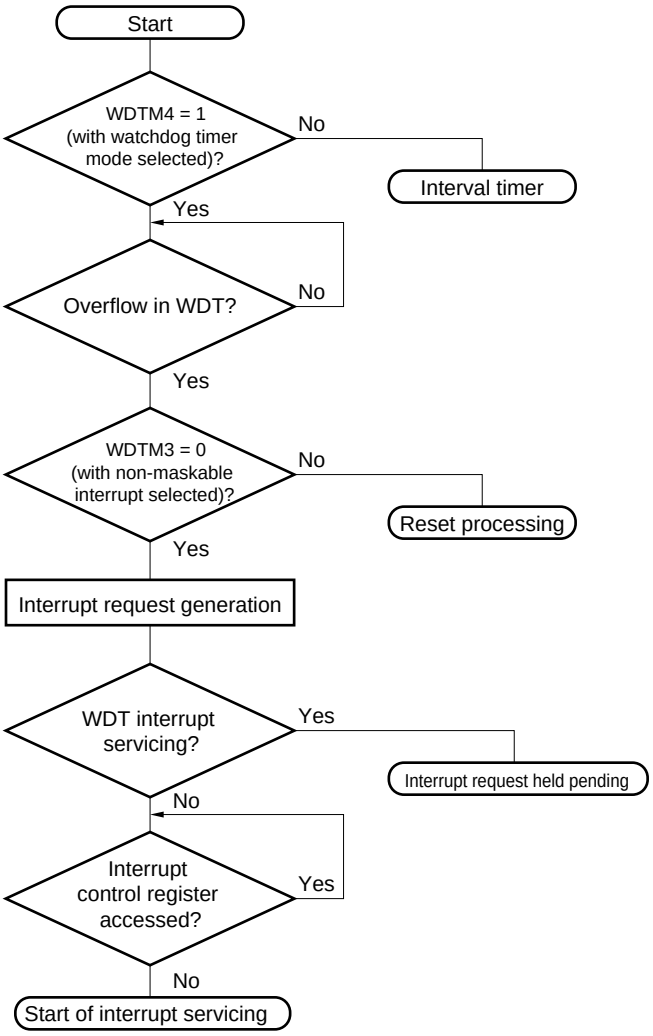
A non-maskable interrupt request is unconditionally acknowledged even if in an interrupt request acknowledge disable state. It does not undergo interrupt priority control and has highest priority over all other interrupts.

If a non-maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag and ISP flag are reset (to 0), and the contents of the vector table are loaded into PC and branched.

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

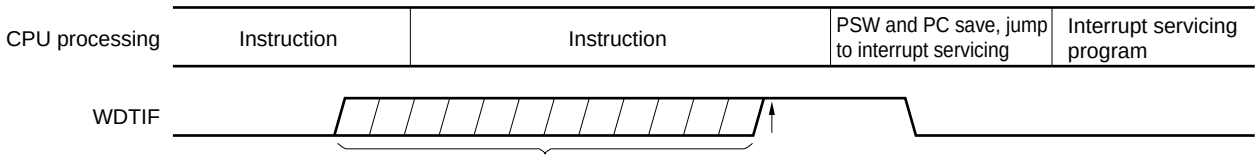
Figures 19-8, 19-9, and 19-10 show the flowchart of the non-maskable interrupt request generation through acknowledge, acknowledge timing of non-maskable interrupt request, and acknowledge operation at multiple non-maskable interrupt request generation, respectively.

Figure 19-8. Non-Maskable Interrupt Request Generation to Acknowledge Flowchart



WDTM: Watchdog timer mode register
WDT: Watchdog timer

Figure 19-9. Non-Maskable Interrupt Request Acknowledge Timing

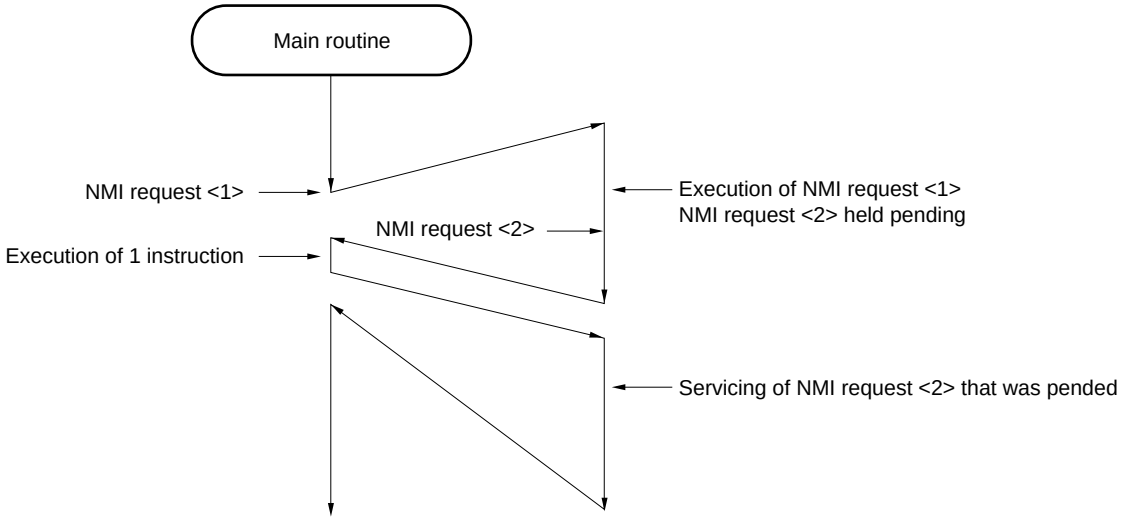


Interrupt request generated during this interval is acknowledged at ↑.

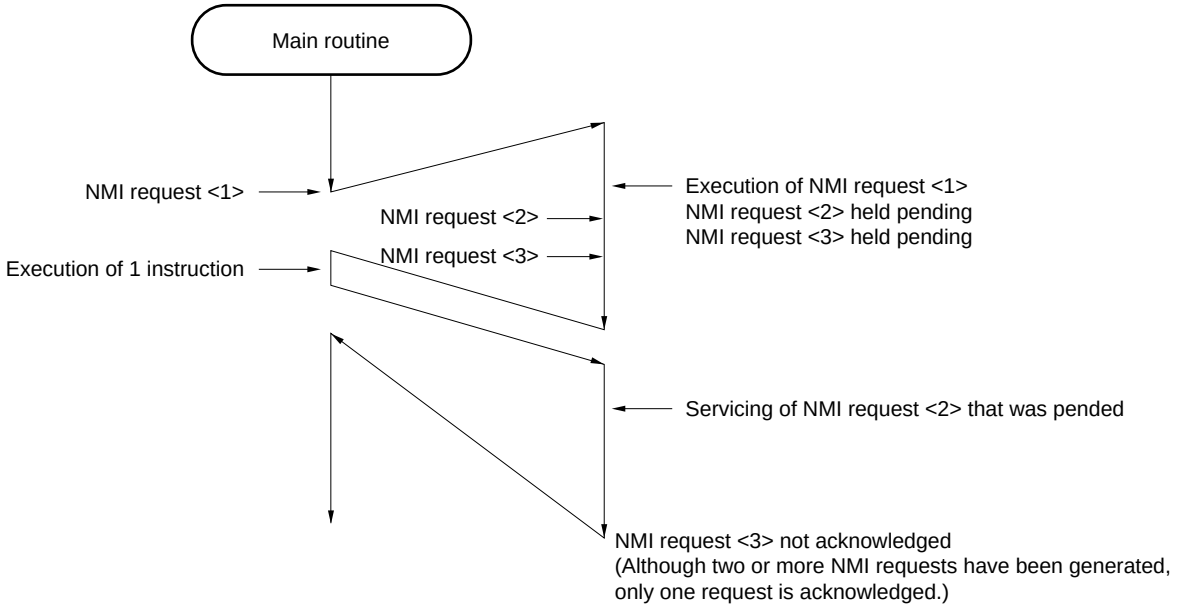
WDTIF: Watchdog timer interrupt request flag

Figure 19-10. Non-Maskable Interrupt Request Acknowledge Operation

(a) If a non-maskable interrupt request is generated during non-maskable interrupt servicing program execution



(b) If two non-maskable interrupt requests are generated during non-maskable interrupt servicing program execution



19.4.2 Maskable interrupt request acknowledge operation

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

The times from generation of a maskable interrupt request until interrupt servicing is performed are listed in Table 19-3 below.

For the interrupt request acknowledge timing, see the Figures 19-12 and 19-13.

Table 19-3. Times from Generation of Maskable Interrupt Request until Servicing

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

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

Remark 1 clock: $1/f_{CPU}$ (f_{CPU}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specify flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

