NEC

User's Manual

μPD178054 Subseries

8-Bit Single-Chip Microcontrollers

μPD178053
μPD178054
μPD178F054

Document No. U15104EJ2V0UD00 (2nd edition)
Date Published January 2002 N CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

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

License not needed: μPD178F054GC-8BT
The customer must judge the need for license: μPD178053GC-×××-8BT, 178054GC-×××-8BT

- **The information in this document is current as of October, 2001. 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: 01-3067-5800
Fax: 01-3067-5899

NEC Electronics (France) S.A.
Madrid Office
Madrid, Spain
Tel: 091-504-2787
Fax: 091-504-2860

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.
Novena Square, Singapore
Tel: 253-8311
Fax: 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: 11-6462-6810
Fax: 11-6462-6829

J01.2

Major Revisions in This Edition

Page	Description
Throughout	Change of μPD178053, 178054, and 178F054 status from under development to development completed
pp.8, 9	Modification of **Related Documents**
p.25	Modification of **1.5 Development of 8-Bit DTS Series**
p.55	Modification of bit units for manipulation for OSTS in **Table 3-4 Special Function Registers**
p.84	Deletion of pins P10 to P15 from **Table 4-3 Port Mode Register and Output Latch Settings When Using Alternate Functions**
p.124	Modification of description in **(3) Oscillation stabilization time select register (OSTS)** in **8.3 Registers Controlling Watchdog Timer**
p.240	Addition of **CHAPTER 19 ELECTRICAL SPECIFICATIONS**
p.250	Addition of **CHAPTER 20 PACKAGE DRAWING**
p.251	Addition of **CHAPTER 21 RECOMMENDED SOLDERING CONDITIONS**
p.253	Modification of **Figure A-1 Configuration of Development Tools**
pp.255, 256	Addition of **A.1 Software Package** and **A.3 Control Software**
p.255	Addition of **Note 2** to **A.2 Language Processing Software**
p.257	Addition of description for IE-78K0-NS-A to **A.5 Debugging Tools (Hardware)**
p.260	Deletion of MX78K0 from **A.7 Embedded Software**

The mark ★ shows major revised points.

PREFACE

Readers

This manual has been prepared for user engineers who wish to understand the functions of the μPD178054 Subseries and design and develop its application systems and programs.

Purpose

This manual is intended to give users an understanding of the functions described in the Organization below.

Organization

The μPD178054 Subseries manual is separated into two parts: this manual and the instruction edition (common to the 78K/0 Series).

μPD178054 Subseries User's Manual	78K/0 Series User's Manual Instruction
• Pin functions • Internal block functions • Interrupt • Other on-chip peripheral functions • Electrical specifications	• CPU functions • Instruction set • Explanation of each instruction

How to Read This Manual

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

- When you want to understand the functions in general:
  → Read this manual in the order of the contents.
- To know the μPD178054 Subseries instruction function in detail:
  → Refer to the **78K/0 Series User's Manual Instructions (U12326E)**
- How to interpret the register format:
  → For the circled bit number, the bit name is defined as a reserved word in DF178054 and RA78K0, and in CC78K0, already defined in the header file named sfrbit.h.
- To know the electrical specifications of the μPD178054 Subseries:
  → Refer to **CHAPTER 19 ELECTRICAL SPECIFICATIONS**.

Conventions

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

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

Documents Related to Devices

Document Name		Document No.
μPD178054 Subseries User's Manual		This manual
78K/0 Series Instruction User's Manual		U12326E
78K/0 Series Application Note	Basics (I)	U12704E

Documents Related to Development Tools (Software) (User's Manuals)

Document Name		Document No.
RA78K0 Assembler Package	Operation	U14445E
	Assembly Language	U14446E
	Structured Assembly Language	U11789E
CC78K0 C Compiler	Operation	U14297E
	Language	U14298E
SM78K0S, SM78K0 System Simulator Ver.2.10 or Later Windows™ Based	Operation	U14611E
SM78K Series System Simulator Ver.2.10 or Later	External Part User Open Interface Specifications	U15006E
ID78K0-NS Integrated Debugger Ver.2.00 or Later Windows Based	Operation	U14379E
ID78K0 Integrated Debugger Windows Based	Reference	U11539E
	Guide	U11649E
RX78K0 Real-Time OS	Fundamental	U11537E
	Installation	U11536E
Project Manager Ver. 3.12 or Later (Windows-Based)		U14610E

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document for designing.

Documents Related to Development Tools (Hardware) (User's Manuals)

Document Name	Document No.
IE-78K0-NS In-Circuit Emulator	U13731E
IE-78K0-NS-A In-Circuit Emulator	U14889E
IE-178054-NS-EM1 Emulation Board	To be prepared

Documents Related to Flash ROM Writing

Document Name	Document No.
PG-FP3 Flash Memory Programmer User's Manual	U13502E

Other Related Documents

Document Name	Document No.
SEMICONDUCTOR SELECTION GUIDE -Products & Packages-	X13769E
Semiconductor Device Mounting Technology Manual	C10535E
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document for designing.

TABLE OF CONTENTS

LIST OF FIGURES (1/4)

LIST OF FIGURES (2/4)

LIST OF FIGURES (3/4)

LIST OF FIGURES (4/4)

LIST OF TABLES (1/2)

LIST OF TABLES (2/2)

CHAPTER 1 OUTLINE

1.1 Features

- Internal ROM and RAM

Item / Part Number	Program Memory		Data Memory
			Internal High-Speed RAM
μPD178053	ROM	24 KB	1024 bytes
μPD178054		32 KB	
μPD178F054	Flash memory	32 KB	

- Instruction set suitable for system control
  - Bit processing across entire address space
  - Multiplication/division instructions
- General-purpose I/O ports: 62 pins
- Hardware for PLL frequency synthesizer
  - Dual modulus prescaler (160 MHz MAX.)
  - Programmable divider
  - Phase comparator
  - Charge pump
- Frequency counter
- 8-bit resolution A/D converter: 6 channels
- Serial interface: 3 channels
  - 3-wire serial I/O mode: 2 channels
  - 3-wire serial I/O mode (on-chip time-division transfer function): 1 channel
- Timer: 6 channels
  - Basic timer (timer carry FF): 1 channel
  - 8-bit timer/event counter: 4 channels
  - Watchdog timer: 1 channel
- Buzzer output
- Vectored interrupt

Item / Part Number	Non-Maskable Interrupt[Note]	Maskable Interrupt[Note]		Software Interrupt
		External	Internal	
μPD178053, 178054, 178F054	1 source	5 sources	11 sources	1 source

Note Either a non-maskable interrupt or maskable interrupt (internal) can be selected as the interrupt source of the watchdog timer (INTWDT).

- Test input: 1 pin
- Instruction cycle: 0.45/0.89/1.78/3.56/7.11 μs (with 4.5 MHz crystal resonator)
- Supply voltage: V_{DD} = 4.5 to 5.5 V (with CPU, PLL operating)
  V_{DD} = 3.5 to 5.5 V (with CPU operating)
- Power-on clear circuit

1.2 Applications

Car stereos

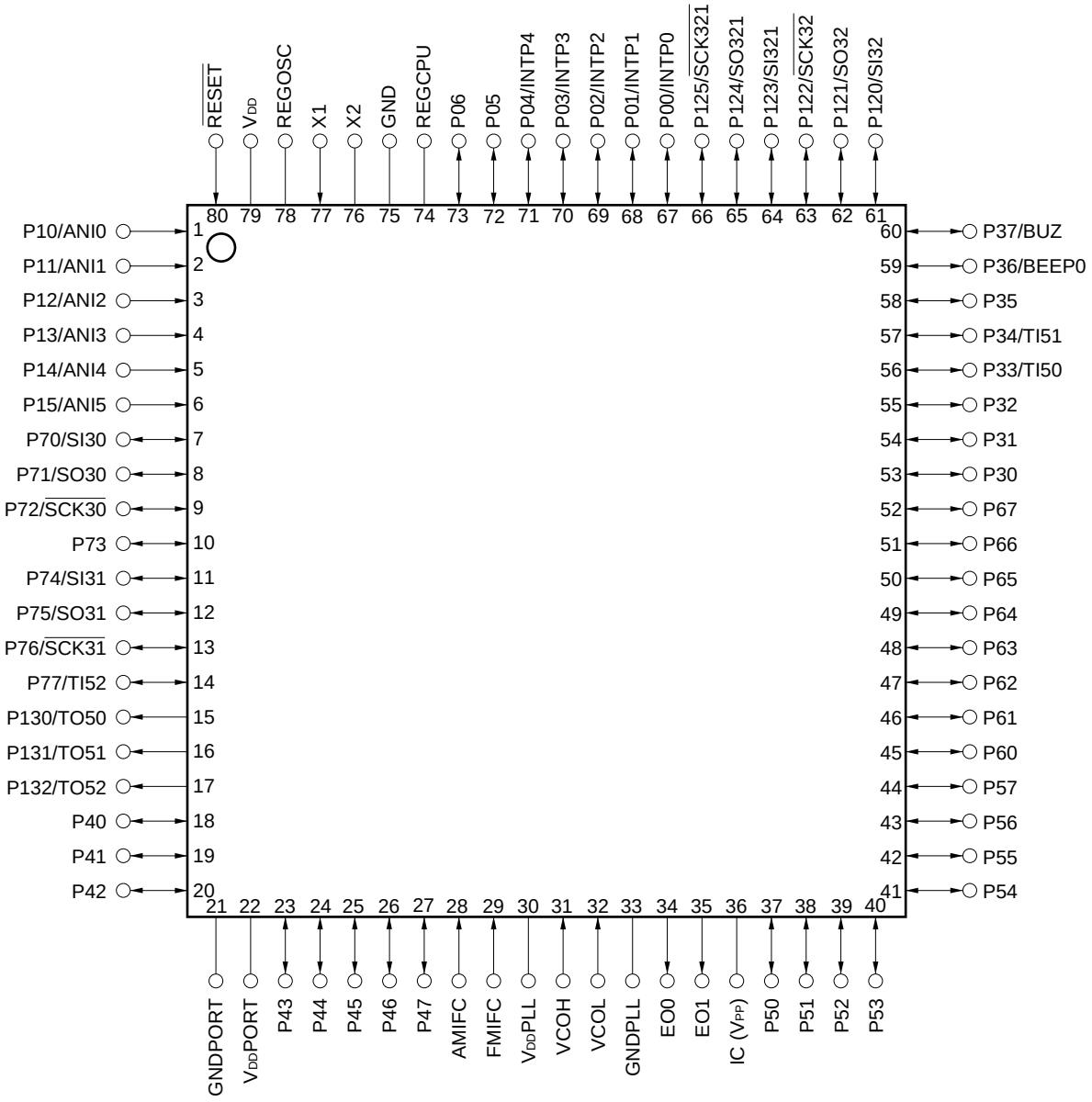
1.3 Ordering Information

Part Number	Package
μPD178053GC-×××-8BT	80-pin plastic QFP (14 × 14)
μPD178054GC-×××-8BT	80-pin plastic QFP (14 × 14)
μPD178F054GC-8BT	80-pin plastic QFP (14 × 14)

Remark ××× indicates ROM code suffix.

1.4 Pin Configuration (Top View)

- **80-pin plastic QFP (14 × 14)**
  μPD178053GC-×××-8BT, 178054GC-×××-8BT, 178F054GC-8BT



Cautions 1. Directly connect the IC (Internally Connected) pin and V_{PP} pin to GND.

2. Keep the V_{DD}PORT and V_{DD}PLL pins as same potential as that at the V_{DD} pin.

3. Keep the GNDPORT and GNDPLL pins as same potential as that at GND.

4. Connect each of the REGOSC and REGCPU pins to GND via a 0.1 μF capacitor.

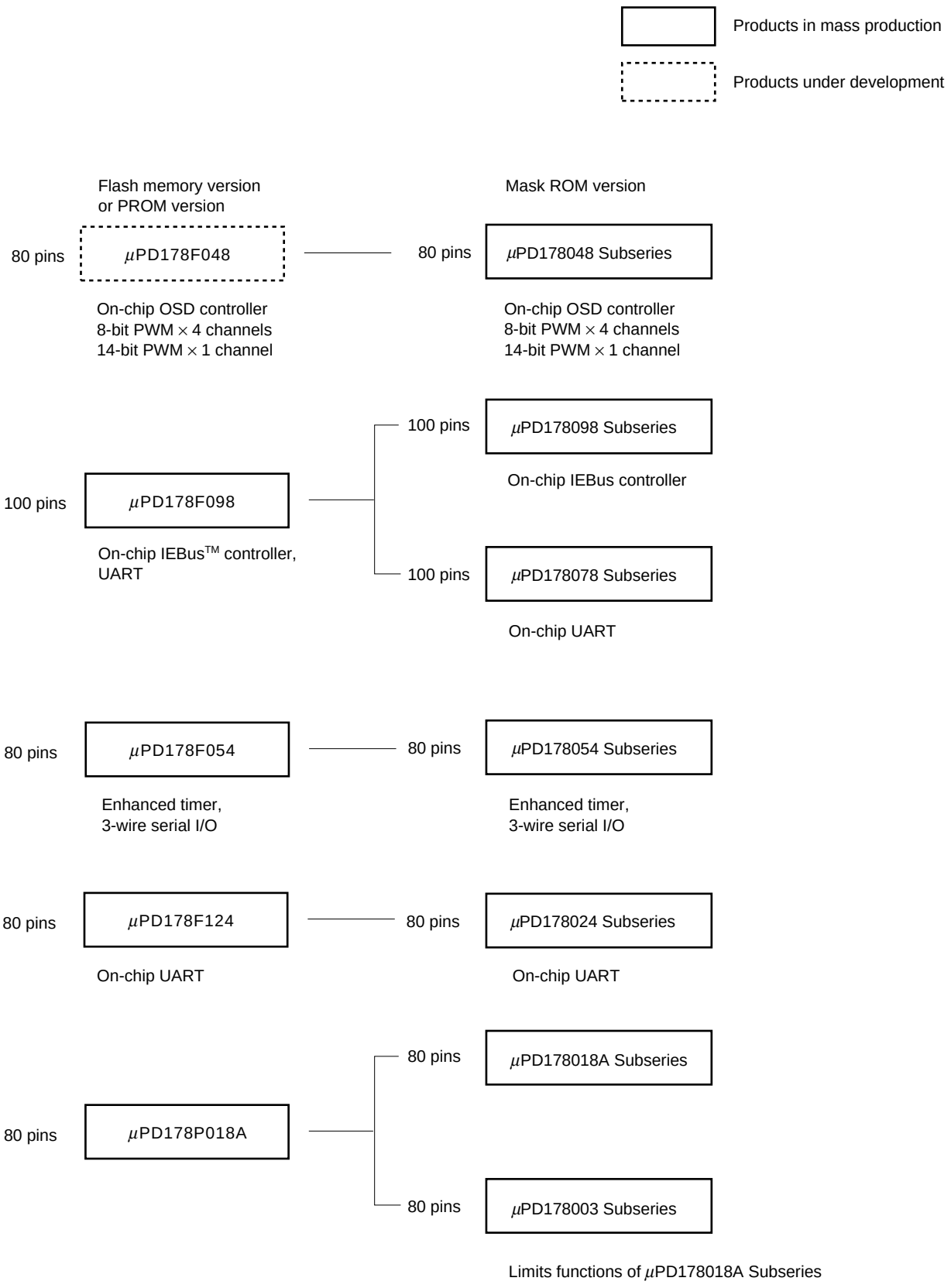
Remark ( ): μPD178F054 only

Pin Name

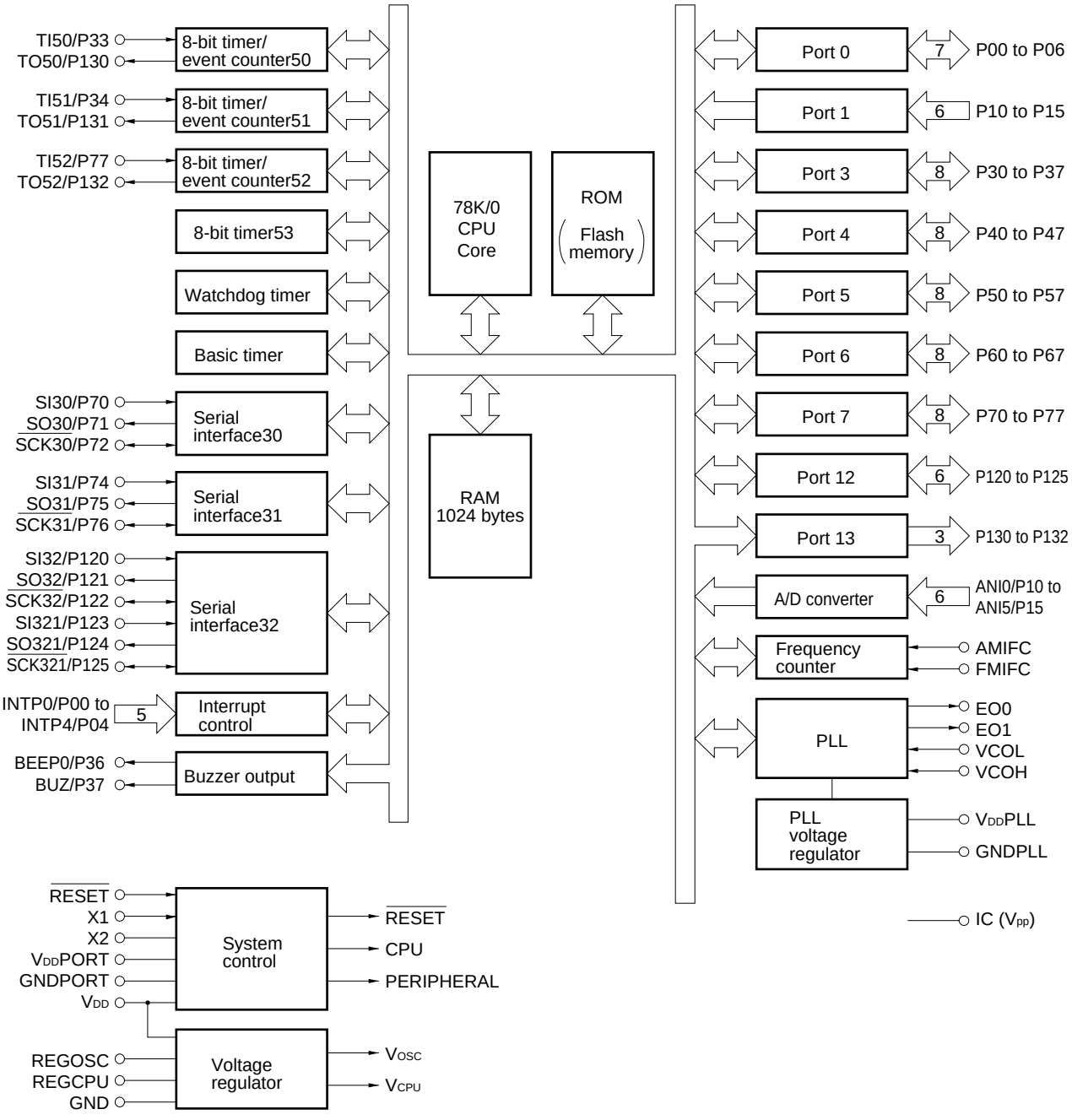
AMIFC:	AM intermediate frequency counter input	P130 to P132:	Port 13
ANI0 to ANI5:	A/D converter input	REGCPU:	Regulator for CPU power supply
BEEP0, BUZ:	Buzzer output	REGOSC:	Regulator for oscillator
EO0, EO1:	Error out output	$\overline{\text{RESET}}$:	Reset input
FMIFC:	FM intermediate frequency counter input	$\overline{\text{SCK30}}$, $\overline{\text{SCK31}}$,:	Serial (SIO3) clock input/output
GND:	Ground	$\overline{\text{SCK32}}$, $\overline{\text{SCK321}}$	
GNDPLL:	PLL ground	SI30, SI31, SI32,:	Serial (SIO3) data input
GNDPORT:	Port ground	SI321	
IC:	Internally connected	SO30, SO31,:	Serial (SIO3) data output
INTP0 to INTP4:	Interrupt input	SO32, SO321	
P00 to P06:	Port 0	TI50 to TI52:	8-bit timer clock input
P10 to P15:	Port 1	TO50 to TO52:	8-bit timer output
P30 to P37:	Port 3	VCOL, VCOH:	Local oscillation input
P40 to P47:	Port 4	V_{DD}:	Power supply
P50 to P57:	Port 5	V_{DD}PLL:	PLL power supply
P60 to P67:	Port 6	V_{DD}PORT:	Port power supply
P70 to P77:	Port 7	V_{PP}**Note**:	Programming power supply
P120 to P125:	Port 12	X1, X2:	Crystal resonator

Note μPD178F054 only

★ 1.5 Development of 8-Bit DTS Series



1.6 Block Diagram



Remarks 1. The internal ROM capacity differs depending on the product.

2. ( ): μPD178F054

1.7 Functional Outline

Item		μPD178053	μPD178054	μPD178F054
Internal	ROM	24 KB (Mask ROM)	32 KB (Mask ROM)	32 KB (Flash memory)
	High-speed RAM	1024 bytes		
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)		
Minimum instruction execution time		0.45 μs/0.89 μs/1.78 μs/3.56 μs/7.11 μs (with crystal resonator of f_X = 4.5 MHz)		
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.		
I/O ports		Total: 62 pins • CMOS I/O: 53 pins • CMOS input: 6 pins • N-ch open-drain output: 3 pins		
A/D converter		8-bit resolution × 6 channels		
Serial interface		• 3-wire serial I/O mode: 2 channels • 3-wire serial I/O mode (on-chip time-division transfer): 1 channel		
Timer		• Basic timer (timer carry FF (10 Hz)): 1 channel • 8-bit timer/event counter: 4 channels • Watchdog timer: 1 channel		
Buzzer output		BEEP pin: 1 kHz, 1.5 kHz, 3 kHz, 4 kHz BUZ pin: 549 Hz, 1.10 kHz, 2.20 kHz, 4.39 kHz		
Vectored interrupt sources	Maskable	Internal : 11 External: 5		
	Non-maskable	Internal: 1		
	Software	1		
PLL frequency synthesizer	Division mode	2 types • Direct division mode (VCOL pin) • Pulse swallow mode (VCOL and VCOH pins)		
	Reference frequency	Seven types selectable in software (1, 3, 9, 10, 12.5, 25, 50 kHz)		
	Charge pump	Error out output: 2 pins		
	Phase comparator	Unlock detectable with program		
Frequency counter		Frequency measurement • AMIFC pin: For 450 kHz counting • FMIFC pin: For 450 kHz/10.7 MHz counting		

Reset		• Reset by $\overline{\text{RESET}}$ pin
		• Internal reset by watchdog timer
		• Reset by power-on clear circuit
		  • Detection of less than 4.5 V**Note** (reset does not occur, however)
		  • Detection of less than 3.5 V**Note** (during CPU operation)
		  • Detection of less than 2.2 V**Note** (in STOP mode)		

Supply voltage		• V_{DD} = 4.5 to 5.5 V (during CPU, PLL operation) • V_{DD} = 3.5 to 5.5 V (during CPU operation)		
Package		80-pin plastic QFP (14 × 14)		

Note For details, refer to **CHAPTER 16 RESET FUNCTION**.

CHAPTER 2 PIN FUNCTION

2.1 Pin Function List

(1) Port pins

Pin Name	I/O	Function	After Reset	Alternate Function
P00 to P04	I/O	Port 0 7-bit I/O port Input/output can be specified in 1-bit units.	Input	INTP0 to INTP4
P05, P06				—
P10 to P15	Input	Port 1 6-bit input port	Input	ANI0 to ANI5
P30 to P32	I/O	Port 3 8-bit I/O port. Input/output can be specified in 1-bit units.	Input	—
P33				TI50
P34				TI51
P35				—
P36				BEEP0
P37				BUZ
P40 to 47	I/O	Port 4 8-bit I/O port. Input/output can be specified in 1-bit units. An on-chip pull-up resistor can be specified by software. Interrupt function by key input is provided.	Input	—
P50 to P57	I/O	Port 5 8-bit I/O port Input/output can be specified in 1-bit units.	Input	—
P60 to P67	I/O	Port 6. 8-bit I/O port. Input/output can be specified in 1-bit units.	Input	—
P70	I/O	Port 7 8-bit I/O port Input/output can be specified in 1-bit units.	Input	SI30
P71				SO30
P72				$\overline{\text{SCK30}}$
P73				—
P74				SI31
P75				SO31
P76				$\overline{\text{SCK31}}$
P77				TI52
P120	I/O	Port 12 6-bit I/O port Input/output can be specified in 1-bit units.	Input	SI32
P121				SO32
P122				$\overline{\text{SCK32}}$
P123				SI321
P124				SO321
P125				$\overline{\text{SCK321}}$
P130	Output	Port 13 3-bit output port N-ch open-drain output port (12 V tolerance)	Low-level output	TO50
P131				TO51
P132				TO52

(2) Pins other than port pins

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0 to INTP4	Input	External maskable interrupt input whose valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	Input	P00 to P04
SI30	Input	Serial data input to serial interface.	Input	P70
S131				P74
S132				P120
SI321				P123
SO30	Output	Serial data output from serial interface.	Input	P71
SO31				P75
SO32				P121
SO321				P124
$\overline{SCK30}$	I/O	Serial clock input/output to/from serial interface.	Input	P72
$\overline{SCK31}$				P76
$\overline{SCK32}$				P122
$\overline{SCK321}$				P125
TI50	Input	External count clock input to 8-bit timer 50	Input	P33
TI51		External count clock input to 8-bit timer 51		P34
TI52		External count clock input to 8-bit timer 52		P77
TO50	Output	8-bit timer 50 output	Low-level output	P130
TO51		8-bit timer 51 output		P131
TO52		8-bit timer 52 output		P132
BEEP0	Output	Buzzer output	Input	P36
BUZ				P37
ANI0 to ANI5	Input	Analog input to A/D converter	Input	P10 to P15
EO0, EO1	Output	Error out output from charge pump of PLL frequency synthesizer	–	–
VCOL	Input	Inputs local oscillation frequency of PLL (in HF and MF modes)	–	–
VCOH		Inputs local oscillation frequency of PLL (in VHF mode)		
AMIFC	Input	Input to AM intermediate frequency counter	Input	–
FMIFC		Input to FM or AM intermediate frequency counter		
$\overline{RESET}$	Input	System reset input	–	–
X1	Input	Connection of crystal resonator for system clock oscillation.	–	–
X2	–		–	–
REGOSC	–	Regulator for oscillator. Connect this pin to GND via 0.1 μF capacitor.	–	–
REGCPU	–	Regulator for CPU power supply. Connect this pin to GND via 0.1 μF capacitor.	–	–
V_{DD}	–	Positive power supply	–	–
GND	–	Ground	–	–
V_{DD}PORT	–	Port power supply	–	–
GNDPORT	–	Port ground	–	–
V_{DD}PLL**Note 1**	–	PLL positive power supply	–	–
GNDPLL**Note 1**	–	PLL ground	–	–
IC	–	Internally connected. Directly connect this pin to GND.	–	–
V_{PP}**Note 2**	–	Pin to apply high voltage at program writing/verifying. Directly connect this pin to GND in normal operating mode.	–	–

Notes 1. Connect a capacitor of about 1000 pF between the V_{DD}PLL and GNDPLL pins.

2. μPD178F054 only.

2.2 Description of Pin Functions

2.2.1 P00 to P06 (Port 0)

P00 to P06 constitute a 7-bit I/O port. In addition to I/O port pins, P00 to P06 also function as external interrupt inputs. The following operating modes can be specified in 1-bit units.

(1) Port mode

These pins function as a 7-bit I/O port for which input or output can be specified in 1-bit units using port mode register 0 (PM0).

(2) Control mode

These pins function as external interrupt input pins (INTP0 to INTP4).

These external interrupt input pins can specify valid edges (rising edge, falling edge, and both rising and falling edges).

2.2.2 P10 to P15 (Port 1)

P10 to P15 constitute a 6-bit input port. In addition to input port pins, P10 to P15 function as A/D converter analog inputs.

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

(1) Port mode

These pins function as a 6-bit input port.

(2) Control mode

These pins function as A/D converter analog input pins (ANI0 to ANI5).

2.2.3 P30 to P37 (Port 3)

P30 to P37 constitute an 8-bit I/O port . In addition to I/O port pins, P30 to P37 function as timer inputs and buzzer outputs.

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

(1) Port mode

These pins function as an 8-bit I/O port for which input or output can be specified in 1-bit units using port mode register 3 (PM3).

(2) Control mode

These pins function as timer inputs (TI50, T51) and buzzer outputs (BEEP0, BUZ).

(a) TI50, TI51

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

(b) BEEP0, BUZ

Buzzer output pins.

2.2.4 P40 to P47 (Port 4)

P40 to P47 constitute an 8-bit I/O port.

These pins can be specified as input or output in 1-bit units using port mode register 4 (PM4).

On-chip pull-up resistors can be specified by pull-up resistor option register 4 (PU4). An interrupt function via key input is also provided.

2.2.5 P50 to P57 (Port 5)

P50 to P57 constitute an 8-bit I/O port.

These pins can be specified as input or output in 1-bit units using port mode register 5 (PM5).

2.2.6 P60 to P67 (Port 6)

P60 to P67 constitute an 8-bit port.

These pins can be specified as input or output in 1-bit units using port mode register 6 (PM6).

2.2.7 P70 to P77 (Port 7)

P70 to P77 pins constitute an 8-bit I/O port. In addition to port pins, P70 to P77 also function as serial interface data I/O, clock I/O, and a timer input.

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

(1) Port mode

These pins function as an 8-bit I/O port for which input or output can be specified in 1-bit units using port mode register 7 (PM7).

(2) Control mode

These pins function as serial interface data I/O, clock I/O, and timer input pins.

(a) SI30, SO30, SI31, SO31

Serial data I/O pins of the serial interface.

(b) $\overline{\text{SCK30}}$, $\overline{\text{SCK31}}$

Serial clock I/O pins of the serial interface.

(c) TI52

External clock input pin to 8-bit timer/event counter.

2.2.8 P120 to P125 (Port 12)

P120 to P125 constitute a 6-bit I/O port. In addition to I/O port pins, P120 to P125 also function as serial interface data I/O and clock I/O.

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

(1) Port mode

These pins function as an 8-bit I/O port for which input or output can be specified in 1-bit units using port mode register 7 (PM7).

(2) Control mode

These pins function as serial interface data I/O and clock I/O pins.

(a) SI32, SO32, SI321, SO321

Serial data I/O pins of the serial interface.

(b) $\overline{\text{SCK32}}$, $\overline{\text{SCK321}}$

Serial clock I/O pins of the serial interface.

2.2.9 P130 to P132 (Port 13)

P130 to P132 constitute a 3-bit N-ch open-drain output port with a 12 V tolerance. In addition to output port pins, P130 to P132 also function as timer outputs.

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

(1) Port mode

These pins function as a 3-bit output port.

(2) Control mode

These pins function as output pins for the 8-bit timer/event counter.

TO50, TO51, TO52

These pins are output pins for the 8-bit timer/event counter.

2.2.10 EO0, EO1

These are the output pins of the charge pump of the PLL frequency synthesizer.

They output the result of phase comparison between the frequency divided by the programmable divider of the local oscillation input (VCOL and VCOH pins) and the reference frequency.

2.2.11 VCOL, VCOH

These pins input the local oscillation frequency (VCO) of the PLL.

Because signals are input to these pins via an AC amplifier, cut the DC component of the input signals using a capacitor.

- VCOL
  - HF, MF input
  - This pin becomes active when the HF or MF mode is selected by software; otherwise, the pin is in the status set by bit 2 (VCOLDMD) of the PLL mode select register (PLLMD). If VCOLDMD is reset to 0 (to connect a pull-down resistor), however, the VCOL pin does not become active even if the HF or MF mode is selected. In this case, set VCOLDMD to 1 (high-impedance state).