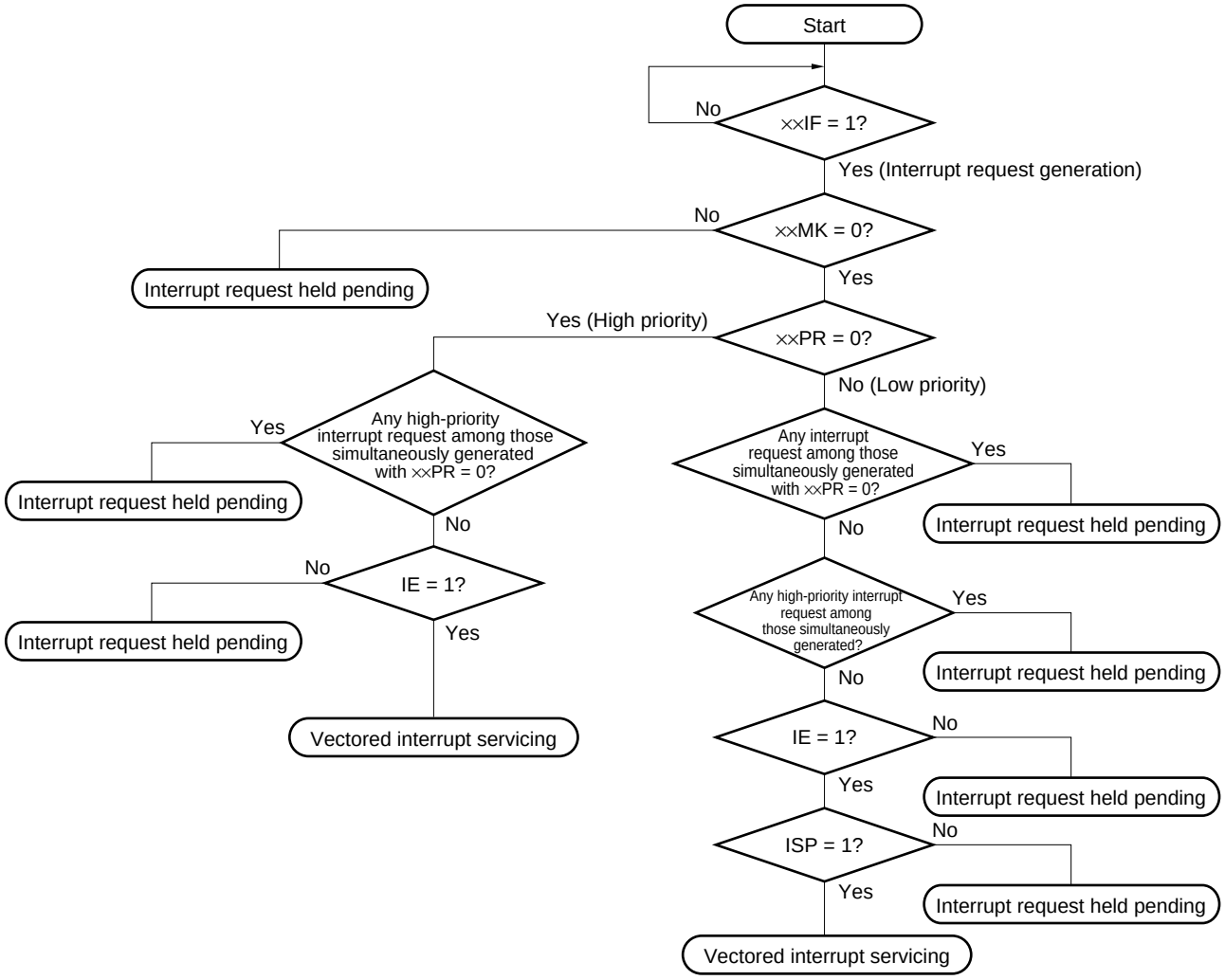
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 19-11 shows the interrupt request acknowledge algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of program status word (PSW), then program counter (PC), the IE flag is reset (to 0), and the contents of the priority specify flag corresponding to the acknowledged interrupt are transferred to the ISP flag. Further, the vector table data determined for each interrupt request is loaded into PC and branched.

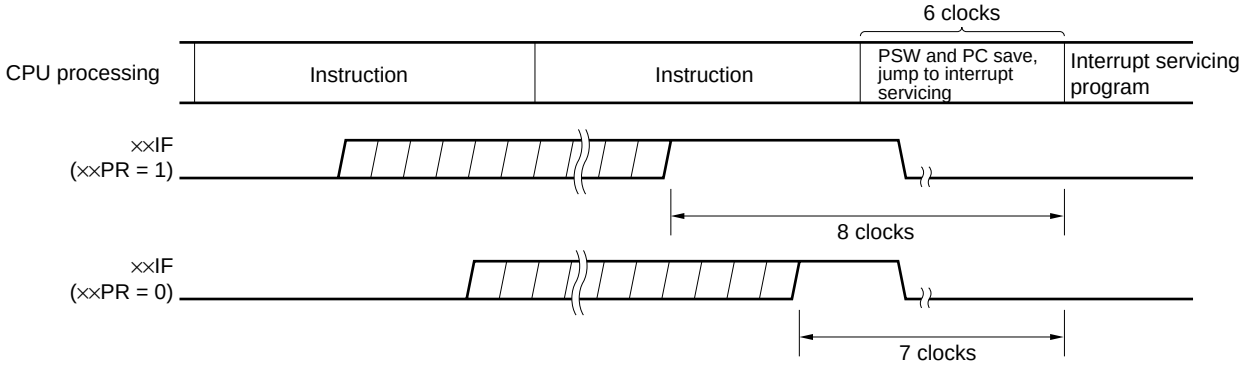
Return from an interrupt is possible with the RETI instruction.

Figure 19-11. Interrupt Request Acknowledge Processing Algorithm



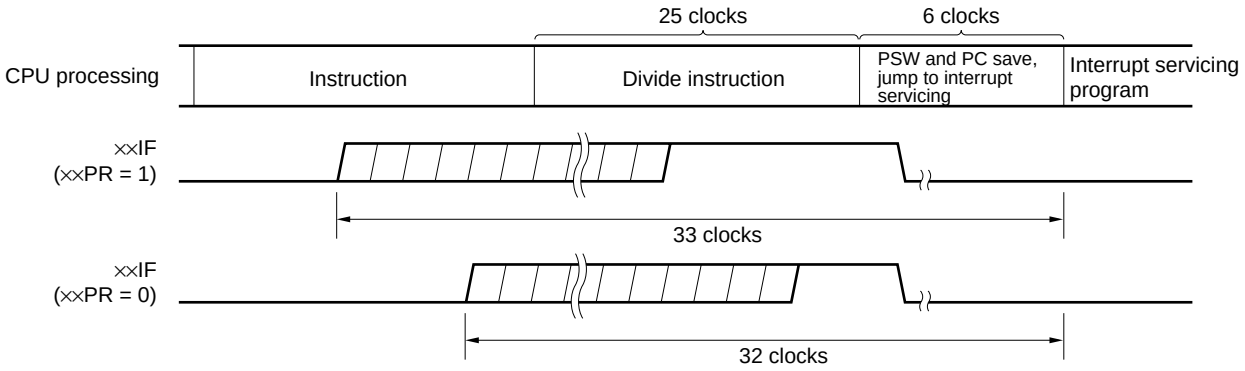
××IF: Interrupt request flag
××MK: Interrupt mask flag
××PR: Priority specify flag
IE: Flag that controls acknowledge of maskable interrupt request (1 = Enable, 0 = Disable)
ISP: Flag that indicates the priority level of the interrupt currently being serviced (0 = High-priority interrupt servicing, 1 = No interrupt request acknowledged, or low-priority interrupt servicing)

Figure 19-12. Interrupt Request Acknowledge Timing (Minimum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

Figure 19-13. Interrupt Request Acknowledge Timing (Maximum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

19.4.3 Software interrupt request acknowledge operation

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

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

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

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

19.4.4 Multiple interrupt servicing

Multiple interrupts occur when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupts do not occur unless the interrupt request acknowledge enable state is selected (IE = 1) (except non-maskable interrupts). Also, when an interrupt request is received, interrupt requests acknowledge becomes disabled (IE = 0). Therefore, to enable multiple interrupts, it is necessary to set the IE flag (to 1) with the EI instruction during interrupt servicing to enable interrupt acknowledge.

Moreover, even if interrupts are enabled, multiple interrupts may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupts.

In the interrupt enable state, if an interrupt request with a priority equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing.

Interrupt requests that are not enabled because of the interrupt disable state or they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of one main processing instruction execution.

Multiple interrupt servicing is not possible during non-maskable interrupt servicing.

Table 19-4 shows interrupt requests enabled for multiple interrupt servicing, and Figure 19-14 shows multiple interrupt examples.

Table 19-4. Interrupt Request Enabled for Multiple Interrupt during Interrupt Servicing

Multiple Interrupt Request / Interrupt Being Serviced		Non-Maskable Interrupt Request	Maskable Interrupt Request			
			××PR = 0		××PR = 1	
			IE = 1	IE = 0	IE = 1	IE = 0
Non-maskable interrupt		×	×	×	×	×
Maskable interrupt	ISP = 0	○	○	×	×	×
	ISP = 1	○	○	×	○	×
Software interrupt		○	○	×	○	×

Remarks 1. ○: Multiple interrupt enable

2. ×: Multiple interrupt disable

3. ISP and IE are flags contained in PSW.

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

ISP = 1: No interrupt request has been acknowledged, or an interrupt with a lower priority is being serviced.

IE = 0: Interrupt request acknowledge is disabled.

IE = 1: Interrupt request acknowledge is enabled.

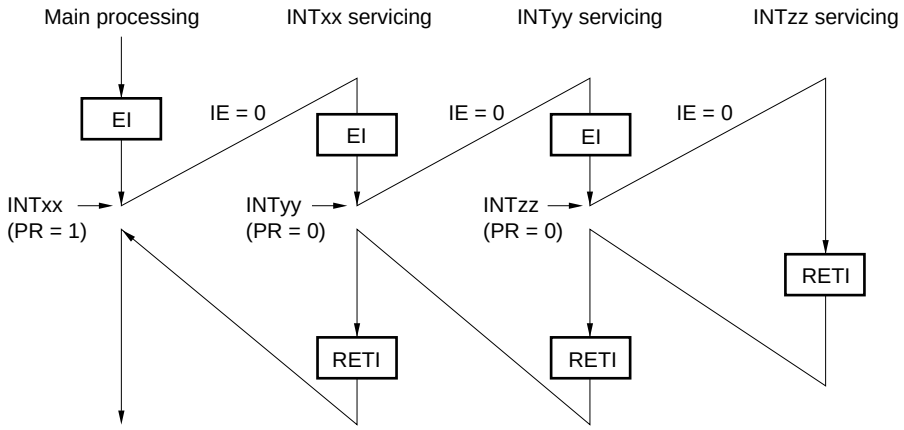
4. ××PR is a flag contained in PR0L, PR0H, PR1L, and PR1H.