- VCOH
  - VHF input
  - This pin becomes active when the FM mode is selected by software; otherwise the pin is in the status set by bit 3 (VCOHDMD) of the PLL mode select register (PLLMD). If VCOHDMD is reset to 0 (to connect a pull-down resistor), however, the VCOL pin does not become active even if the FM mode is selected. In this case, set VCOHDMD to 1 (high-impedance state).

2.2.12 AMIFC

Input pin of the AM intermediate frequency counter.

2.2.13 FMIFC

Input pin of the FM intermediate frequency counter or AM intermediate frequency counter.

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

Low-level active system reset input pin.

2.2.15 X1, X2

Crystal resonator connection pins for system clock oscillation.

2.2.16 REGOSC

Regulator pin for oscillator. Connect to GND via a 0.1 μF capacitor.

2.2.17 REGCPU

Regulator pin for CPU power supply. Connect to GND via a 0.1 μF capacitor.

2.2.18 V_{DD}

Positive power supply pin.

2.2.19 GND

Ground potential pin.

2.2.20 V_{DD}PORT

Positive power supply pin for port.

2.2.21 GNDPORT

Ground potential pin for port.

2.2.22 V_{DD}PLL

Positive power supply pin for PLL.

2.2.23 GNDPLL

Ground potential pin for PLL.

2.2.24 V_{PP} (μPD178F054 only)

This pin applies a high voltage when the flash memory programming mode is set or when a program is written or verified.

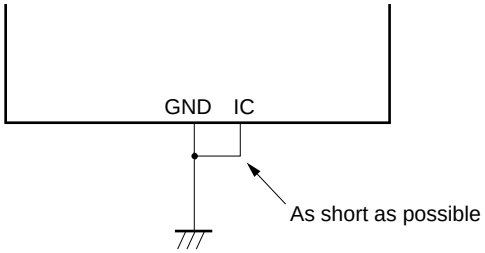
In the normal operation mode, directly connect this pin to GND.

2.2.25 IC (Mask ROM version only)

The IC (Internally Connected) pin is provided to set the test mode to check the μPD178054 Subseries at delivery. Connect it directly to the GND pin with the shortest possible wire in the normal operating mode.

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

○ Connect IC pin to GND pin directly.



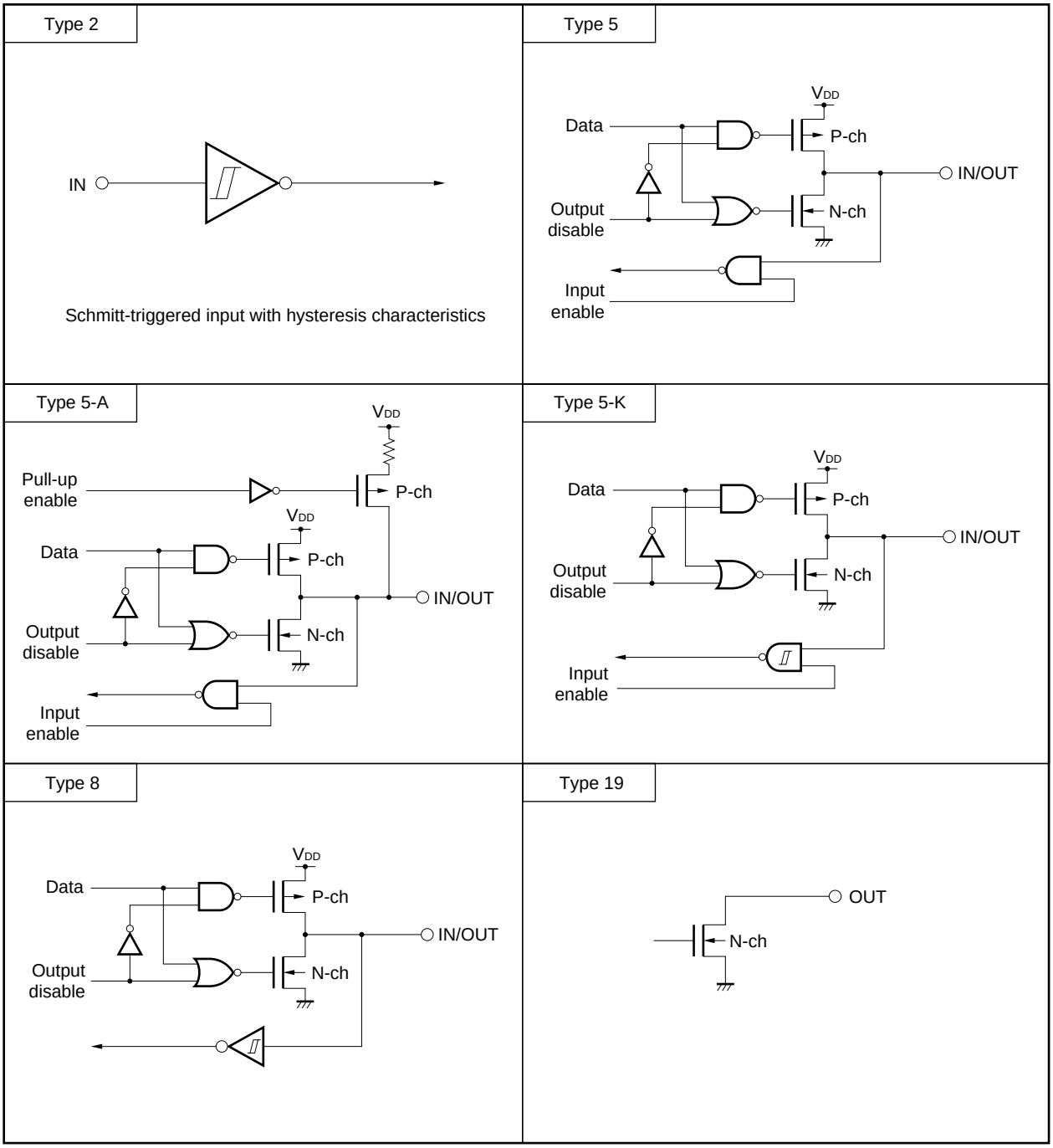
2.3 Pin I/O Circuits and Recommended Connections of Unused Pins

Table 2-1 shows the types of the I/O circuits of the respective pins and the recommended connections of the pins when they are not used. For the configuration of the I/O circuit of each pin, refer to Figure 2-1.

Table 2-1. Pin I/O Circuit Type and Recommended Connections of Unused Pins

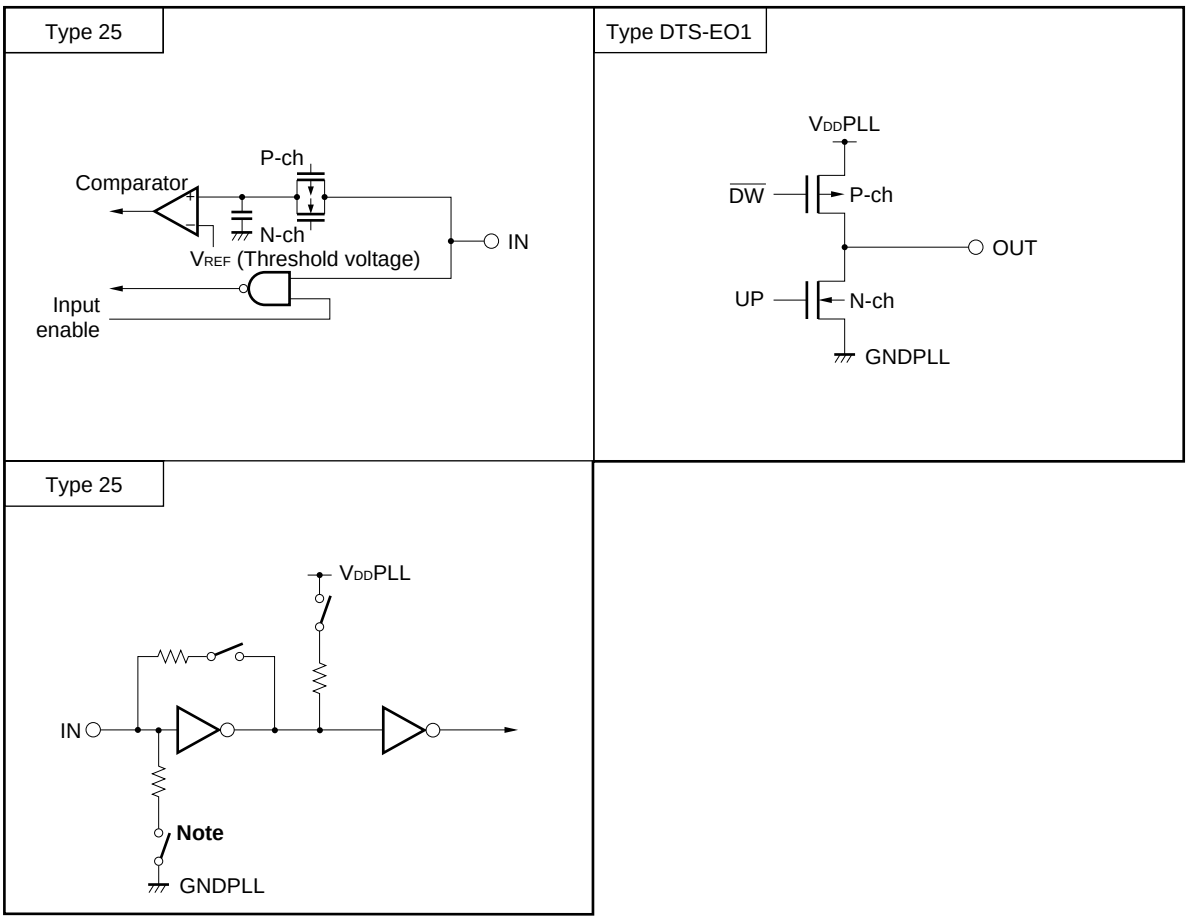
Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pin
P00/INTP0 to P04/INTP4	8	I/O	Input: Connect to V_{DD}, V_{DD}PORT, GND, or GNDPORT via a resistor. Output: Leave open.
P05, P06			
P10/ANI0 to P15/ANI5	25	Input	Connect to V_{DD}, V_{DD}PORT, GND, or GNDPORT.
P30 to P32	5	I/O	Input: Connect to V_{DD}, V_{DD}PORT, GND, or GNDPORT via a resistor. Output: Leave open.
P33/TI50	5-K		
P34/TI51			
P35	5		
P36/BEEP0			
P37/BUZ			
P40 to P47	5-A		
P50 to P57	5		
P60 to P67			
P70/SI30	5-K		
P71/SO30	5		
P72/$\overline{\text{SCK30}}$	5-K		
P73	5		
P74/SI31	5-K		
P75/SO31	5		
P76/$\overline{\text{SCK31}}$	5-K		
P77/TI52			
P120/SI32			
P121/SO32	5		
P122/$\overline{\text{SCK32}}$	5-K		
P123/SI321			
P124/SO321	5		
P125/$\overline{\text{SCK321}}$	5-K		
P130/TO50	19	Output	Leave open.
P131/TO51			
P132/TO52			
EO0, EO1	DTS-EO1		
VCOL, VCOH	DTS-AMP	Input	Disable PLL in software and select pull-down.
AMIFC, FMIFC			Set these pins in general-purpose input port mode by software and connect each of them to V_{DD}, V_{DD}PORT, GND, or GNDPORT via a resistor.
REGOSC, REGCPU	–	–	Connect these pins to GND via 0.1 μF capacitor.
$\overline{\text{RESET}}$	2	Input	–
V_{DD}PLL	–	–	Connect to V_{DD}.
GNDPLL			Directly connect to GND or GNDPORT.
IC (Mask ROM version)			
V_{PP} (μPD178F054)			

Figure 2-1. Pin I/O Circuits (1/2)



Remark V_{DD} and GND are the positive power supply and ground pins for all port pins. Read V_{DD} and GND as V_{DD}PORT and GNDPORT.

Figure 2-1. Pin I/O Circuits (2/2)



Note This switch is selectable by software only for the VCOL and VCOH pins.

Remark V_{DD} and GND are the positive power supply and ground pins for all port pins. Read V_{DD} and GND as V_{DD}PORT and GNDPORT.

CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

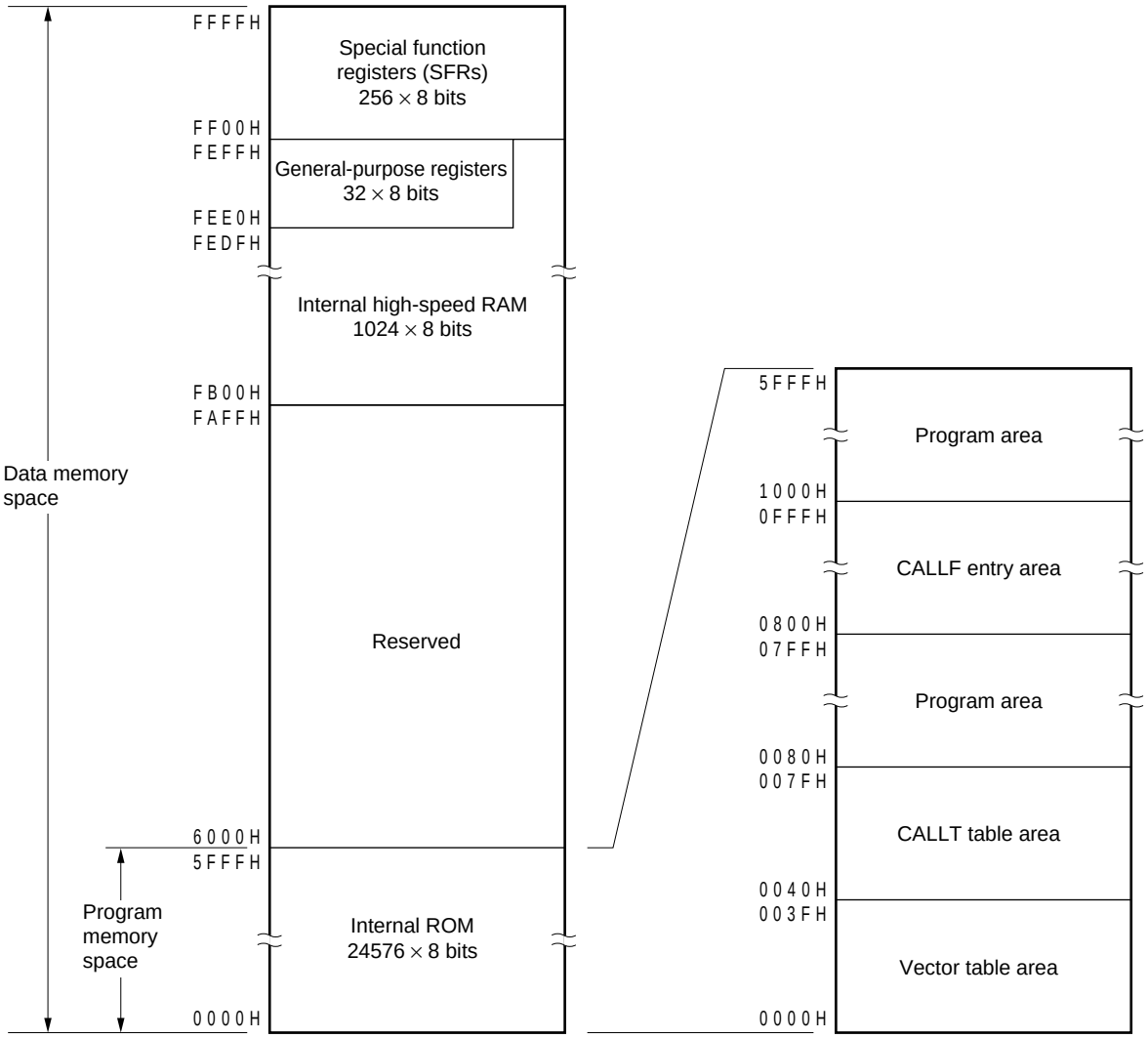
The initial value of the memory size switching register (IMS) is CFH. The following values must be set to the registers of each model.

Part Number	IMS
μPD178053	C6H
μPD178054	C8H
μPD178F054	Value equivalent to mask ROM version

(1) μPD178053

Set the value of the memory size switching register (IMS) to C6H. The initial value is CFH.

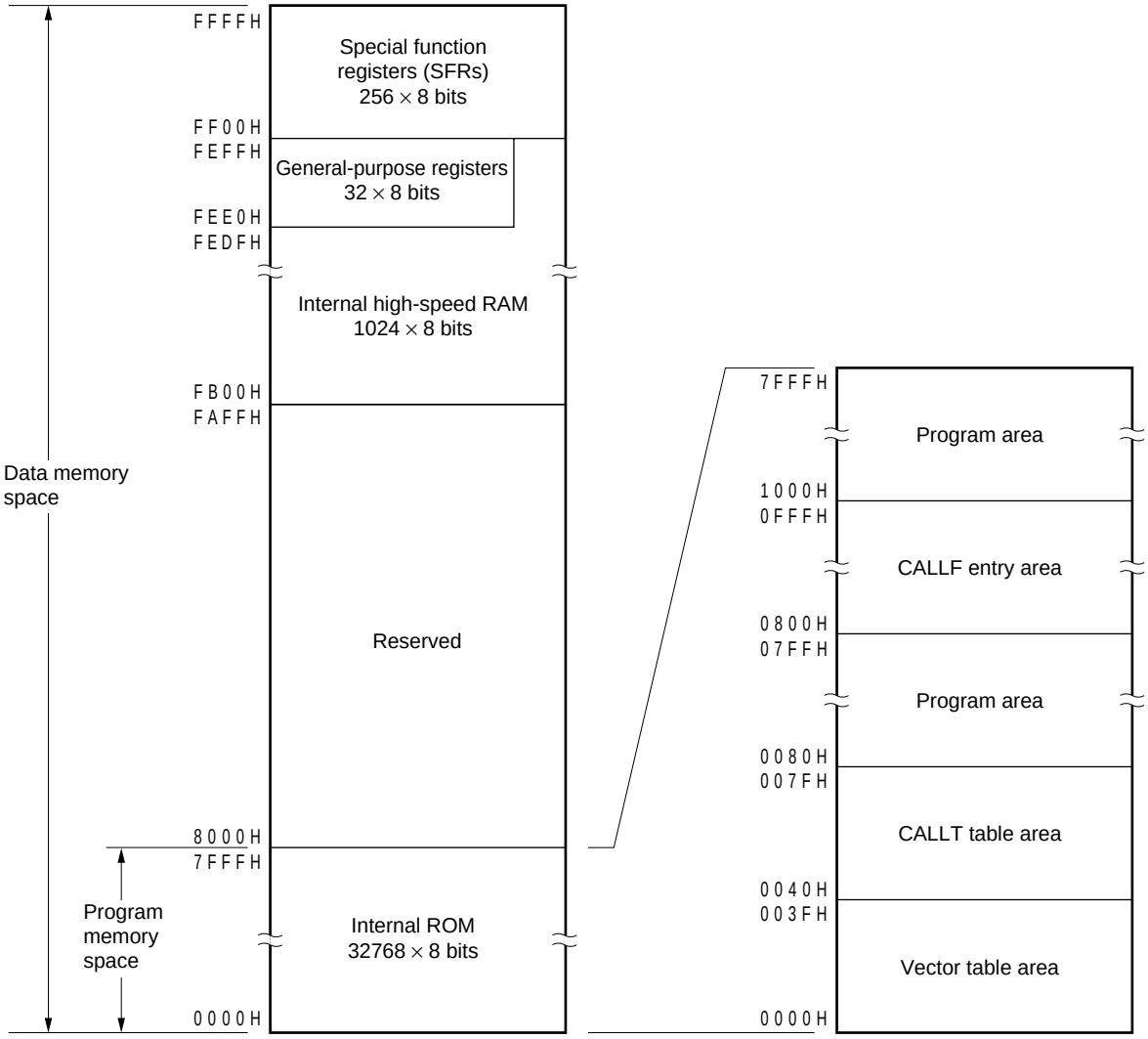
Figure 3-1. Memory Map of μPD178053



(2) μPD178054

Set the value of the memory size switching register (IMS) to C8H. The initial value is CFH.

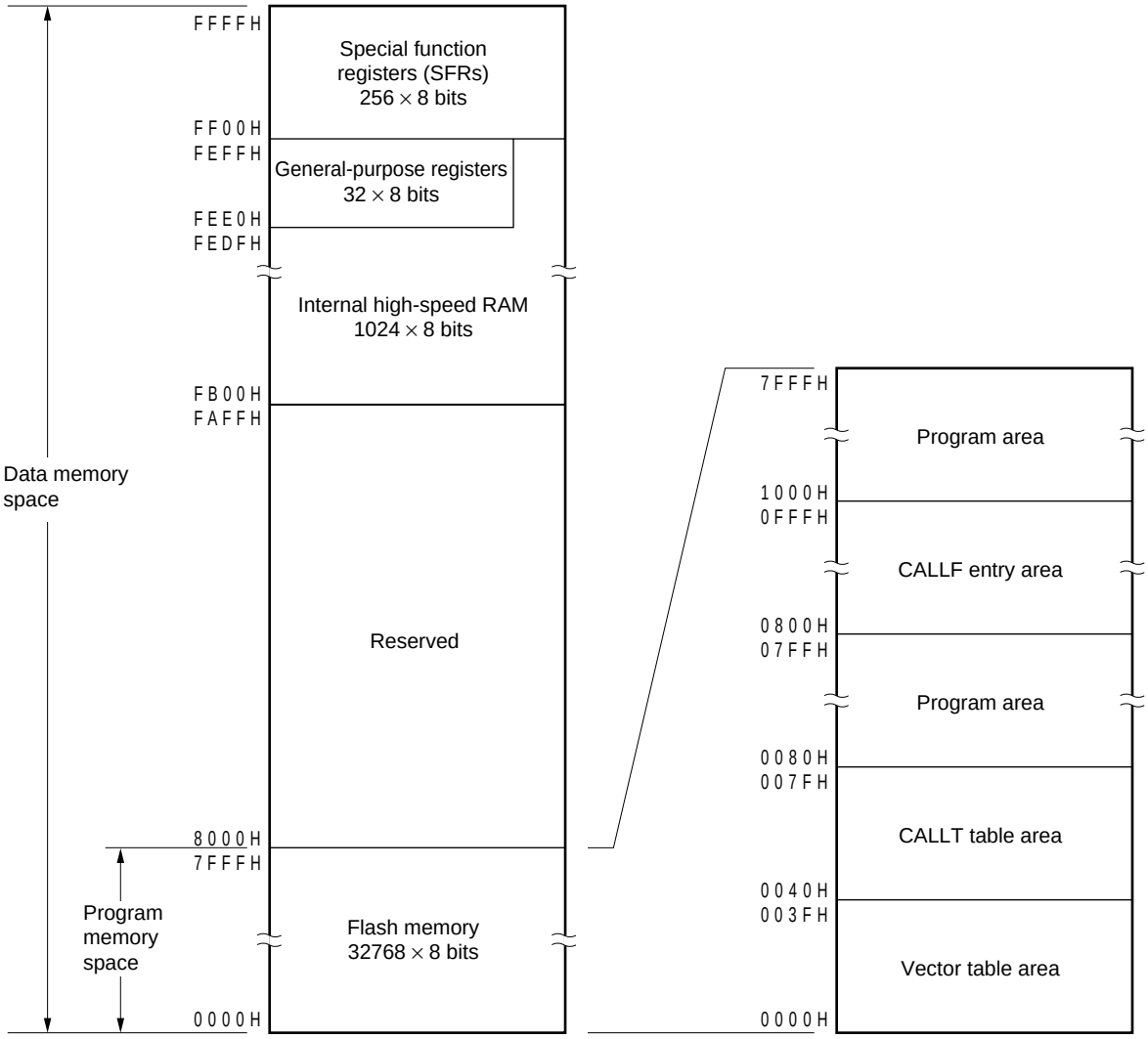
Figure 3-2. Memory Map of μPD178054



(3) μPD178F054

Set the value of the memory size switching register (IMS) to the value corresponding to that of the mask ROM versions. The initial value is CFH.

Figure 3-3. Memory Map of μPD178F054



3.1.1 Internal program memory space

Programs and table data are stored in internal program memory space, and are usually addressed by the program counter (PC).

The μPD178054 Subseries has internal ROM (or flash memory) as shown in the following table.

Table 3-1. Internal Memory Capacities

Part Number	Structure	Capacity
μPD178053	Mask ROM	24576 × 8 bits (0000F to 5FFFH)
μPD178054		32768 × 8 bits (0000H to 7FFFH)
μPD178F054	Flash memory	

The following areas are assigned to the internal program memory space.

(1) Vector table area

The 64-byte area 0000H to 003FH is reserved as a vector table area. The reset input and program start addresses for branch upon generation of each interrupt request are stored in the vector table area. Of the 16-bit address, the lower 8 bits are stored at even addresses and the higher 8 bits are stored at odd addresses.

Table 3-2. Vector Table

Vector Table Address	Interrupt Request
0004H	INTWDT
0006H	INTP0
0008H	INTP1
000AH	INTP2
000CH	INTP3
000EH	INTP4
0010H	INTKY
0012H	INTCSI31
0014H	INTBTM0
0016H	INTAD3
0018H	INTCSI32
001AH	INTCSI30
001CH	INTTM50
001EH	INTTM51
0020H	INTTM52
0022H	INTTM53
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 μPD178054 Subseries products incorporate the following RAMs.

(1) Internal high-speed RAM

The μPD178053, 178054, and 178F054 have a RAM structure of 1024 $\times$ 8 bits.

In this area, four banks of general-purpose registers, each bank consisting of eight 8-bit registers, are allocated in the 32-byte area FEE0H to FEFFH.

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

3.1.3 Special Function Register (SFR) area

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

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, refer to **3.3 Instruction Address Addressing**).

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the μPD178054 Subseries, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFR) and general-purpose registers are available for use. Data memory addressing is illustrated in Figures 3-4 to 3-6. For the details of each addressing mode, refer to **3.4 Operand Address Addressing**.