××PR = 0: Higher priority level

××PR = 1: Lower priority level

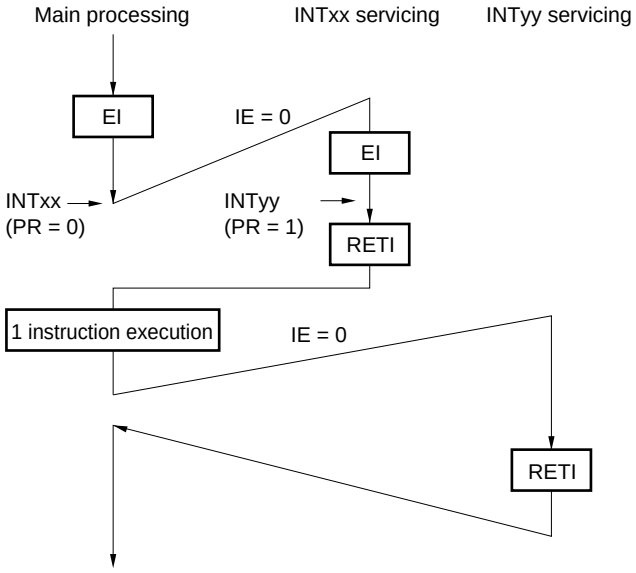
Figure 19-14. Multiple Interrupt Examples (1/2)

Example 1. Multiple interrupts occur twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledge.

Example 2. Multiple interrupt servicing does not occur due to priority control

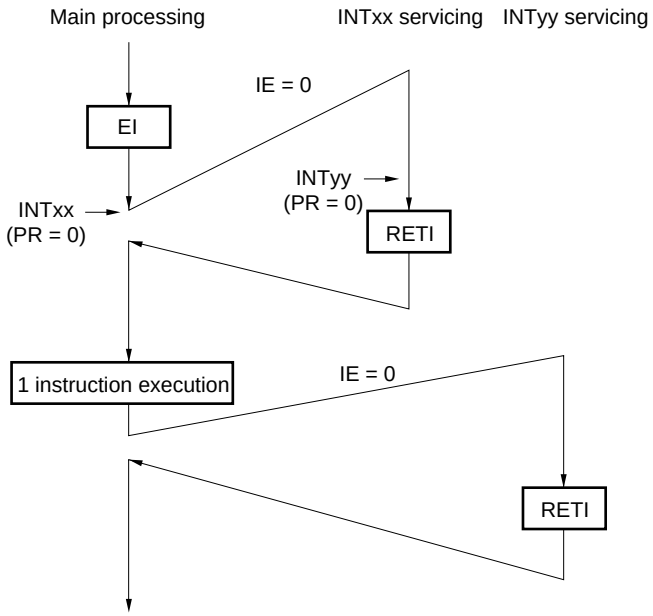


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
PR = 1: Lower priority level
IE = 0: Interrupt request acknowledge disable

Figure 19-14. Multiple Interrupt Examples (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupt is not enabled



Interrupt is not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
IE = 0: Interrupt request acknowledge disable

19.4.5 Interrupt request hold

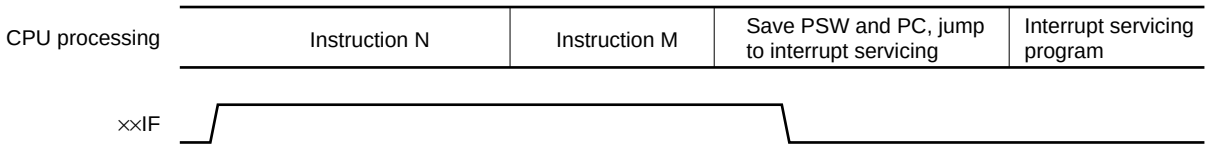
There are instructions where, even if an interrupt request is issued for them while another instruction is executed, request acknowledge is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW.bit, CY
- MOV1 CY, PSW.bit
- AND1 CY, PSW.bit
- OR1 CY, PSW.bit
- XOR1 CY, PSW.bit
- SET1 PSW.bit
- CLR1 PSW.bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW.bit, $addr16
- BF PSW.bit, $addr16
- BTCLR PSW.bit, $addr16
- EI
- DI
- Manipulate instructions for the IF0L, IF0H, IF1L, 1F1H, MK0L, MK0H, MK1L, MK1H, PR0L, PR0H, PR1L, PR1H, EGP, and EGN registers

Caution The BRK instruction is not one of the above-listed interrupt request hold instruction. However, the software interrupt activated by executing the BRK instruction causes the IE flag to be cleared to 0. Therefore, even if a maskable interrupt request is generated during execution of the BRK instruction, the interrupt request is not acknowledged. However, a non-maskable interrupt request is acknowledged.

The timing with which interrupt requests are held pending is shown in Figure 19-15.

Figure 19-15. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

3. The ××PR (priority level) values do not affect the operation of ××IF (interrupt request).

[MEMO]

CHAPTER 20 STANDBY FUNCTION

20.1 Standby Function and Configuration

20.1.1 Standby function

The standby function is designed to decrease power consumption of the system. The following two modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. The HALT mode is intended to stop the CPU operation clock. The system clock oscillator continues oscillating. In this mode, current consumption is not decreased as much as in the STOP mode. However, the HALT mode is effective to restart operation immediately upon interrupt request and to carry out intermittent operations such as watch operation.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the main system clock oscillator stops, stopping the whole system, thereby considerably reducing the CPU current consumption.

Data memory low-voltage hold (down to V_{DD} = 2.0 V) is possible. Thus, the STOP mode is effective to hold data memory contents with ultra-low current consumption.

Because this mode can be cleared upon interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure an oscillation stabilization time after the STOP mode is cleared, select the HALT mode if it is necessary to start processing immediately upon interrupt request.

In either of these two modes, all the contents of registers, flags, and data memory just before the standby mode is set are held. The input/output port output latch and output buffer status are also held.

Cautions 1. When operation is transferred 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: First clear bit 7 (ADCS1) of the A/D converter mode register (ADM1) to 0 to stop the A/D conversion operation, and then execute the HALT or STOP instruction.

20.1.2 Standby function control register

The wait time after the STOP mode is cleared upon interrupt request is controlled with the oscillation stabilization time select register (OSTS).

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

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

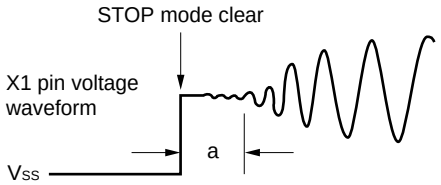
Figure 20-1. Oscillation Stabilization Time Select Register (OSTS) Format

Address: FFFAH After Reset: 04H R/W

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

OSTS2	OSTS1	OSTS0	Oscillation Stabilization Time Selection When STOP Mode Is Cleared
0	0	0	$2^{12}/f_X$ (488 μs)
0	0	1	$2^{14}/f_X$ (1.95 ms)
0	1	0	$2^{15}/f_X$ (3.91 ms)
0	1	1	$2^{16}/f_X$ (7.81 ms)
1	0	0	$2^{17}/f_X$ (15.6 ms)
Other than above			Setting prohibited

Caution The wait time after the STOP mode is cleared does not include the time (see "a" in the illustration below) from STOP mode clear to clock oscillation start, regardless of clearance by $\overline{\text{RESET}}$ input or by interrupt request generation.



Remarks 1. f_X: Main system clock oscillation frequency

2. Figures in parentheses apply to operation with f_X = 8.38 MHz.

20.2 Standby Function Operations

20.2.1 HALT mode

(1) HALT mode setting and operating status

The HALT mode is set by executing the HALT instruction.

The operating status in the HALT mode is described below.

Table 20-1. HALT Mode Operating Status

HALT Mode Setting / Item	During HALT Instruction Execution Using Main System Clock
Clock generator	Main system clock can be oscillated. Clock supply to CPU stops.
CPU	Operation stops.
Port (Output latch)	Status before HALT mode setting is held.
16-bit timer	Operable
8-bit timer	
Watch timer	
Watchdog timer	
A/D converter	Operation stops.
Serial interface	Operable
LCD controller/driver	
External interrupt	
Sound generator	
Meter controller/driver	

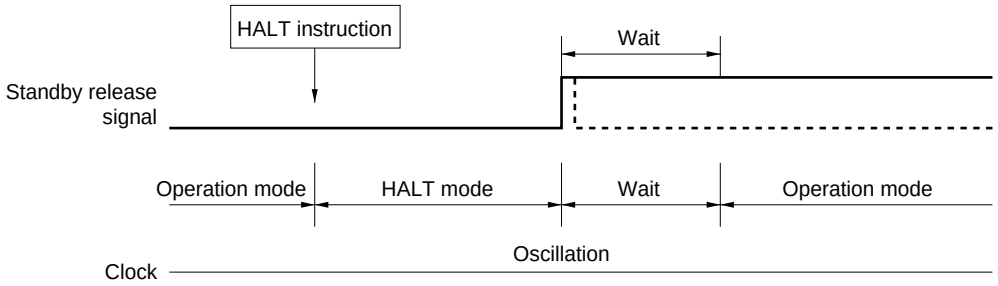
(2) HALT mode clear

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 interrupt acknowledge is disabled, the next address instruction is executed.

Figure 20-2. HALT Mode Clear upon Interrupt Generation



Remarks 1. The broken line indicates the case when the interrupt request which has cleared the standby mode is acknowledged.

2. Wait times are as follows:

- When vectored interrupt service is carried out: 8 to 9 clocks
- When vectored interrupt service is not carried out: 2 to 3 clocks

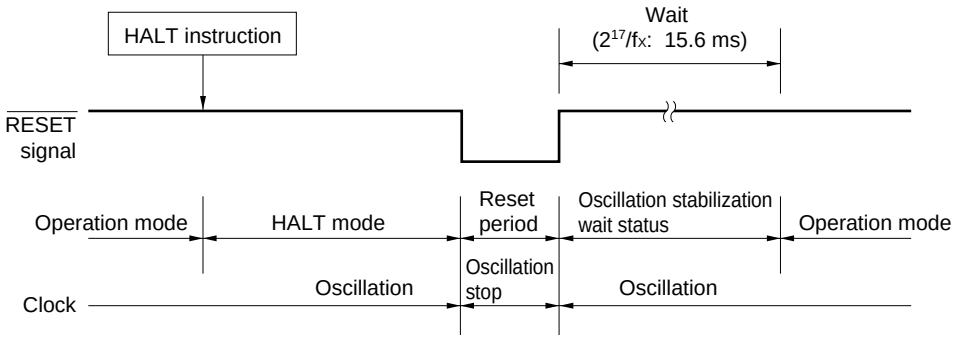
(b) Clear upon non-maskable interrupt request

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

Figure 20-3. HALT Mode Clear upon $\overline{\text{RESET}}$ Input



Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses apply to operation with fx = 8.38 MHz.