Figure 3-4. Data Memory Addressing of μPD178053

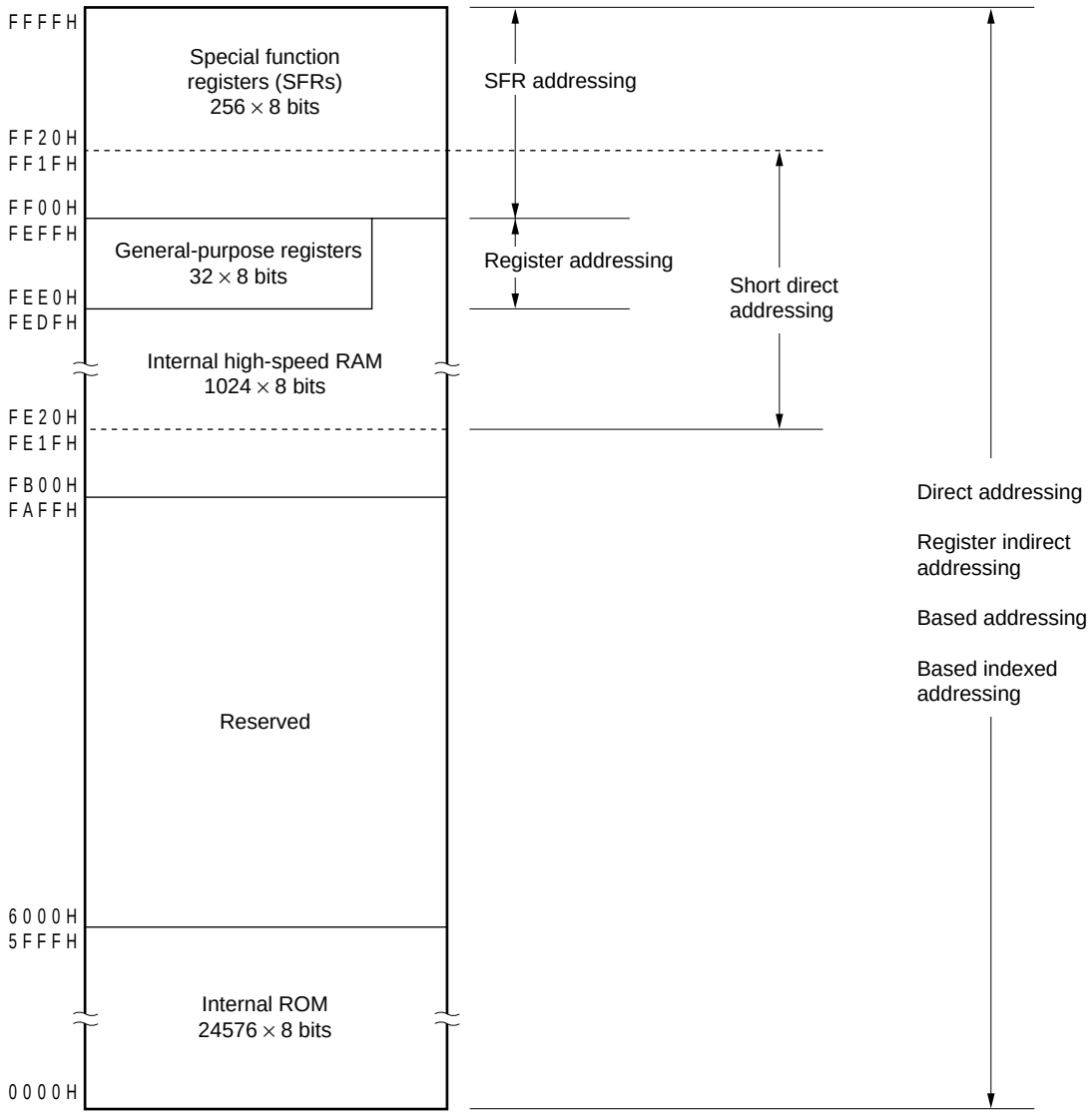


Figure 3-5. Data Memory Addressing of μPD178054

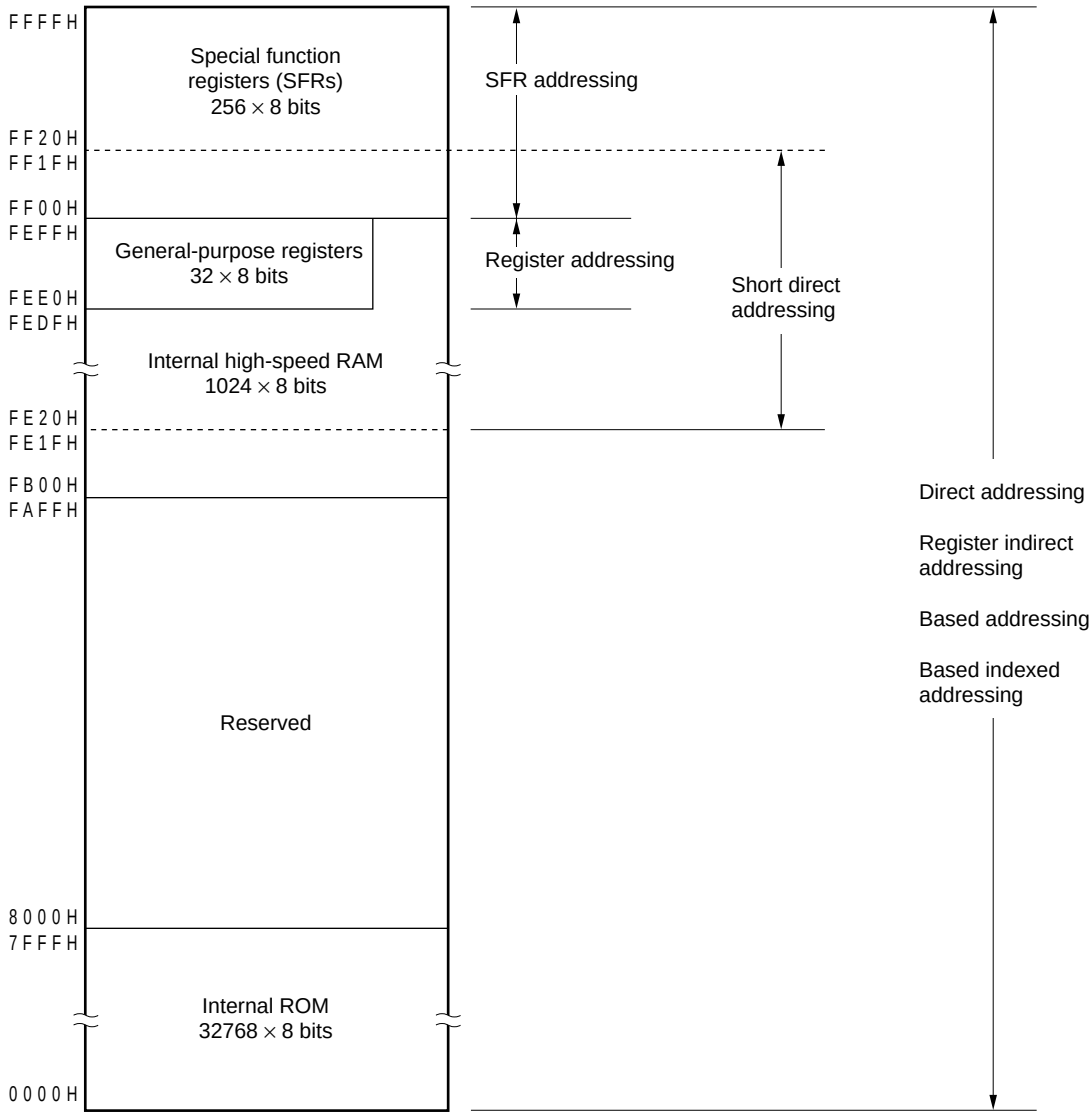
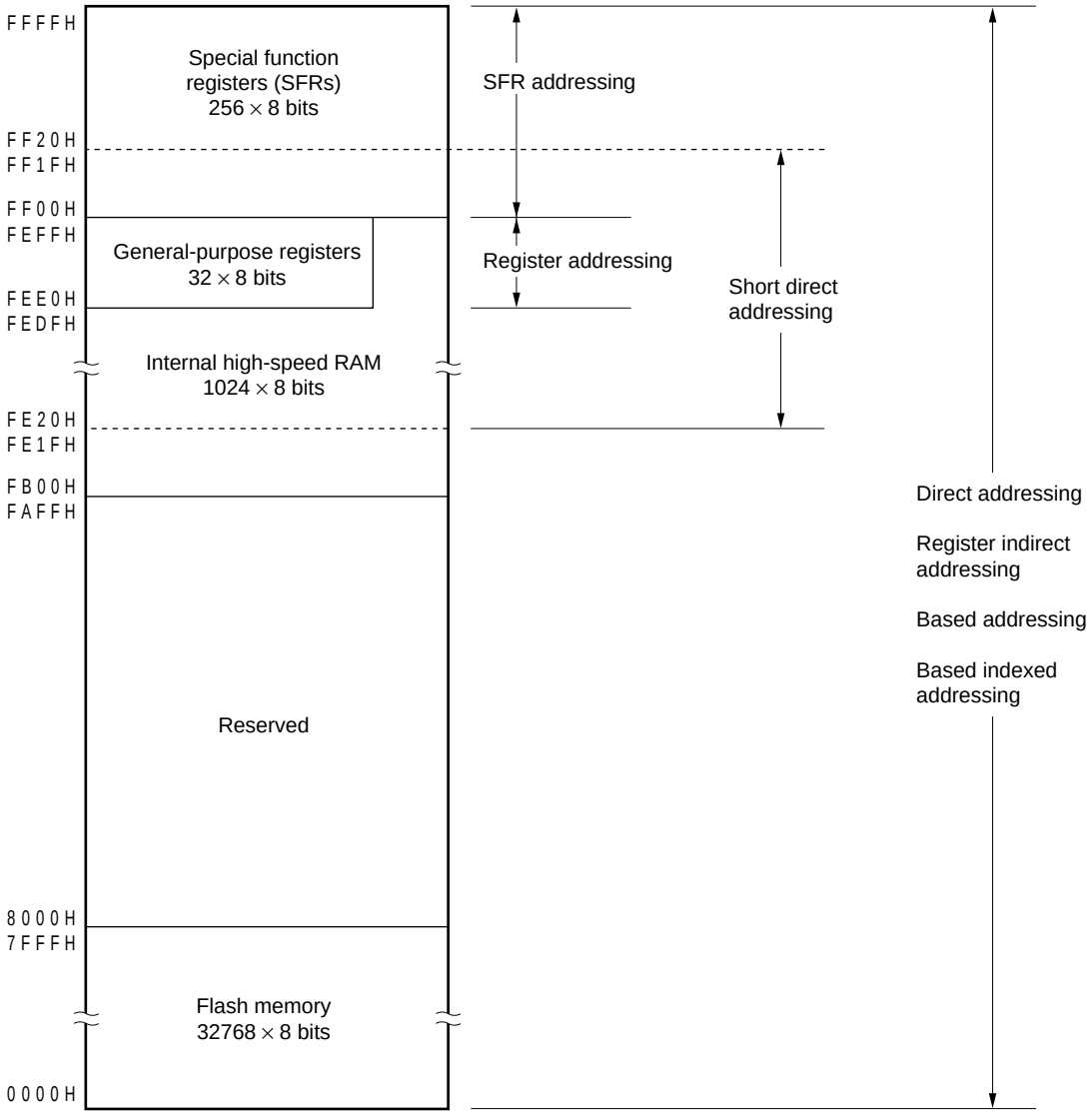


Figure 3-6. Data Memory Addressing of μPD178F054



3.2 Processor Registers

The μPD178054 Subseries units incorporate the following processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

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

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

Reset input sets the reset vector table values at addresses 0000H and 0001H to the program counter.

Figure 3-7. Configuration of Program Counter

	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 restored upon execution of the RETB, RETI and POP PSW instructions.

Reset input sets the PSW to 02H.

Figure 3-8. Configuration of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY

(a) Interrupt enable flag (IE)

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

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

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

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

(b) Zero flag (Z)

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

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

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

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

(d) Auxiliary carry flag (AC)

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

(e) In-service priority flag (ISP)

This flag manages the priority of acknowledgeable maskable vectored interrupts.

When ISP = 0, acknowledging the vectored interrupt requests to which a low priority is assigned by the priority specification flag registers (PR0L, PR0H) (refer to **12.3 (3) Priority specification flag registers (PR0L, PR0H**) is disabled. Whether an interrupt request is actually accepted depends on the status of the interrupt enable flag (IE).

(f) Carry flag (CY)

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

(3) Stack pointer (SP)

This is a 16-bit register to hold the start address of the memory stack area. Only the internal high-speed RAM area (FB00H to FEFFH for μPD178053, 178054, and 178F054) can be set as the stack area.

Figure 3-9. Configuration of Stack Pointer

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

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

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

Caution Since 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

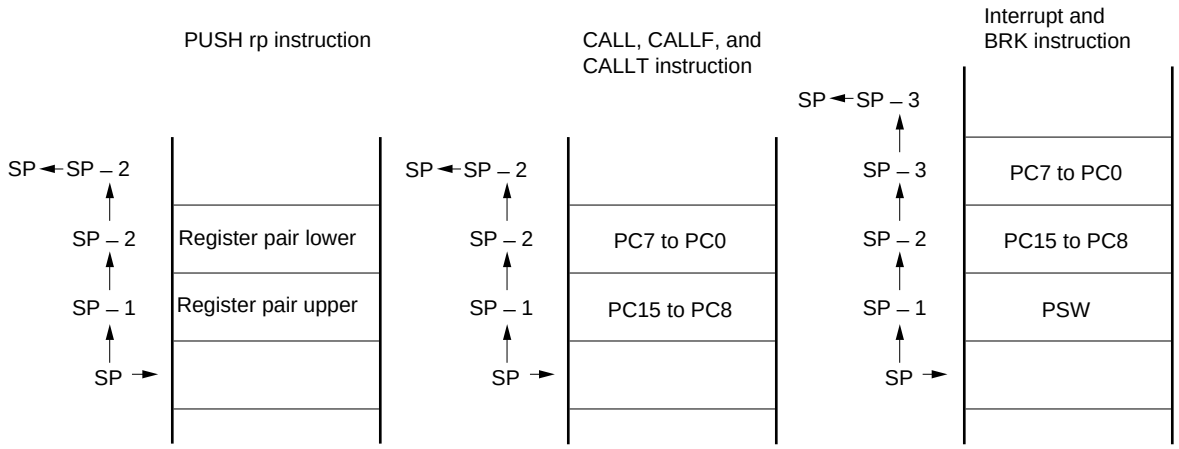
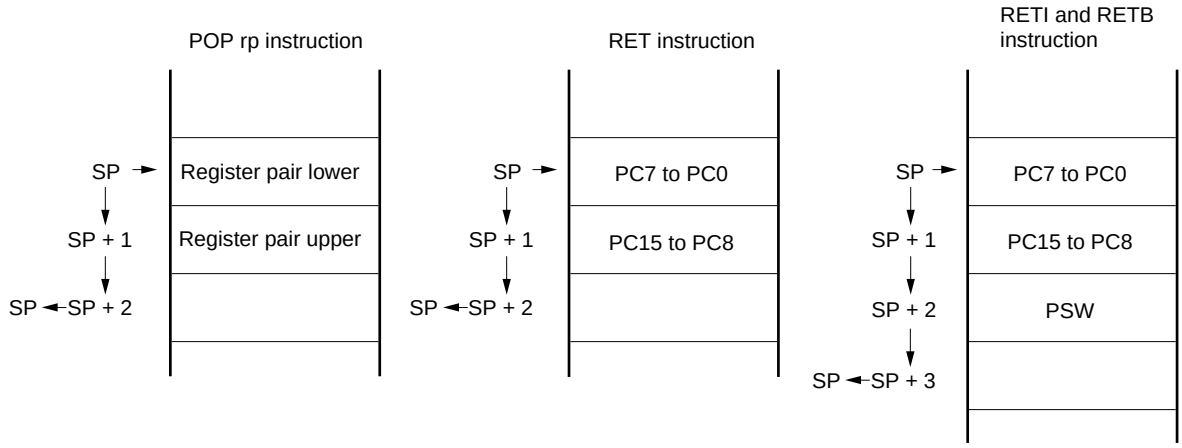


Figure 3-11. Data to Be Restored from Stack Memory



3.2.2 General-purpose registers

The general-purpose registers are mapped at particular address FEE0H to FEFFH in the data memory. They consist of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

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

They can be written with 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).

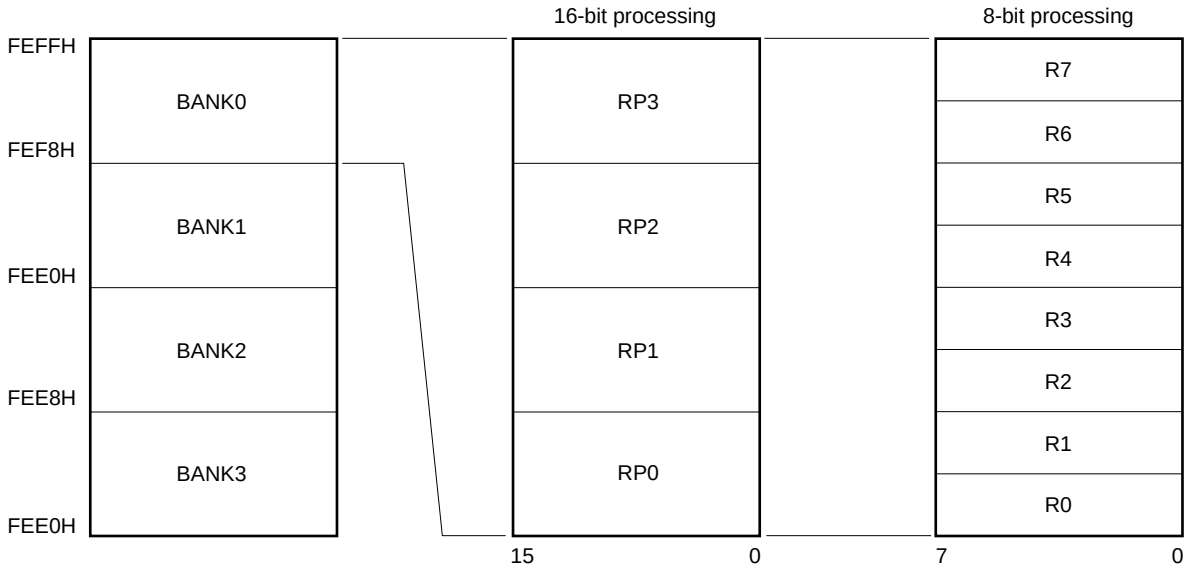
Register banks to be used for instruction execution are set with the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupt for each bank.

Table 3-3. Absolute Address of General-Purpose Registers

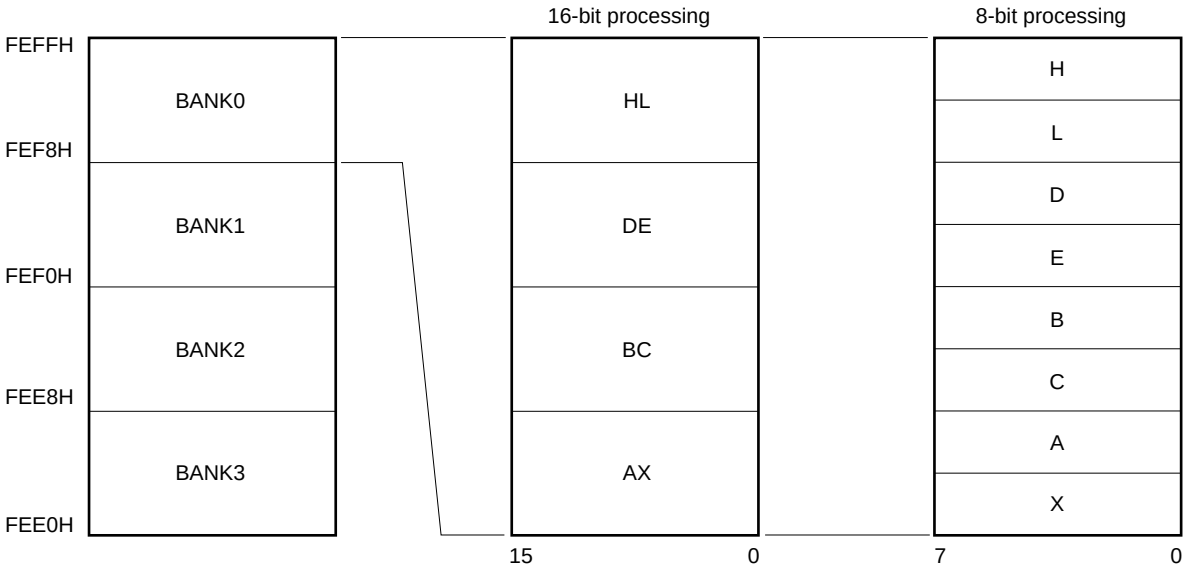
Bank	Register		Absolute Address	Bank	Register		Absolute Address
	Function Name	Absolute Name			Function Name	Absolute Name	
BANK0	H	R7	FEFFH	BANK2	H	R7	FEEFH
	L	R6	FEFEH		L	R6	FEEEH
	D	R5	FEFDH		D	R5	FEEDH
	E	R4	FEFCH		E	R4	FEECH
	B	R3	FEFBH		B	R3	FEEBH
	C	R2	FEFAH		C	R2	FEEAH
	A	R1	FEF9H		A	R1	FEE9H
	X	R0	FEF8H		X	R0	FEE8H
BANK1	H	R7	FEF7H	BANK3	H	R7	FEE7H
	L	R6	FEF6H		L	R6	FEE6H
	D	R5	FEF5H		D	R5	FEE5H
	E	R4	FEF4H		E	R4	FEE4H
	B	R3	FEF3H		B	R3	FEE3H
	C	R2	FEF2H		C	R2	FEE2H
	A	R1	FEF1H		A	R1	FEE1H
	X	R0	FEF0H		X	R0	FEE0H

Figure 3-12. Configuration of General-Purpose Register

(a) Absolute Name



(b) Function Name



3.2.3 Special Function Registers (SFR)

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

SFRs are allocated in the FF00H to FFFFH area.

SFRs are can be manipulated like general-purpose registers, using operation, transfer and bit manipulation instructions. The manipulatable bit units: 1, 8, and 16, depends on the special function register type.

The manipulatable bit units can be specified as follows.

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

Table 3-4 gives a list of special function registers. The meanings of items in the table are as follows.

- **Symbol**
  This is a symbol to indicate an address of the special function register.
  These symbols are reserved for the DF178054 and RA78K0, and defined by header file sfrbit.h for the CC78K0.
  They can be written as instruction operands when the RA78K0, ID78K0, or ID78K0-NS is used.
- **R/W**
  Indicates whether the corresponding special function register can be read or written.
  R/W: Read/write enable
  R: Read only
  R&Reset: Read only (reset to 0 when read)
  W: Write only
- **Bit units for manipulation**
  ○ indicates the manipulatable bit units: 1, 8, and 16. – indicates the bit units that cannot be manipulated.
- **After reset**
  Indicates each register status upon reset. The values of special function registers whose addresses are not shown in the table are undefined at reset.

Table 3-4. Special Function Registers (1/3)

Address	Special Function Register (SFR) Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0	R/W	○	○	—	00H
FF01H	Port 1	P1	R	○	○	—	
FF03H	Port 3	P3	R/W	○	○	—	
FF04H	Port 4	P4		○	○	—	
FF05H	Port 5	P5		○	○	—	
FF06H	Port 6	P6		○	○	—	
FF07H	Port 7	P7		○	○	—	
FF0CH	Port 12	P12		○	○	—	
FF0DH	Port 13	P13		○	○	—	
FF10H	A/D conversion result register 3 Note 1	ADCR3	—	—	—	—	—
FF11H			R	—	○	—	Undefined
FF12H	A/D converter mode register 3	ADM3	R/W	○	○	—	00H
FF13H	Analog input channel specification register 3	ADS3		—	○	—	
FF15H	Power-fail comparison threshold value register 3	PFT3		—	○	—	
FF16H	Power-fail comparison mode register 3	PFM3		○	○	—	
FF1BH	POC status register	POCS	R&Reset	—	○	—	Retained Note 2
FF20H	Port mode register 0	PM0	R/W	○	○	—	FFH
FF23H	Port mode register 3	PM3		○	○	—	
FF24H	Port mode register 4	PM4		○	○	—	
FF25H	Port mode register 5	PM5		○	○	—	
FF26H	Port mode register 6	PM6		○	○	—	
FF27H	Port mode register 7	PM7		○	○	—	
FF2CH	Port mode register 12	PM12		○	○	—	
FF34H	Pull-up resistor option register 4	PU4		○	○	—	00H
FF40H	Clock output select register	CKS		○	○	—	
FF41H	BEEP clock select register 0	BEEPCL0		○	○	—	
FF42H	Watchdog timer clock select register	WDCS		○	○	—	
FF48H	External interrupt rising edge enable register	EGP		○	○	—	
FF49H	External interrupt falling edge enable register	EGN		○	○	—	
FF69H	Serial port select register 32	SIO32SEL		○	○	—	
FF6AH	Serial I/O shift register 32	SIO32		—	○	—	Undefined
FF6BH	Serial operating mode register 32	CSIM32		○	○	—	00H
FF6CH	Serial I/O shift register 31	SIO31		—	○	—	Undefined
FF6DH	Serial operating mode register 31	CSIM31		○	○	—	00H

Notes 1. This register can be accessed only in 8-bit units. When ADCR3 is read, the value of FF11H is read.

2. The value of this register is 03H only at reset by power-on clear. This register is not reset by the $\overline{\text{RESET}}$ pin or watchdog timer.

Caution Do not access addresses to which no SFR is assigned.

Table 3-4. Special Function Registers (2/3)

Address	Special Function Register (SFR) Name		Symbol		R/W	Bit Units for Manipulation			After Reset
						1 Bit	8 Bits	16 Bits	
FF6EH	Serial I/O shift register 30		SIO30		R/W	—	○	—	Undefined
FF6FH	Serial operating mode register 30		CSIM30			○	○	—	00H
FF70H	8-bit compare register 52		CR52		W	—	○	—	Undefined
FF71H	8-bit compare register 53		CR53			—	○	—	
FF72H	8-bit timer counter 52		TM523	TM52	R	—	○	○	00H
FF73H	8-bit timer counter 53			TM53		—	○		
FF74H	Timer clock select register 52		TCL52		R/W	—	○	—	
FF75H	8-bit timer mode control register 52		TMC52			○	○	—	
FF77H	Timer clock select register 53		TLC53			—	○	—	
FF78H	8-bit timer mode control register 53		TMC53			○	○	—	
FF80H	8-bit compare register 50		CR50			—	○	—	Undefined
FF81H	8-bit compare register 51		CR51			—	○	—	
FF82H	8-bit timer counter 50		TM501	TM50	R	—	○	○	00H
FF83H	8-bit timer counter 51			TM51		—	○		
FF84H	Timer clock select register 50		TCL50		R/W	—	○	—	
FF85H	8-bit timer mode control register 50		TMC50			○	○	—	
FF87H	Timer clock select register 51		TCL51			—	○	—	
FF88H	8-bit timer mode control register 51		TMC51			○	○	—	
FFA0H	PLL mode select register		PLLMD			○	○	—	
FFA1H	PLL reference mode register		PLLRF			○	○	—	0FH
FFA2H	PLL unlock F/F judge register		PLLUL		R&Reset	○	○	—	Retained[Note 1]
FFA3H	PLL data transfer register		PLLNS		W	○	○	—	00H
FFA6H	PLL data registers	PLL data register L	PLLR	PLLRL	R/W	○	○	○	Undefined
FFA7H		PLL data register H		PLLRH		○	○		
FFA8H	PLL data register 0		PLLR0			○	○	—	
FFA9H	IF counter mode select register		IFCMD			○	○	—	00H
FFAAH	DTS system clock select register		DTSCK			○	○	—	00H[Note 2]
FFABH	IF counter gate judge register		IFCJG		R	○	○	—	00H
FFACH	IF counter control register		IFCCR		W	○	○	—	
FFAEH	IF counter register		IFCR	IFCRL	R	—	—	○	
FFAFH				IFCRH		—	—		

Notes 1. Undefined by power-on clear reset only.

2. Though the initial value of the DTS system clock select register (DTSCK) is 00H, be sure to set this register to 01H before using it.

Caution Do not access addresses to which no SFR is assigned.

Table 3-4. Special Function Registers (3/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Bit Units for Manipulation			After Reset
					1 Bit	8 Bits	16 Bits	
FFD0H | FFDFH	External access area[Note 1]			R/W	○	○	—	Undefined
FFE0H	Interrupt request flag register 0L	IF0	IF0L		○	○	○	00H
FFE1H	Interrupt request flag register 0H		IF0H		○	○		
FFE4H	Interrupt mask flag register 0L	MK0	MK0L		○	○	○	FFH
FFE5H	Interrupt mask flag register 0H		MK0H		○	○		
FFE8H	Priority specification flag register 0L	PR0	PR0L		○	○	○	
FFE9H	Priority specification flag register 0H		PR0H		○	○		
FFF0H	Memory size switching register	IMS			—	○	—	CFH[Note 2]
FFF4H	Internal expansion RAM size switching register	IXS			—	○	—	0CH[Note 3]
FFF9H	Watchdog timer mode register	WDTM			○	○	—	00H
★ FFFAH	Oscillation stabilization time switching register	OSTS			—	○	—	04H
FFFBH	Processor clock control register	PCC			○	○	—	

Notes 1. The external access area cannot be accessed by means of SFR addressing. Use direct addressing to access this area.

2. The initial value of the memory size switching register (IMS) is CFH. Set the values of these registers of each model as follows:

Part Number	IMS
μPD178053	C6H
μPD178054	C8H
μPD178F054	Value equivalent to mask ROM version

3. Do not assign a value other than the initial value to the internal expansion RAM size switching register (IXS).

Caution Do not access addresses to which no SFR is assigned.

3.3 Instruction Address Addressing

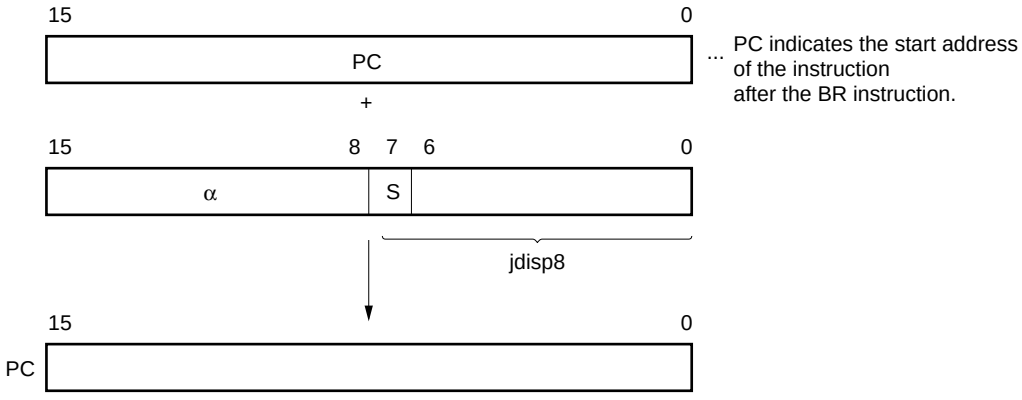
An instruction address is determined by program counter (PC) contents, and the contents are normally incremented (+1 for each byte) automatically according to the number of bytes of an instruction to be fetched each time another instruction is executed. When a branch instruction is executed, the branch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

3.3.1 Relative addressing

[Function]

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

[Illustration]



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

3.3.2 Immediate addressing

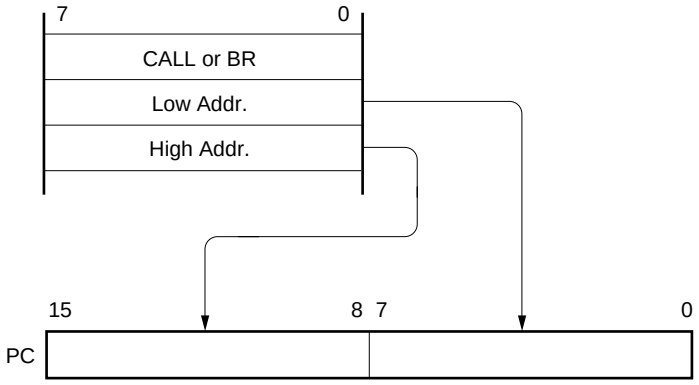
[Function]

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

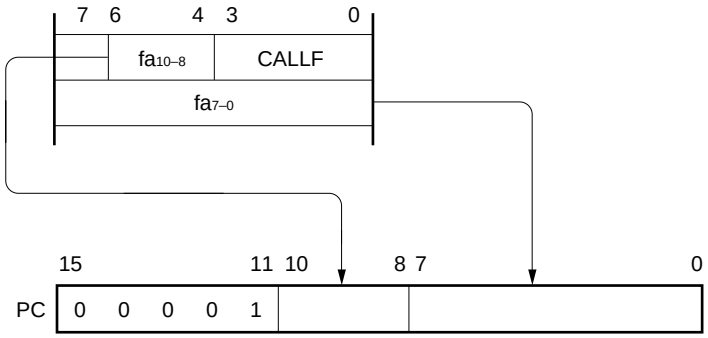
This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed. The CALL !addr16 and BR !add16 instructions can be used to branch to any location in the memory. The CALLF !addr11 instruction is used to branch to the area between 0800H through 0FFFH.

[Illustration]

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



In the case of CALLF !addr11 instruction



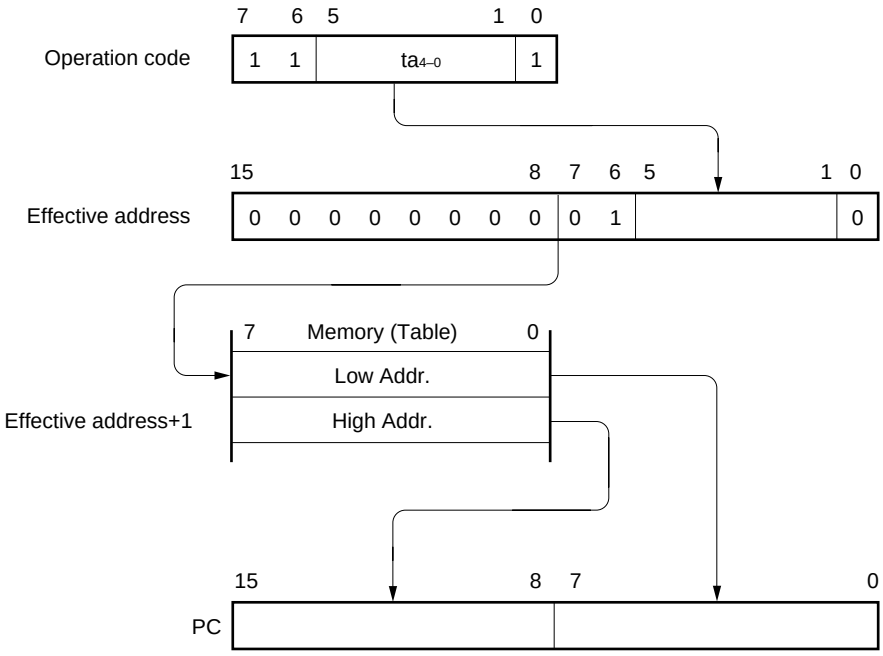
3.3.3 Table indirect addressing

[Function]

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

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

[Illustration]



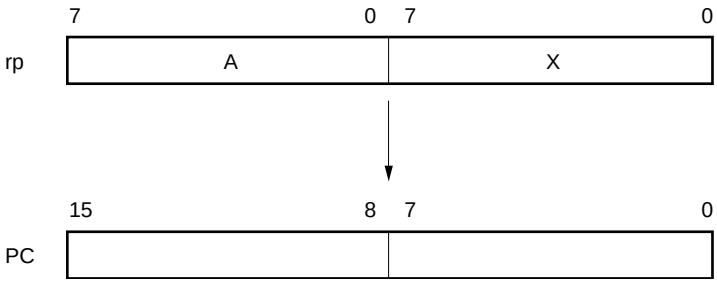
3.3.4 Register addressing

[Function]

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

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

[Illustration]



3.4 Operand Address Addressing

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

3.4.1 Implied addressing

[Function]

The registers that functions as an accumulator (A and AX) among the general-purpose registers are automatically addressed (implied). Of the μPD178054 Subseries instruction words, the following instructions employ implied addressing.

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

[Operand format]

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

[Example]

In the case of MULU X

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

3.4.2 Register addressing

[Function]

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

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

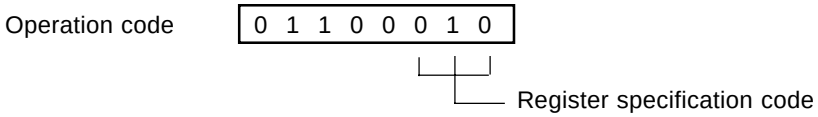
[Operand format]

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

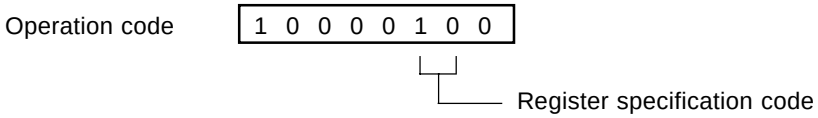
'r' and 'rp' can be written with function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) as well as absolute names (R0 to R7 and RP0 to RP3).