Table 20-2. Operation after HALT Mode Clear

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

×: don't care

20.2.2 STOP mode

(1) STOP mode setting 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 operation mode is set.

The operating status in the STOP mode is described below.

Table 20-3. STOP Mode Operating Status

Item \ STOP Mode Setting		During STOP Instruction Execution Using Main System Clock
Clock generator		Only main system clock oscillation is stopped.
CPU		Operation stops.
Port (Output latch)		Status before STOP mode setting is held.
16-bit timer		Operation stops.
8-bit timer		Operable only when TIO2 and TIO3 are selected as count clock.
Watch timer		Operation stops.
Watchdog timer		Operation stops.
A/D converter		Operation stops.
Serial interface	Other than UART	Operable only when externally supplied input clock is specified as the serial clock.
	UART	Operation stops.
LCD controller/driver		Operation stops.
External interrupt		Operable
Sound generator		Operation stops.
Meter controller/driver		Operation stops.

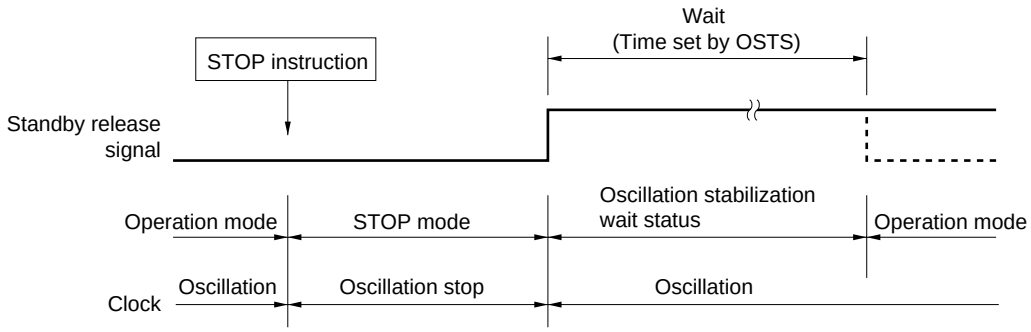
(2) STOP mode clear

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

(a) Clear upon unmasked interrupt request

An unmasked interrupt request is used to clear 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 20-4. STOP Mode Clear upon Interrupt Generation

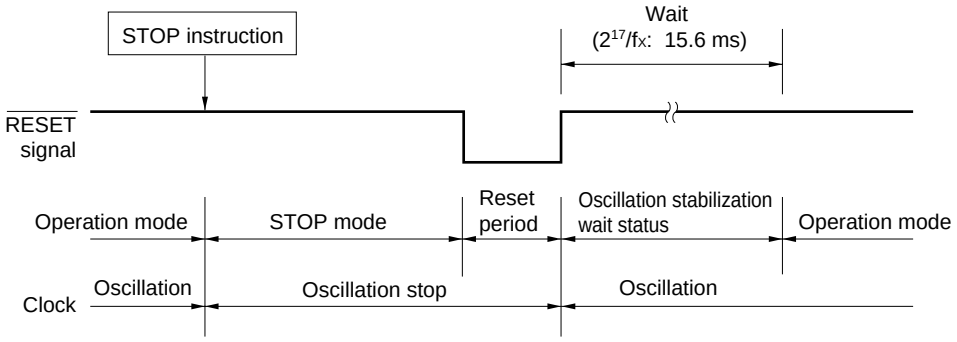


Remark The broken line indicates the case when the interrupt request which has cleared the standby mode is acknowledged.

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

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

Figure 20-5. STOP Mode Clear upon $\overline{\text{RESET}}$ Input



Remarks 1. fx: Main system clock oscillation frequency

2. Figures in parentheses apply to operation with fx = 8.38 MHz.

Table 20-4. Operation after STOP Mode Clear

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

×: don't care

CHAPTER 21 RESET FUNCTION

21.1 Reset Functions

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 shown in Table 21-1. Each pin has high impedance during reset input or during oscillation stabilization time just after reset clear.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is cleared and program execution starts after the lapse of oscillation stabilization time ($2^{17}/f_X$). The reset applied by watchdog timer overflow is automatically cleared after a reset and program execution starts after the lapse of oscillation stabilization time ($2^{17}/f_X$) (see Figures 21-2 to 21-4).

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
2. During reset input, main system clock oscillation stops but subsystem clock oscillation continues.
3. When the STOP mode is cleared by reset, the STOP mode contents are held during reset input. However, the port pin becomes high-impedance.

Figure 21-1. Reset Function Block Diagram

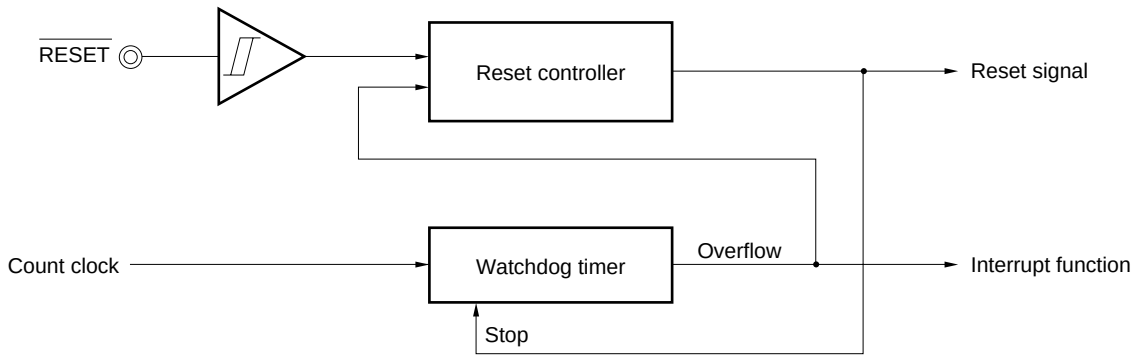


Figure 21-2. Timing of Reset by $\overline{\text{RESET}}$ Input

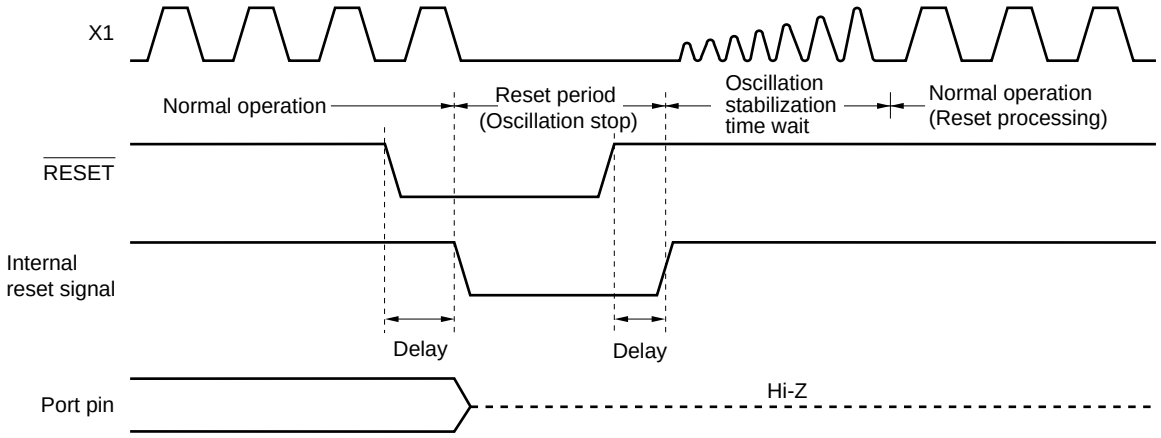


Figure 21-3. Timing of Reset due to Watchdog Timer Overflow

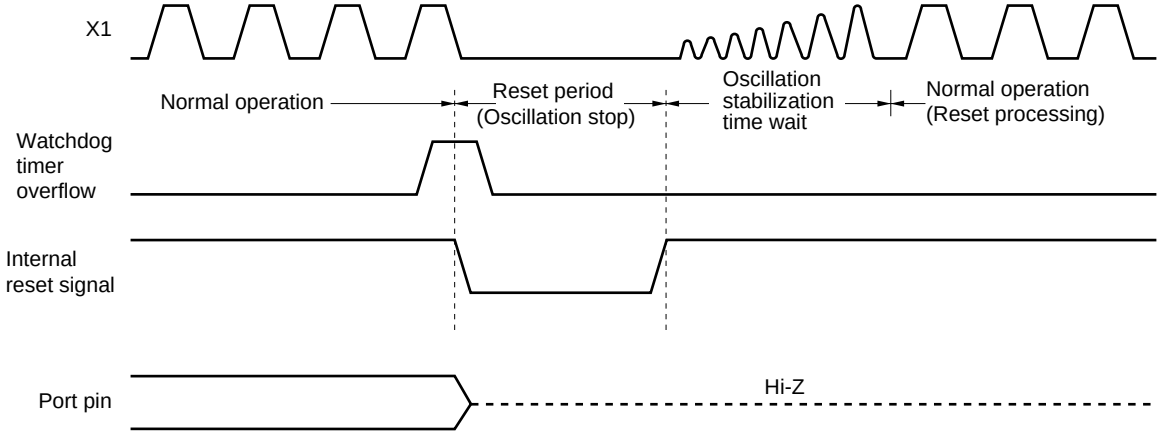


Figure 21-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input

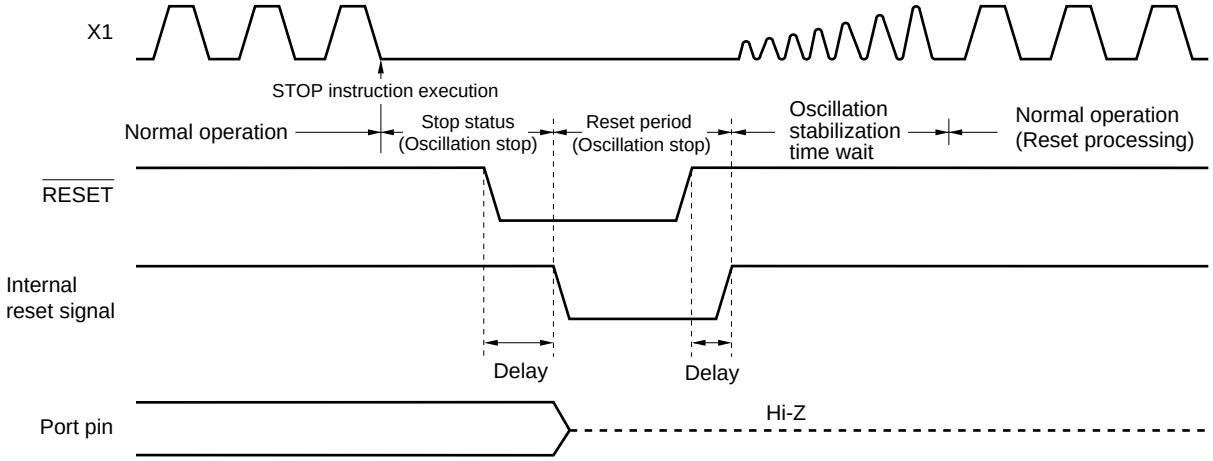


Table 21-1. Hardware Status after Reset (1/2)

Hardware		Status after Reset
Program counter (PC) Note 1		Contents of reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined Note 2
	General register	Undefined Note 2
Port (Output latch)		00H
Port mode registers (PM0 to PM6, PM8, PM9)		FFH
Pull-up resistor option register (PU0)		00H
Processor clock control register (PCC)		04H
Memory size switching register (IMS)		CFH
Internal expansion RAM size switching register (IXS)		0CH
Oscillation stabilization time select register (OSTS)		04H
Oscillator mode register (OSCM) Note 3		00H
16-bit timer 0 TM0	Timer register (TM0)	00H
	Capture registers (CR00 to CR02)	00H
	Prescaler mode register (PRM0)	00H
	Mode control register (TMC0)	00H
	Capture pulse control register 0 (CRC0)	00H

Notes 1. During reset input or oscillation stabilization time wait, only the PC contents among the hardware status become undefined. All other hardware statuses remain unchanged after reset.

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

3. For μPD780851(A), 780852(A) only.

Table 21-1. Hardware Status after Reset (2/2)

Hardware		Status after Reset
8-bit timer TM1 to TM3	Timer counters (TM1 to TM3)	00H
	Compare registers (CR1 to CR3)	00H
	Clock select registers (TCL1 to TCL3)	00H
	Mode control registers (TMC1 to TMC3)	00H
Watch timer	Mode control register (WTM)	00H
Watchdog timer	Clock select register (WDCS)	00H
	Mode register (WDTM)	00H
Clock output controller	Clock output selection register (CKS)	00H
A/D converter	Mode register (ADM1)	00H
	Conversion result register (ADCR1)	00H
	Analog input channel specification register (ADS1)	00H
	Power-fail compare mode register (PFM)	00H
	Power-fail compare threshold value register (PFT)	00H
Serial interface UART	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)	
Serial interface SIO2	Operation mode register 2 (CSIM2)	00H
	Shift register 2 (SIO2)	00H
	Receive data buffer register (SIRB2)	Undefined
	Receive data buffer status register (SRBS2)	00H
Serial interface SIO3	Operation mode register 3 (CSIM3)	00H
	Shift register 3 (SIO3)	00H
LCD controller/driver	Display mode register (LCDM)	00H
	Display control register (LCDC)	00H
Sound generator	Control register (SGCR)	00H
	Amplitude register (SGAM)	00H
	Buzzer control register (SGBR)	00H
Meter controller/driver	Compare registers (MCMP10, MCMP11, MCMP20, MCMP21, MCMP30, MCMP31, MCMP40, MCMP41)	00H
	Timer mode control register (MCNTC)	00H
	Port mode control register (PMC)	00H
	Compare control registers (MCMPC1 to MCMPC3)	00H
Interrupt	Request flag registers (IF0L, IF0H, IF1L)	00H
	Mask flag registers (MK0L, MK0H, MK1L)	FFH
	Priority specify flag registers (PR0L, PR0H, PR1L)	FFH
	External interrupt rising edge enable register (EGP)	00H
	External interrupt falling edge enable register (EGN)	00H

CHAPTER 22 μPD78F0852

The μPD78F0852 is a version with a flash memory in μPD780852 Subseries.

The μPD78F0852 replaces the internal ROM of the μPD780852(A) with flash memory to which a program can be written, deleted and overwritten while mounted on a board. Table 22-1 lists the differences between the μPD78F0852 and the mask ROM version.

Table 22-1. Differences between μPD78F0852 and Mask ROM Version

Item	μPD78F0852	μPD780851(A)	μPD780852(A)
Internal ROM type	Flash memory	Mask ROM	
Internal ROM capacity	40 Kbytes	32 Kbytes	40 Kbytes
IC pin	None	Available	
V_{PP} pin	Available	None	
Electrical specifications	See data sheet of each product.		
Quality grade	Standard (for general electronic devices)	Special (for highly-reliable electronic devices)	

Caution There are differences in noise immunity and noise radiation between the flash memory and mask ROM versions. When pre-producing an application set with the flash memory version and then mass-producing it with the mask ROM version, be sure to conduct sufficient evaluations for the commercial samples (not engineering samples) of the mask ROM version.

22.1 Memory Size Switching Register (IMS)

The μPD78F0852 allows users to select the internal memory capacity using the memory size switching register (IMS) so that the same memory map as that of the mask ROM version with a different size of internal memory capacity can be achieved.

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

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

Figure 22-1. Memory Size Switching Register (IMS) Format

Address: FFF0H After Reset: CFH R/W

Symbol	7	6	5	4	3	2	1	0
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0

RAM2	RAM1	RAM0	Internal High-Speed RAM Capacity Selection
1	1	0	1,024 bytes
Other than above			Setting prohibited

ROM3	ROM2	ROM1	ROM0	Internal ROM Capacity Selection
1	0	0	0	32 Kbytes
1	0	1	0	40 Kbytes
Other than above				Setting prohibited

The IMS settings to obtain the same memory map as the mask ROM version are shown in Table 22-2.

Table 22-2. Memory Size Switching Register Settings

Mask ROM Version	IMS Setting
μPD780851(A)	C8H
μPD780852(A)	CAH

22.2 Internal Expansion RAM Size Switching Register (IXS)

The internal expansion RAM size switching register (IXS) is used to select the capacity of the internal expansion RAM.

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

$\overline{\text{RESET}}$ input sets IXS to 0CH.

Figure 22-2. Internal Expansion RAM Size Switching Register (IXS) Format

Address: FFF4H After Reset: 0CH R/W

Symbol	7	6	5	4	3	2	1	0
IXS	0	0	0	IXRAM4	IXRAM3	IXRAM2	IXRAM1	IXRAM0

IXRAM4	IXRAM3	IXRAM2	IXRAM1	IXRAM0	Internal Expansion RAM Capacity Selection
0	1	0	1	1	512 bytes
Other than above					Setting prohibited

Caution The initial value of IXS is 0CH. Always set 0BH in this register.

22.3 Flash Memory Programming

On-board writing of flash memory (with the device mounted on the target system) is supported.

On-board writing is done after connecting a dedicated flash writer (Flashpro III (part number: FL-PR3, PG-FP3) to the host machine and target system.

Moreover, writing to flash memory can also be performed using a flash memory writing adapter connected to Flashpro III.

Remark Flashpro III is a product of Naito Densei Machida Mfg. Co., Ltd.

22.3.1 Selection of transmission method

Writing to flash memory is performed using Flashpro III and serial communication. Select the transmission method for writing from Table 22-3. For the selection of the transmission method, a format like the one shown in Figure 22-3 is used. The transmission methods are selected with the V_{PP} pulse numbers shown in Table 22-3.

Table 22-3. Transmission Method List

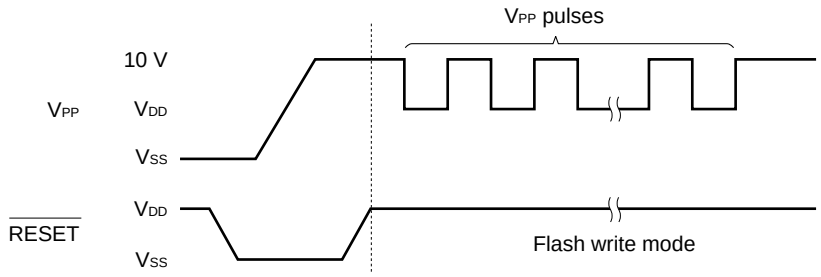
Transmission Method	Number of Channels	Pin Used Note	Number of V_{PP} Pulses
3-wire serial I/O	2	SI3/P52 SO3/P51 $\overline{SCK3}$/P50	0
		SI2/P05 SO2/P04 $\overline{SCK2}$/P03	1
UART	1	RxD/P53 TxD/P54	8

★ **Note** When the device enters the flash memory programming mode, the pins not used for flash memory programming are in the same status as immediately after reset. If the external device connected to each port does not recognize the port status immediately after reset, therefore, the pin must be connected to V_{DD} or V_{SS} via a resistor.

Cautions 1. Be sure to select the number of V_{PP} pulses shown in Table 22-3 for the transmission method.

2. If performing write operations to flash memory with the UART transmission method, set the main system clock oscillation frequency to 4.19 MHz or higher.

Figure 22-3. Transmission Method Selection Format



22.3.2 Flash memory programming function

Flash memory programming functions such as flash memory writing are performed through command and data transmit/receive operations using the selected transmission method. The main functions are listed in Table 22-4.