[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 with immediate data in an instruction word is directly addressed.

[Operand format]

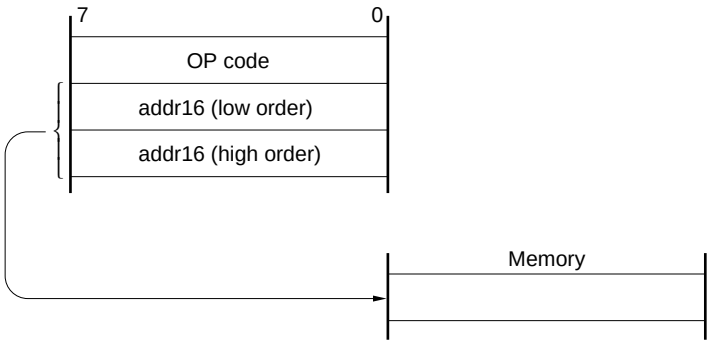
Symbol	Description
addr16	Label or 16-bit immediate data

[Example]

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

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

[Illustration]



3.4.4 Short direct addressing

[Function]

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

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

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

[Operand format]

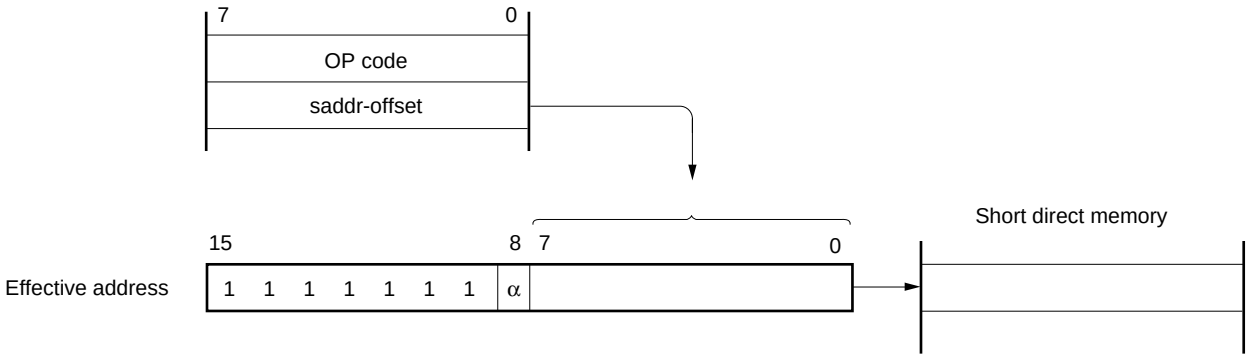
Symbol	Description
saddr	Label of FE20H to FF1FH immediate data
saddrp	Label of FE20H to FF1FH immediate data (even address only)

[Example]

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

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

[Illustration]



When 8-bit immediate data is 20H to FFH, $\alpha = 0$

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

3.4.5 Special Function Register (SFR) addressing

[Function]

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

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

[Operand format]

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

[Example]

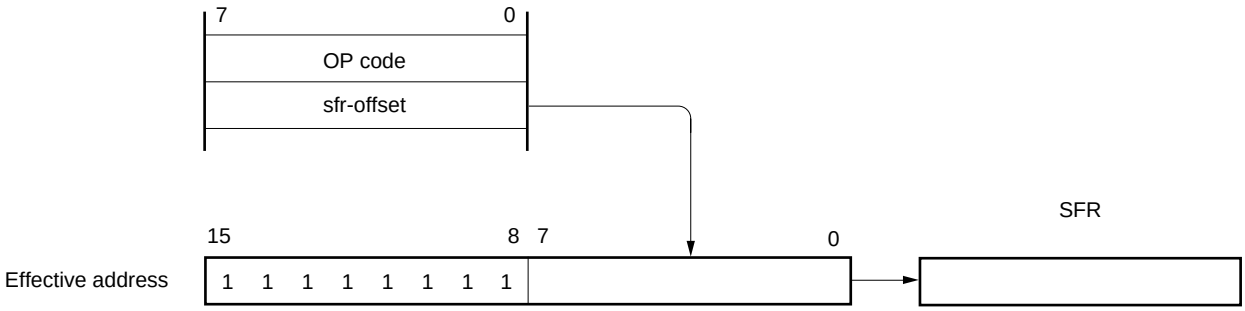
MOV PM0, A; when selecting PM0 (FF20H) as sfr

Operation code

1	1	1	1	0	1	1	0

0	0	1	0	0	0	0	0

[Illustration]



3.4.6 Register indirect addressing

[Function]

This addressing is used to address the memory to be manipulated by using the contents of the register pair specified by the register pair code in an instruction word as the operand address. The register pair specified is in the register bank specified by the register bank select flags (RBS0 and RBS1). This addressing can be used for the entire memory space.

[Operand format]

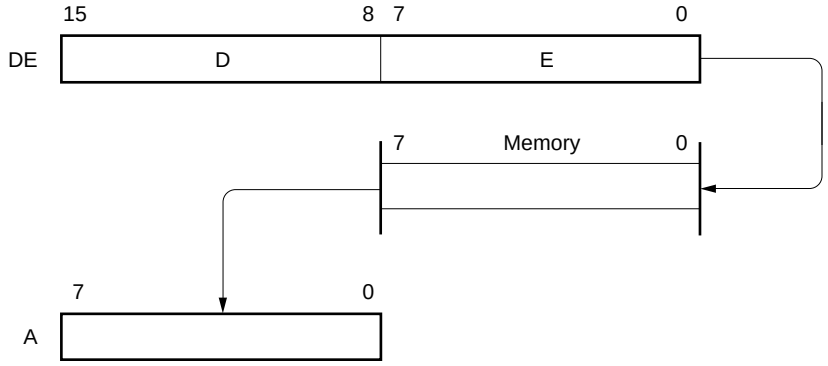
Symbol	Description
—	[DE], [HL]

[Example]

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

Operation code	1	0	0	0	0	1	0	1

[Illustration]



3.4.7 Based addressing

[Function]

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

[Operand format]

Symbol	Description
—	[HL + byte]

[Example]

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

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

3.4.8 Based indexed addressing

[Function]

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

Addition is performed by expanding the offset data as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all the memory spaces.

[Operand format]

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

[Example]

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

Operation code	1	0	1	0	1	0	1	1

3.4.9 Stack addressing

[Function]

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

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

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

[Example]

In the case of PUSH DE

Operation code	1	0	1	1	0	1	0	1

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The μPD178054 Subseries units incorporate input, output, and I/O ports consisting of 6, 3, and 53 pins, respectively. Figure 4-1 shows the port configuration. Every port is capable of 1-bit and 8-bit manipulations and can carry out considerably varied control operations. Besides port functions, the ports can also serve as on-chip hardware input/output pins.

Figure 4-1. Port Types

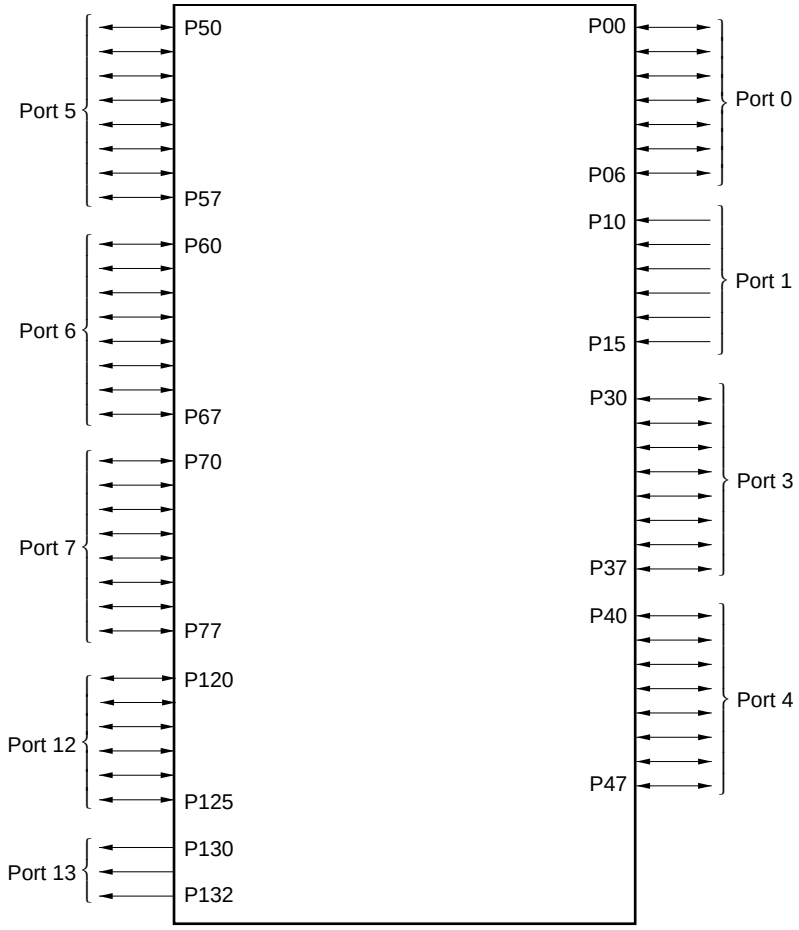


Table 4-1. Port Functions

Pin Name	I/O	Function	Alternate Function
P00 to P04	I/O	Port 0 7-bit I/O port Input/output can be specified in 1-bit units.	INTP0 to INTP4
P05, P06			—
P10 to P15	Input	Port 1 6-bit input port	ANI0 to ANI5
P30 to P32	I/O	Port 3 8-bit I/O port Input/output can be specified in 1-bit units.	—
P33			TI50
P34			TI51
P35			—
P36			BEEP0
P37			BUZ
P40 to 47	I/O	Port 4 8-bit I/O port Input/output can be specified in 1-bit units. An on-chip pull-up resistor can be specified by software. Interrupt function by key input is provided.	—
P50 to P57	I/O	Port 5 8-bit I/O port Input/output can be specified in 1-bit units.	—
P60 to P67	I/O	Port 6 8-bit I/O port Input/output can be specified in 1-bit units.	—
P70	I/O	Port 7 8-bit I/O port Input/output can be specified in 1-bit units.	SI30
P71			SO30
P72			$\overline{\text{SCK30}}$
P73			—
P74			SI31
P75			SO31
P76			$\overline{\text{SCK31}}$
P77			TI52
P120	I/O	Port 12 6-bit I/O port Input/output can be specified in 1-bit units.	SI32
P121			SO32
P122			$\overline{\text{SCK32}}$
P123			SI321
P124			SO321
P125			$\overline{\text{SCK321}}$
P130	Output	Port 13 3-bit output port N-ch open-drain output port (12 V tolerance)	TO50
P131			TO51
P132			TO52

4.2 Port Configuration

The ports consist of the following hardware.

Table 4-2. Port Configuration

Item	Configuration
Control register	Port mode register (PMm: m = 0, 3 to 7, 12)
Port	Total: 62 port pins (6 inputs, 3 outputs, 53 I/Os)

4.2.1 Port 0

Port 0 is a 7-bit I/O port with an output latch. Input or output mode can be specified for port 0 in 1-bit units using port mode register 0 (PM0).

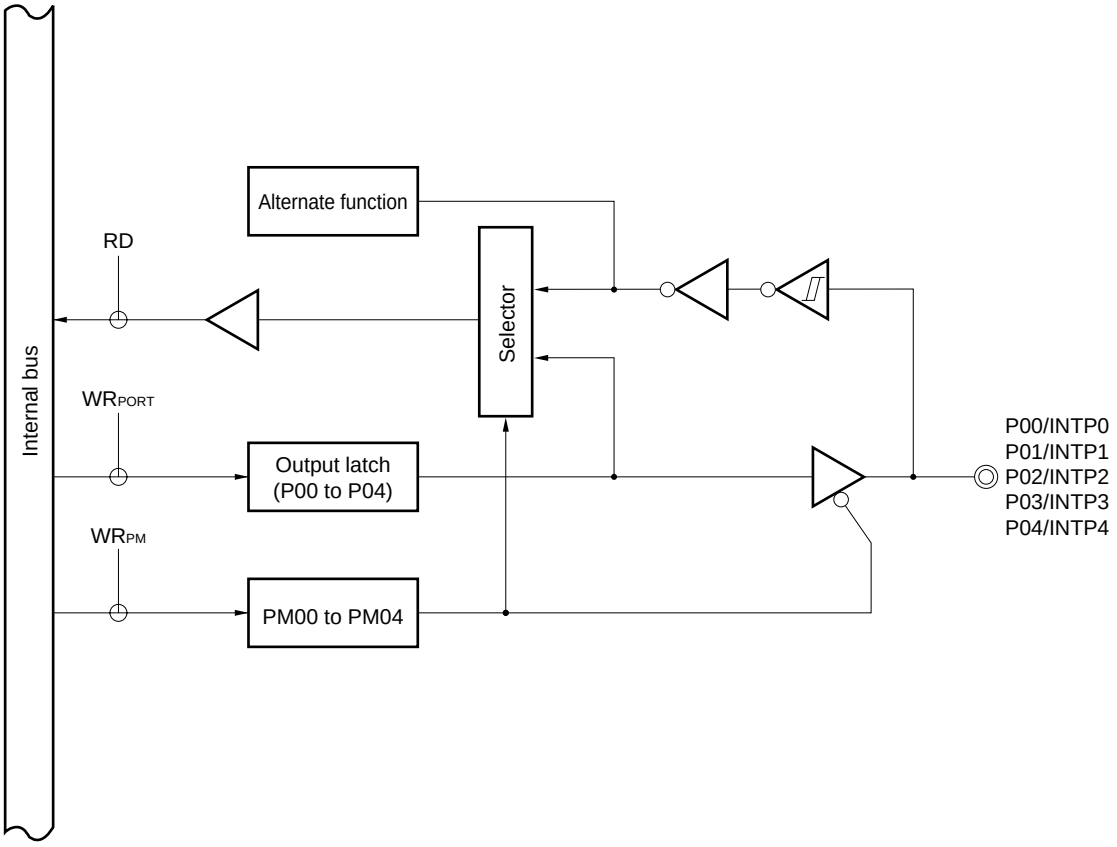
Alternate functions include external interrupt request input.

Reset input sets port 0 to the input mode.

Figures 4-2 and 4-3 show the block diagrams of port 0.