Table 22-4. Main Functions of Flash Memory Programming

Function	Description
Reset	Detects write stop and transmission synchronization.
Batch verify	Compares entire memory contents and input data.
Batch delete	Deletes the entire memory contents.
Batch blank check	Checks the deletion status of the entire memory.
High-speed write	Performs writing to flash memory according to write start address and number of write data (bytes).
Continuous write	Performs successive write operations using the data input with high-speed write operation.
Status	Checks the current operation mode and operation end.
Oscillation frequency setting	Inputs the resonator oscillation frequency information.
Delete time setting	Inputs the memory delete time.
Baud rate setting	Sets the transmission rate when the UART method is used.
Silicon signature read	Outputs the device name, memory capacity, and device block information.

22.3.3 Flashpro III connection

Connection of Flashpro III and the μPD78F0852 differs depending on the transmission method (3-wire serial I/O and UART). Each case of connection shows in Figures 22-4, 22-5, and 22-6.

Figure 22-4. Flashpro III Connection Using 3-Wire Serial I/O Method (SIO3)

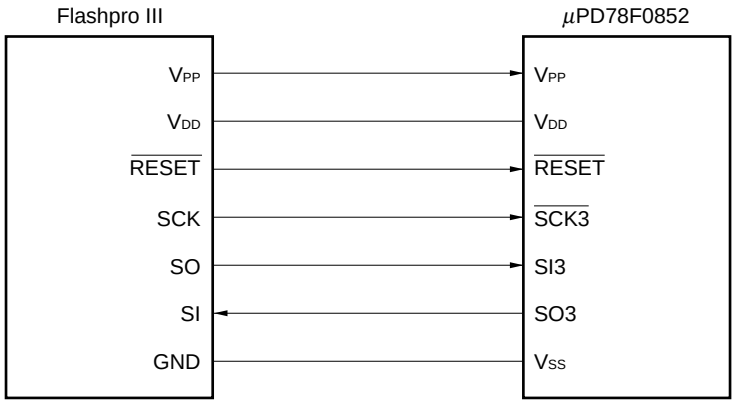


Figure 22-5. Flashpro III Connection Using 3-Wire Serial I/O Method (SIO2)

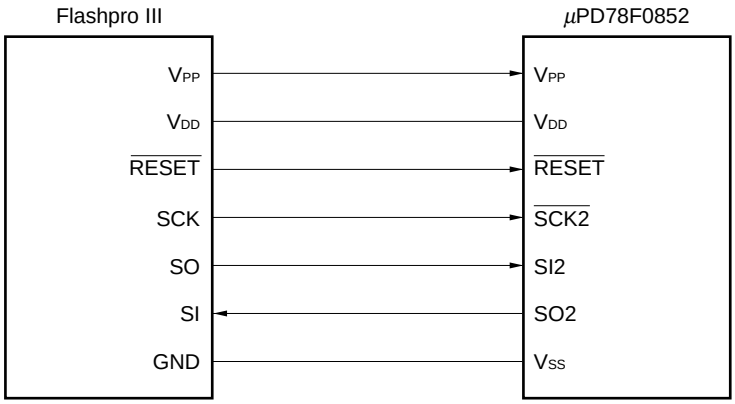
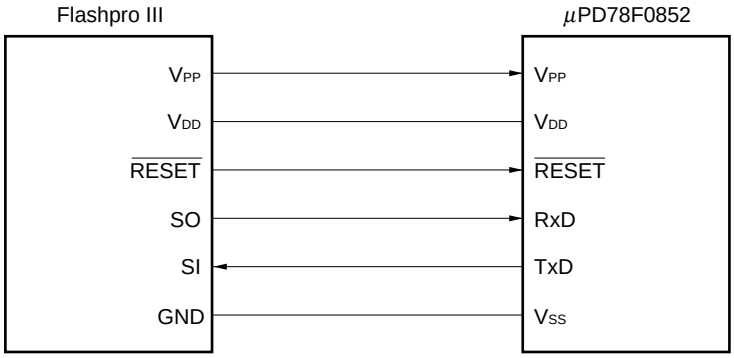


Figure 22-6. Flashpro III Connection Using UART Method



CHAPTER 23 INSTRUCTION SET

This chapter lists the instruction set of the μPD780852 Subseries. For details of the operation and machine language (instruction code), refer to the separate document **78K/0 SERIES USER'S MANUAL Instructions (U12326E)**.

23.1 Legend for Operation List

23.1.1 Operand identifiers and description formats

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

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

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 23-1. Operand Identifiers and Description Formats

Identifier	Description Format
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special-function register symbol **Note**
sfrp	Special-function register symbol (16-bit manipulatable register even addresses only) **Note**
saddr	FE20H to FF1FH Immediate data or labels
saddrp	FE20H to FF1FH Immediate data or labels (even addresses only)
addr16	0000H to FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions)
addr11	0800H to 0FFFH Immediate data or labels
addr5	0040H to 007FH Immediate data or labels (even addresses only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special-function register symbols, see **Table 3-3 Special-Function Register List**.

23.1.2 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)

23.1.3 Description of “flag operation” column

(Blank):	Not affected
0:	Cleared to 0
1:	Set to 1
$\times$:	Set/cleared according to the result
R:	Previously saved value is restored

23.2 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 processor clock control register (PCC).

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 processor clock control register (PCC).

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	SUB	A, #byte	2	4	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r	×	×	×
		r, A	2	4	–	r, CY ← r – A	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr)	×	×	×
		A, !addr16	3	8	9	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C)	×	×	×
	SUBC	A, #byte	2	4	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r – CY	×	×	×
		r, A	2	4	–	r, CY ← r – A – CY	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr) – CY	×	×	×
		A, !addr16	3	8	9	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C) – CY	×	×	×
	AND	A, #byte	2	4	–	A ← A ∧ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	4	–	A ← A ∧ r	×		
		r, A	2	4	–	r ← r ∧ A	×		
		A, saddr	2	4	5	A ← A ∧ (saddr)	×		
		A, !addr16	3	8	9	A ← A ∧ (addr16)	×		
		A, [HL]	1	4	5	A ← A ∧ (HL)	×		
		A, [HL + byte]	2	8	9	A ← A ∧ (HL + byte)	×		
		A, [HL + B]	2	8	9	A ← A ∧ (HL + B)	×		
		A, [HL + C]	2	8	9	A ← A ∧ (HL + C)	×		

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit operation	OR	A, #byte	2	4	–	A ← A∨byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr)∨byte	×
		A, r Note 3	2	4	–	A ← A∨r	×
		r, A	2	4	–	r ← r∨A	×
		A, saddr	2	4	5	A ← A∨(saddr)	×
		A, !addr16	3	8	9	A ← A∨(addr16)	×
		A, [HL]	1	4	5	A ← A∨(HL)	×
		A, [HL + byte]	2	8	9	A ← A∨(HL + byte)	×
		A, [HL + B]	2	8	9	A ← A∨(HL + B)	×
		A, [HL + C]	2	8	9	A ← A∨(HL + C)	×
	XOR	A, #byte	2	4	–	A ← A⊻byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr)⊻byte	×
		A, r Note 3	2	4	–	A ← A⊻r	×
		r, A	2	4	–	r ← r⊻A	×
		A, saddr	2	4	5	A ← A⊻(saddr)	×
		A, !addr16	3	8	9	A ← A⊻(addr16)	×
		A, [HL]	1	4	5	A ← A⊻(HL)	×
		A, [HL + byte]	2	8	9	A ← A⊻(HL + byte)	×
		A, [HL + B]	2	8	9	A ← A⊻(HL + B)	×
		A, [HL + C]	2	8	9	A ← A⊻(HL + C)	×
	CMP	A, #byte	2	4	–	A – byte	× × ×
		saddr, #byte	3	6	8	(saddr) – byte	× × ×
		A, r Note 3	2	4	–	A – r	× × ×
		r, A	2	4	–	r – A	× × ×
		A, saddr	2	4	5	A – (saddr)	× × ×
		A, !addr16	3	8	9	A – (addr16)	× × ×
		A, [HL]	1	4	5	A – (HL)	× × ×
		A, [HL + byte]	2	8	9	A – (HL + byte)	× × ×
		A, [HL + B]	2	8	9	A – (HL + B)	× × ×
		A, [HL + C]	2	8	9	A – (HL + C)	× × ×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

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, $A_7 \leftarrow$ CY, $A_{m-1} \leftarrow A_m$) × 1 time			×
	ROLC	A, 1	1	2	–	(CY ← A_7, $A_0 \leftarrow$ 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 processor clock control register (PCC).

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	**AND1**	CY, saddr.bit	3	6	7	CY ← CY ∧ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ∧ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ∧ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ∧ (HL).bit			×
	OR1	CY, saddr.bit	3	6	7	CY ← CY ∨ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ∨ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ∨ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ∨ (HL).bit			×
	XOR1	CY, saddr.bit	3	6	7	CY ← CY ⊻ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ⊻ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ⊻ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ⊻ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ⊻ (HL).bit			×
	SET1	saddr.bit	2	4	6	(saddr.bit) ← 1			
		sfr.bit	3	–	8	sfr.bit ← 1			
		A.bit	2	4	–	A.bit ← 1			
		PSW.bit	2	–	6	PSW.bit ← 1	×	×	×
		[HL].bit	2	6	8	(HL).bit ← 1			
	CLR1	saddr.bit	2	4	6	(saddr.bit) ← 0			
		sfr.bit	3	–	8	sfr.bit ← 0			
		A.bit	2	4	–	A.bit ← 0			
		PSW.bit	2	–	6	PSW.bit ← 0	×	×	×
		[HL].bit	2	6	8	(HL).bit ← 0			
	SET1	CY	1	2	–	CY ← 1			1
	CLR1	CY	1	2	–	CY ← 0			0
	NOT1	CY	1	2	–	CY ← $\overline{CY}$			×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call/return	**CALL**	!addr16	3	7	–	$(SP-1) \leftarrow (PC+3)_H$, $(SP-2) \leftarrow (PC+3)_L$, $PC \leftarrow$ addr16, $SP \leftarrow SP-2$			
	CALLF	!addr11	2	5	–	$(SP-1) \leftarrow (PC+2)_H$, $(SP-2) \leftarrow (PC+2)_L$, $PC_{15-11} \leftarrow 00001$, $PC_{10-0} \leftarrow$ addr11, $SP \leftarrow SP-2$			
	CALLT	[addr5]	1	6	–	$(SP-1) \leftarrow (PC+1)_H$, $(SP-2) \leftarrow (PC+1)_L$, $PC_H \leftarrow (00000000, addr5+1)$, $PC_L \leftarrow (00000000, addr5)$, $SP \leftarrow SP-2$			
	BRK		1	6	–	$(SP-1) \leftarrow PSW$, $(SP-2) \leftarrow (PC+1)_H$, $(SP-3) \leftarrow (PC+1)_L$, $PC_H \leftarrow (003FH)$, $PC_L \leftarrow (003EH)$, $SP \leftarrow SP-3$, $IE \leftarrow 0$			
	RET		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $SP \leftarrow SP+2$			
	RETI		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP+2)$, $SP \leftarrow SP+3$, $NMIS \leftarrow 0$	R	R	R
	RETB		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP+2)$, $SP \leftarrow SP+3$	R	R	R
Stack manipulate	**PUSH**	PSW	1	2	–	$(SP-1) \leftarrow PSW$, $SP \leftarrow SP-1$			
		rp	1	4	–	$(SP-1) \leftarrow rp_H$, $(SP-2) \leftarrow rp_L$, $SP \leftarrow SP-2$			
	POP	PSW	1	2	–	$PSW \leftarrow (SP)$, $SP \leftarrow SP+1$	R	R	R
		rp	1	4	–	$rp_H \leftarrow (SP+1)$, $rp_L \leftarrow (SP)$, $SP \leftarrow SP+2$			
	MOVW	SP, #word	4	–	10	$SP \leftarrow$ word			
		SP, AX	2	–	8	$SP \leftarrow AX$			
		AX, SP	2	–	8	$AX \leftarrow SP$			
Unconditional branch	**BR**	!addr16	3	6	–	$PC \leftarrow$ addr16			
		$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8			
		AX	2	8	–	$PC_H \leftarrow A$, $PC_L \leftarrow X$			
Conditional branch	**BC**	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if CY = 1			
	BNC	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if CY = 0			
	BZ	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if Z = 1			
	BNZ	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if Z = 0			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Conditional branch	**BT**	saddr.bit, $addr16	3	8	9	PC ← PC + 3 + jdisp8 if (saddr.bit) = 1	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 1	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1	
		PSW.bit, $addr16	3	–	9	PC ← PC + 3 + jdisp8 if PSW.bit = 1	
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 1	
	BF	saddr.bit, $addr16	4	10	11	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 0	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 0	
		PSW.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if PSW.bit = 0	
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 0	
	BTCLR	saddr.bit, $addr16	4	10	12	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1 then reset (saddr.bit)	
		sfr.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit	
		PSW.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	× × ×
		[HL].bit, $addr16	3	10	12	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit	
	DBNZ	B, $addr16	2	6	–	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0	
		C, $addr16	2	6	–	C ← C – 1, then PC ← PC + 2 + jdisp8 if C ≠ 0	
		saddr, $addr16	3	8	10	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if (saddr) ≠ 0	
CPU control	**SEL**	RBn	2	4	–	RBS1, 0 ← n	
	NOP		1	2	–	No Operation	
	EI		2	–	6	IE ← 1 (Enable Interrupt)	
	DI		2	–	6	IE ← 0 (Disable Interrupt)	
	HALT		2	6	–	Set HALT Mode	
	STOP		2	6	–	Set STOP Mode	

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

23.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

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

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

Note Except r = A

(2) 16-bit instructions

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

1st Operand \ Second Operand	#word	AX	rp Note	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW Note						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, HL

(3) Bit manipulation instructions

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

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

(4) Call/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

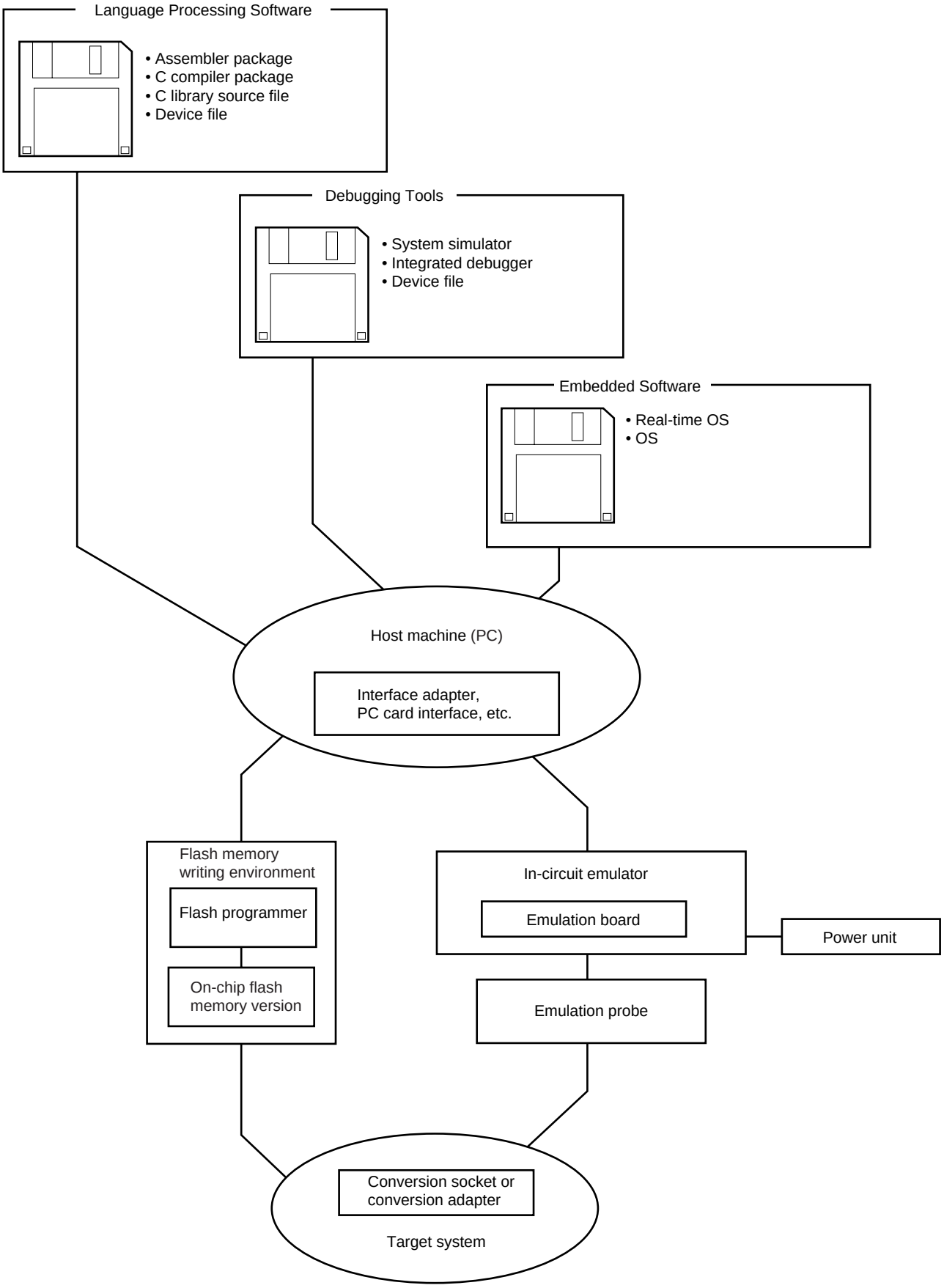
[MEMO]

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the μPD780852 Subseries.

Figure A-1 shows the development tool configuration.

Figure A-1. Development Tool Configuration



A.1 Language Processing Software

RA78K/0 Assembler Package	This assembler converts programs written in mnemonics into an object code executable with a microcontroller. Further, this assembler is provided with functions capable of automatically creating symbol tables and branch instruction optimization. This assembler is used in combination with an optional device file (DF780852). <Caution when using in PC environment> This assembler package is a DOS-based application, however using Project Manager, which is included in the assembler package, enables use of this assembler in a Windows environment.
	Part Number: μS××××RA78K0
CC78K/0 C Compiler Package	This compiler converts programs written in C language into an object code executable with a microcontroller. This compiler is used in combination with an optional assembler package (RA78K/0) and device file (DF780852). <Caution when using in PC environment> This C compiler package is a DOS-based application, however using Project Manager, which is included in the assembler package, enables use of this compiler in a Windows environment.
	Part Number: μS××××CC78K0
DF780852 Device File	This file contains information peculiar to the device. This file is used in combination with each optional tools RA78K/0, CC78K/0, SM78K0, and ID78K0. Supported OS and host machine depend on each tool.
	Part Number: μS××××DF780852
CC78K/0-L C Library Source File	This is a source of functions configuring the object library included in the C compiler package (CC78K/0). It is required to make the object library included in the CC78K/0 conform to the customer's specifications. Operation environment does not depend on OS because the source file is used.
	Part Number: μS××××CC78K0-L

Note The DF780852 is used in common with the RA78K/0, CC78K/0, SM78K0, and ID78K0.

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××RA78K0
μS××××CC78K0
μS××××DF780852
μS××××CC78K0-L

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 Series	Windows Japanese version Note	3.5-inch 2HD FD
AB13	IBM PC/AT™ and compatibles	Windows Japanese version Note	3.5-inch 2HC FD
BB13		Windows English version Note	
3P16	HP9000 Series 700™	HP-UX™ (Rel. 9.05)	DAT (DDS)
3K13	SPARCstation™	SunOS™ (Rel. 4.1.4)	3.5-inch 2HC FD
3K15		Solaris™ (Rel. 2.5.1)	1/4-inch CGMT
3R13	NEWS™ (RISC)	NEWS-OS™ (Rel. 6.1)	3.5-inch 2HC FD

Note WindowsNT™ is not supported.

A.2 Flash Memory Writing Tools

Flashpro III (part number: FL-PR3, PG-FP3) Flash Writer	Dedicated flash writer for microcontrollers with on-chip flash memory.

Remark FL-PR3 is a product of Naito Densei Machida Mfg. Co., Ltd.
For details, contact Naito Densei Machida Mfg. Co., Ltd. (TEL +81-44-822-3813).

A.3 Debugging Tools

A.3.1 Hardware

IE-78K0-NS In-circuit Emulator	This in-circuit emulator is used to debug hardware and software when developing application systems using the 78K/0 Series. It is compatible with the integrated debugger (ID78K0). This emulator is used in combination with an emulation probe and an interface adapter for connection to a host machine.
IE-70000-MC-PS-B Power Unit	This is an adapter for power supply from a receptacle of 100 to 240 VAC.
IE-70000-98-IF-C Interface Adapter	This adapter is required when using the PC-9800 Series computer (except notebook type) as the IE-78K0-NS host machine. (It is compatible with the C bus.)
IE-70000-CD-IF-A PC Card Interface	These PC card and interface cable are required when using a notebook as the IE-78K0-NS host machine. (It is compatible with the PCMCIA socket.)
IE-70000-PC-IF-C Interface Adapter	This adapter is required when using an IBM PC/AT or compatible as the IE-78K0-NS host machine.
IE-70000-PCI-IF Interface Adapter	This adapter is required when using on-chip PCI bus as the IE-78K0-NS host machine.
IE-780852-NS-EM4, IE-78K0-NS-P04 Probe Board	Probe board and I/O board for emulating the μPD780852 Subseries.
NP-80GC-TQ	Emulation probe for 80-pin plastic QFP (GC-8BT type)

Remark NP-80GC-TQ is a product of Naito Densei Machida Mfg. Co., Ltd.
For details, contact Naito Densei Machida Mfg. Co., Ltd. (TEL +81-44-822-3813).

A.3.2 Software (1/2)

SM78K0 System Simulator	This system simulator is used to perform debugging at C source level or assembler level while simulating the operation of the target system on a host machine. The SM78K0 operates on Windows. Use of the SM78K0 allows the execution of application logical testing and performance testing on an independent basis from hardware development without having to use an in-circuit emulator, thereby providing higher development efficiency and software quality. The SM78K0 is used in combination with the optional device file (DF780852).
	Part Number: μS××××SM78K0

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SM78K0

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 Series	Windows Japanese version **Note**	3.5-inch 2HD FD
AB13	IBM PC/AT and compatibles	Windows Japanese version **Note**	3.5-inch 2HC FD
BB13		Windows English version **Note**	

Note WindowsNT is not supported.

A.3.2 Software (2/2)

ID78K0-NS Integrated Debugger (Supports in-circuit emulator IE-78K0-NS) ID78K0 Integrated Debugger (Supports in-circuit emulator IE-78001-R-A)	This is a control program used to debug the 78K/0 Series. The graphical user interfaces employed are Windows for personal computers and OSF/Motif™ for EWSs, offering the standard appearance and operability typical of these interfaces. Further, debugging functions supporting C language are reinforced, and the trace result can be displayed in C language level by using a window integrating function that associates the source program, disassemble display, and memory display with the trace result. In addition, it can enhance the debugging efficiency of a program using a real-time OS by incorporating function expansion modules such as a task debugger and system performance analyzer. This debugger is used in combination with an optional device file (DF780852).
	Part Number: μS××××ID78K0-NS, μS××××ID78K0

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××ID78K0-NS

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 Series	Windows Japanese version **Note**	3.5-inch 2HD FD
AB13	IBM PC/AT and compatibles	Windows Japanese version **Note**	3.5-inch 2HC FD
BB13		Windows English version **Note**	

Note WindowsNT is not supported.

μS××××ID78K0

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 Series	Windows Japanese version **Note**	3.5-inch 2HD FD
AB13	IBM PC/AT and compatibles	Windows Japanese version **Note**	3.5-inch 2HC FD
BB13		Windows English version **Note**	
3P16	HP9000 Series 700	HP-UX (Rel. 9.05)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel. 4.1.4)	3.5-inch 2HC FD
3K15		Solaris (Rel. 2.5.1)	1/4-inch CGMT
3R13	NEWS (RISC)	NEWS-OS (Rel. 6.1)	3.5-inch 2HC FD

Note WindowsNT is not supported.

[MEMO]

APPENDIX B EMBEDDED SOFTWARE

For efficient development and maintenance of the μPD780852 Subseries, the following embedded software products are available.

Real-Time OS (1/2)

RX78K/0 Real-time OS	RX78K/0 is a real-time OS conforming with the μITRON specifications. Tool (configurator) for generating nucleus of RX78K/0 and plural information tables is supplied. Used in combination with an optional assembler package (RA78K/0) and device file (DF780852). <Caution when using in PC environment> Real-time OS is a DOS-based application. Use DOS prompt in Windows.
	Part number: μS××××RX78013-ΔΔΔΔ

Caution When purchasing the RX78K/0, fill in the purchase application form in advance and sign the User Agreement.

Remark ×××× and ΔΔΔΔ in the part number differ depending on the host machine and OS used.

μS××××RX78013-ΔΔΔΔ

ΔΔΔΔ	Product Outline	Upper Limit of Mass-Production Quantity
001	Evaluation object	Do not use for mass-produced products.
100K	Object for mass-produced product	0.1 million units
001M		1 million units
010M		10 million units
S01	Source program	Source program for mass-produced object

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 Series	Windows Japanese version Notes 1, 2	3.5-inch 2HD FD
AB13	IBM PC/AT and compatibles	Windows Japanese version Notes 1, 2	3.5-inch 2HC FD
BB13		Windows English version Notes 1, 2	
3P16	HP9000 Series 700	HP-UX (Rel. 9.05)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel. 4.1.4) Solaris (Rel. 2.5.1)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS (RISC)	NEWS-OS (Rel. 6.1)	3.5-inch 2HC FD

Notes 1. DOS is also supported.

2. WindowsNT is not supported.

Real-Time OS (2/2)

MX78K0 OS	μITRON specification subset OS. Nucleus of MX78K0 is supplied. This OS performs task management, event management, and time management. It controls the task execution sequence for task management and selects the task to be executed next. <Caution when using in PC environment> MX78K0 is a DOS-based application. Use DOS prompt in Windows.
	Part number: μS××××MX78K0-ΔΔΔ

Remark ×××× and ΔΔΔ in the part number differ depending on the host machine and OS used.

μS××××MX78K0-ΔΔΔ

ΔΔΔ	Product Outline	Note
001	Evaluation object	Use for trial product.
XX	Object for mass-produced product	Use for mass-produced product.
S01	Source program	Can be purchased only when object for mass-produced product is purchased.

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 Series	Windows Japanese version Notes 1, 2	3.5-inch 2HD FD
AB13	IBM PC/AT and compatibles	Windows Japanese version Notes 1, 2	3.5-inch 2HC FD
BB13		Windows English version Notes 1, 2	
3P16	HP9000 Series 700	HP-UX (Rel. 9.05)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel. 4.1.4) Solaris (Rel. 2.5.1)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS (RISC)	NEWS-OS (Rel. 6.1)	3.5-inch 2HC FD

Notes 1. DOS is also supported.

2. WindowsNT is not supported.

APPENDIX C REGISTER INDEX

C.1 Register Index (in Alphabetical Order with Respect to Register Name)

C.2 Register Index (in Alphabetical Order with Respect to Register Symbol)

[O]

[P]

[R]

APPENDIX D REVISION HISTORY

The following table shows the revision history of this manual. "Chapter" indicates the chapter of the newest edition where revision was made.

Edition	Major Revision from Previous Edition	Chapter
2nd edition	Changing **1.5 Pin Configuration (Top View)**	**CHAPTER 1 OUTLINE**
	Changing description of supply voltage in **1.8 Outline of Function**	
	Changing **6.4 (4) Port mode register 4 (PM4)**	**CHAPTER 6 16-BIT TIMER 0 TM0**
	Changing **Figure 6-11 Capture Register Data Retention Timing**	
	Adding **Caution 3** to **Figure 16-4 LCD Display Control Register (LCDC) Format**	**CHAPTER 16 LCD CONTROLLER/ DRIVER**
	Adding **Note** to **Table 22-3 Transmission Method List**	**CHAPTER 22 μPD78F0852**

{MEMO]

NEC

Facsimile Message

From:

Name

Company

Tel. FAX

Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: 1-800-729-9288
1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Technical Documentation Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6462-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

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

Taiwan
NEC Electronics Taiwan Ltd.
Fax: 02-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: 044-435-9608

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	❑	❑	❑	❑

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 8-bit Microcontrollers - MCU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

CY8C20524-12PVXIT CY8C28433-24PVXIT MB95F012KPFT-G-SNE2 MB95F013KPMC-G-SNE2 MB95F263KPF-G-SNE2 MB95F264KPFT-G-SNE2 MB95F398KPMC-G-SNE2 MB95F478KPMC2-G-SNE2 MB95F562KPF-G-SNE2 MB95F564KPF-G-SNE2 MB95F634KPMC-G-SNE2 MB95F636KWQN-G-SNE1 MB95F696KPMC-G-SNE2 MB95F698KPMC1-G-SNE2 MB95F698KPMC2-G-SNE2 MB95F698KPMC-G-SNE2 MB95F818KPMC1-G-SNE2 MC908JK1ECDWER MC9S08PA32AVLD MC9S08PT60AVLD R5F1076CMSPV0 R5F5631ECDFBV0 C8051F389-B-GQ C8051F392-A-GMR ISD-ES1600_USB_PROG 901015X SC705C8AE0VFBE STM8TL53G4U6 PIC16F877-04/P-B R5F10Y17ASP#30 CY8C3MFIDOCK-125 403708R MB95F354EPF-G-SNE2 MB95F564KPFT-G-SNE2 MB95F564KWQN-G-SNE1 MB95F636KP-G-SH-SNE2 MB95F636KPMC-G-SNE2 MB95F694KPMC-G-SNE2 MB95F778JPMC1-G-SNE2 MB95F818KPMC-G-SNE2 MC908QY8CDWER MC9S08PT16AVLD MC9S08PT32AVLH MC9S08PT60AVLC MC9S08PT60AVLH C8051F500-IQR LC87F0G08AUJA-AH CP8361BT STM8S207C6T3 CG8421AF