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

Figure 4-2. Block Diagram of P00 to P04

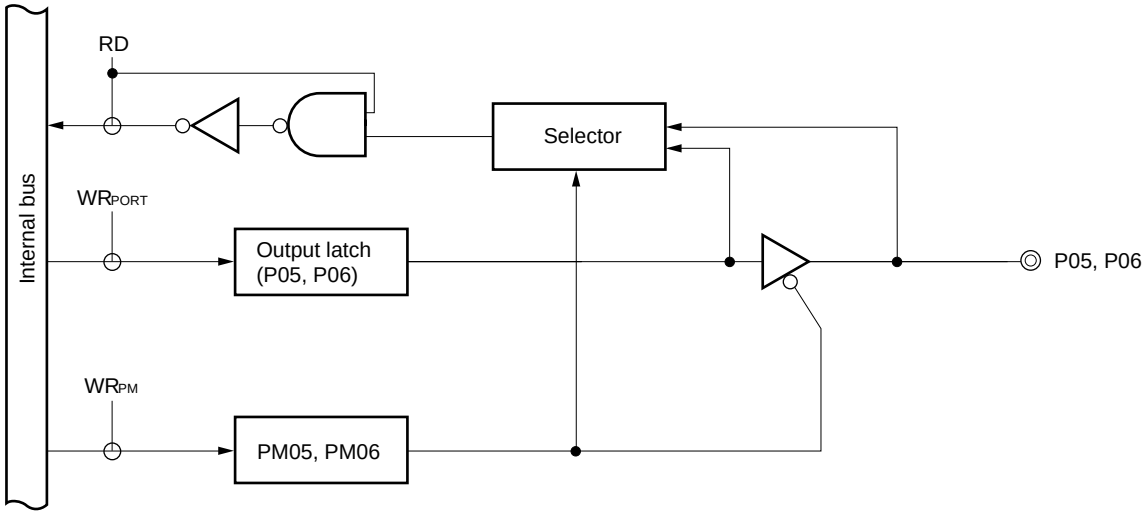


PM: Port mode register

RD: Port 0 read signal

WR: Port 0 write signal

Figure 4-3. Block Diagram of P05 and P06



PM: Port mode register
RD: Port 0 read signal
WR: Port 0 write signal

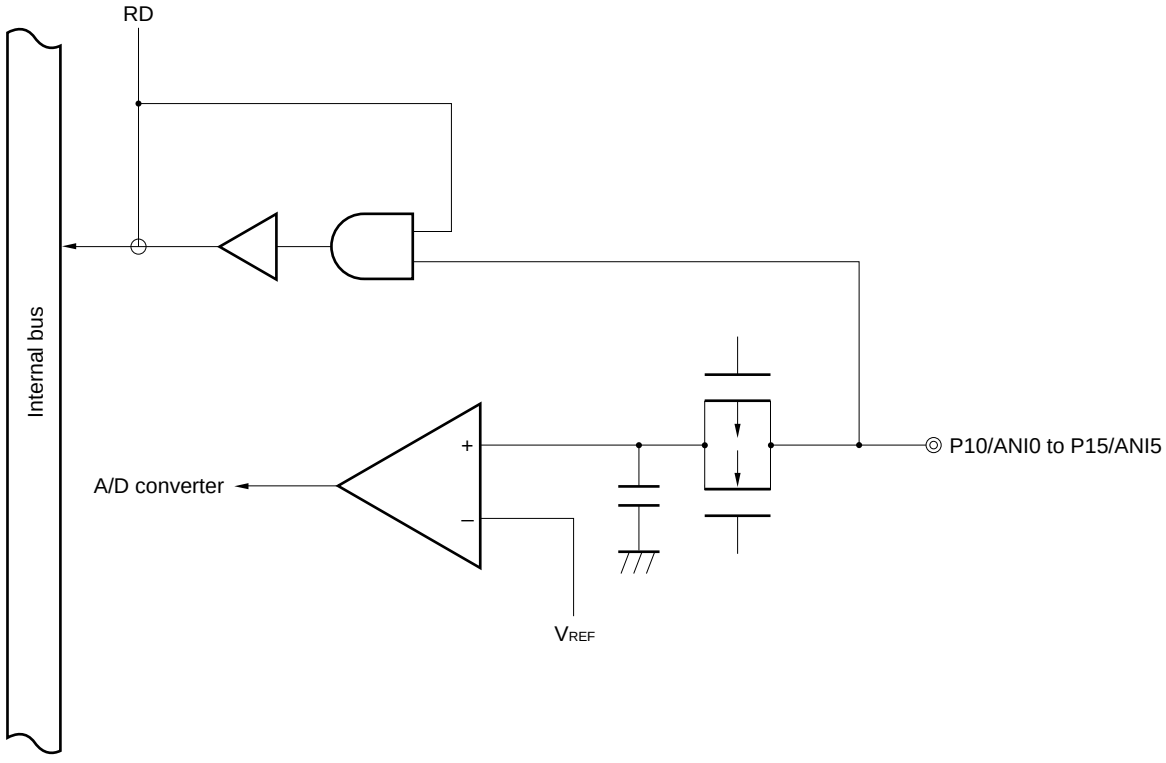
4.2.2 Port 1

Port 1 is a 6-bit input port.

Alternate functions include A/D converter analog input.

Figure 4-4 shows the block diagram of port 1.

Figure 4-4. Block Diagram of P10 to P15



RD : Port 1 read signal

4.2.3 Port 3

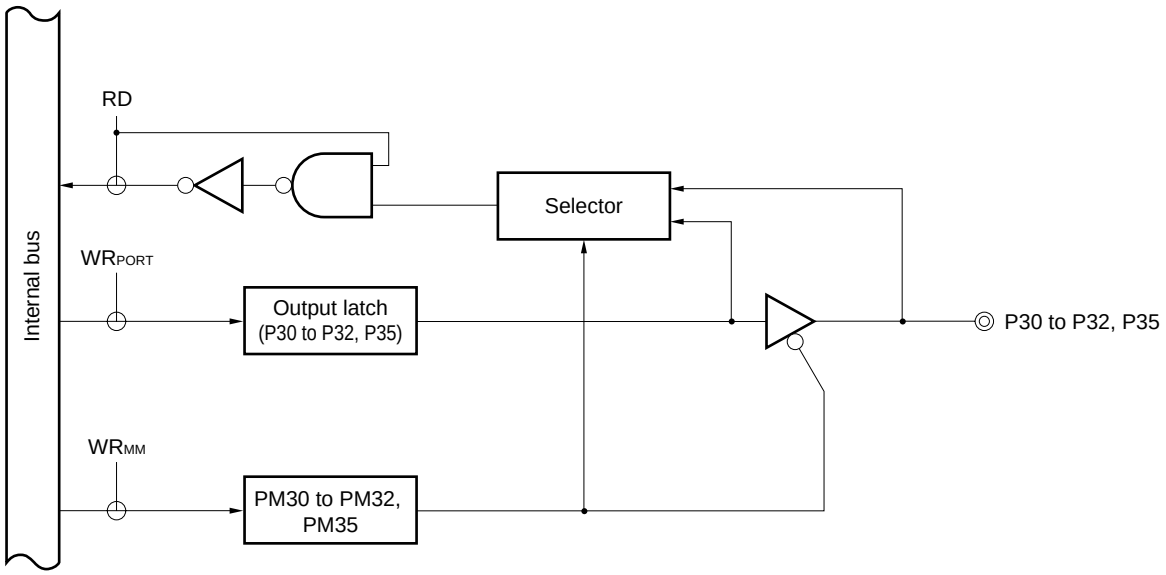
Port 3 is an 8-bit I/O port with an output latch. Input or output mode can be specified for port 3 in 1-bit units using port mode register 3 (PM3).

Alternate functions include timer input and buzzer output.

Reset input sets port 3 to the input mode.

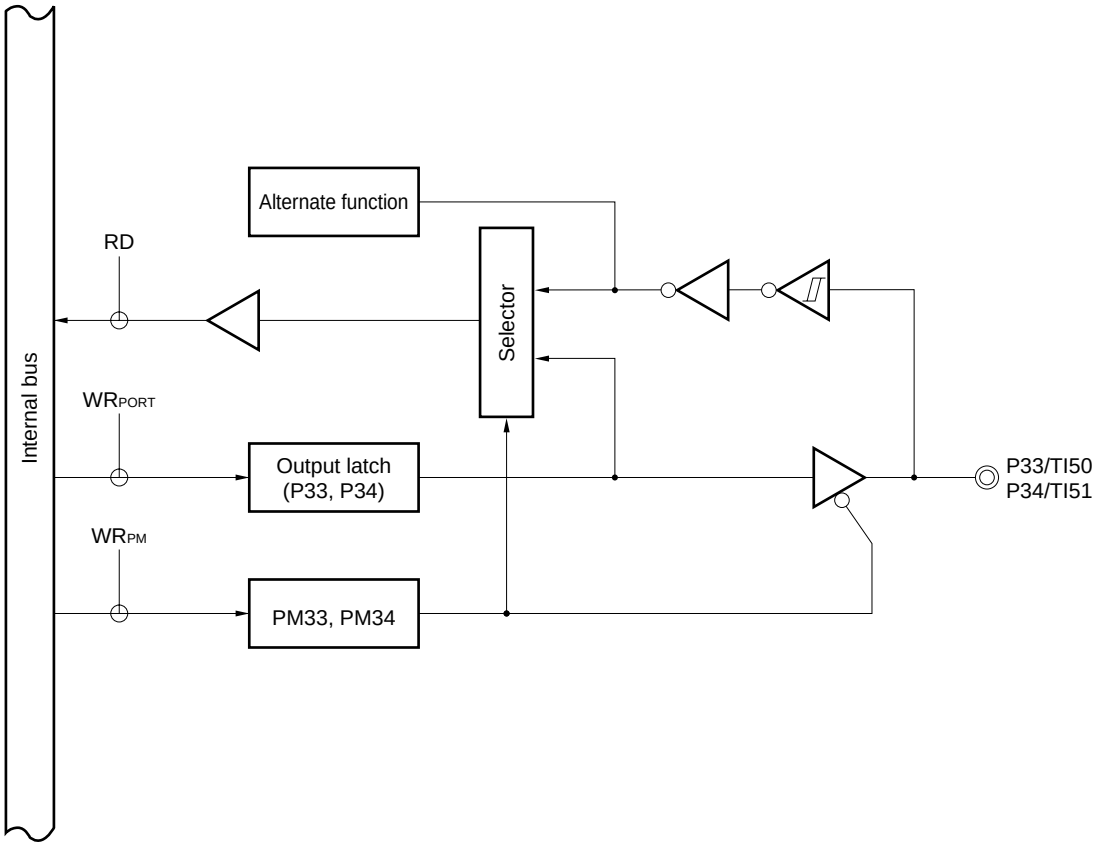
Figures 4-5 to 4-7 show the block diagrams of port 3.

Figure 4-5. Block Diagram of P30 to P32 and P35



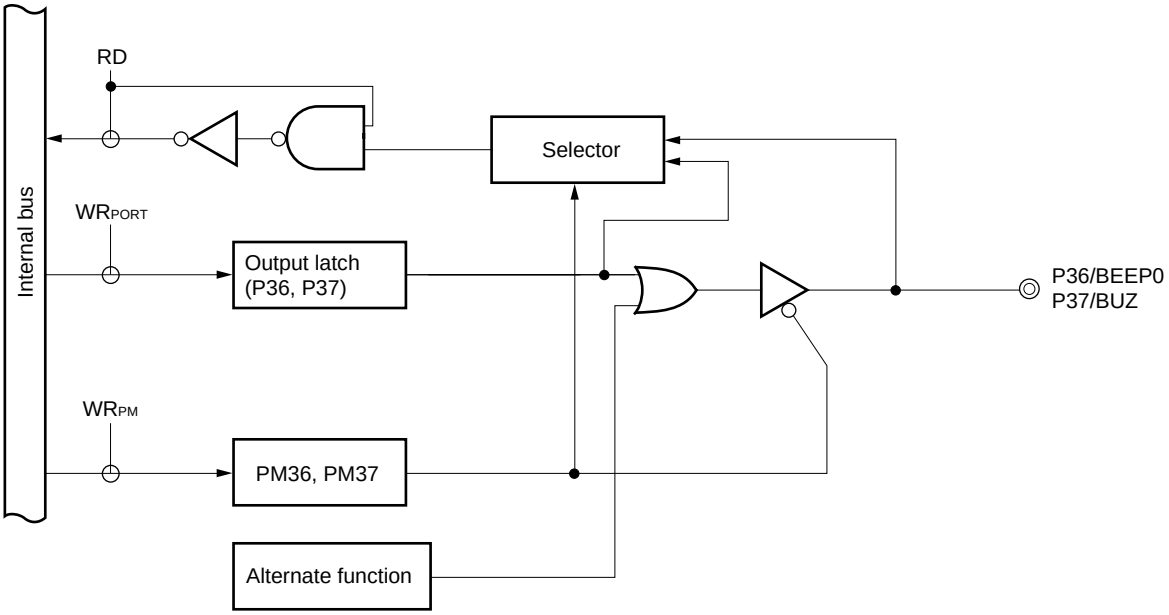
PM: Port mode register
RD: Port 3 read signal
WR: Port 3 write signal

Figure 4-6. Block Diagram of P33 and P34



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

Figure 4-7. Block Diagram of P36 and P37



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

4.2.4 Port 4

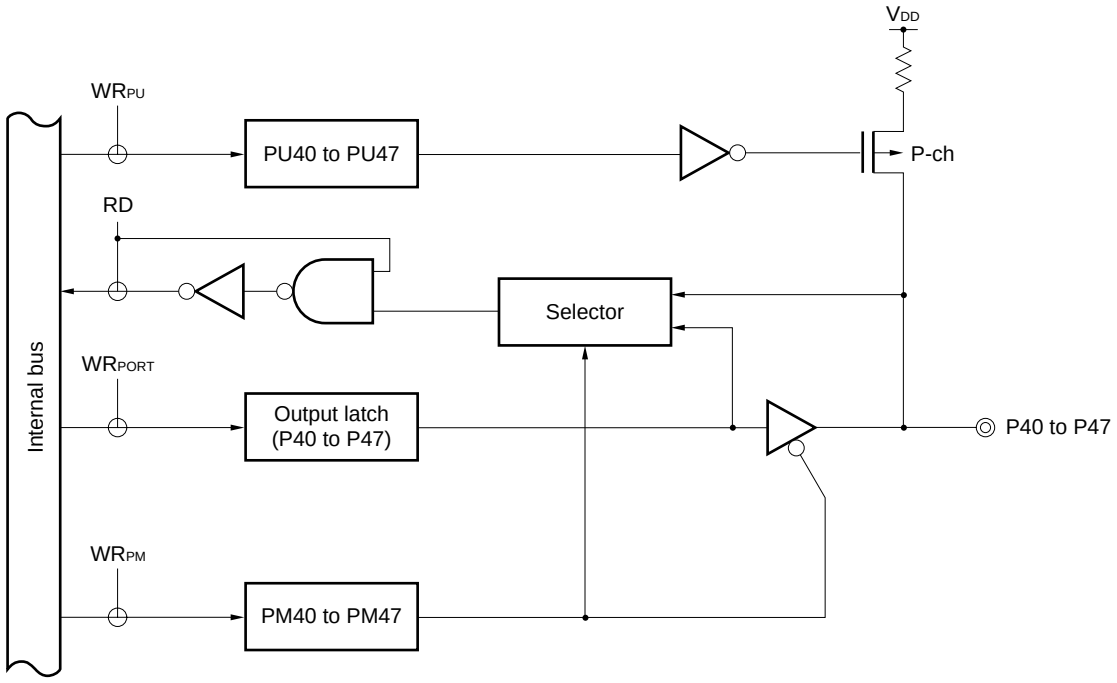
Port 4 is an 8-bit I/O port with an output latch. Input or output mode can be specified for port 4 in 1-bit units using port mode register 4 (PM4). Connection of pull-up resistors can be specified in 1-bit units using pull-up resistor option register 4 (PU4).

The interrupt request flag (KYIF) can be set to 1 by detecting key inputs. When using this function, be sure to set the MEM register to 01H.

Reset input sets port 4 to input mode.

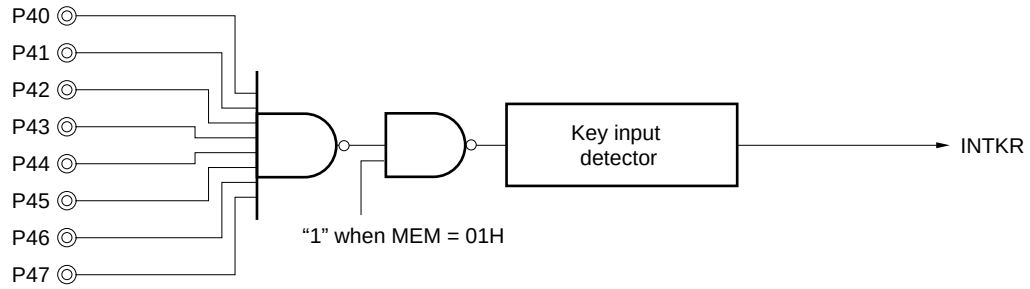
Figures 4-8 and 4-9 show a block diagram of port 4 and block diagram of the key input detector, respectively.

Figure 4-8. Block Diagram of P40 to P47



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

Figure 4-9. Block Diagram of Key Input Detector



Cautions 1. This register is valid only when the MEM register is set to 01H.

★ **2. Key return can be detected only when all the pins of P40 to P47 are high level. When any one is low level, even if falling edge is generated at the other pins, the key return signal cannot be detected.**

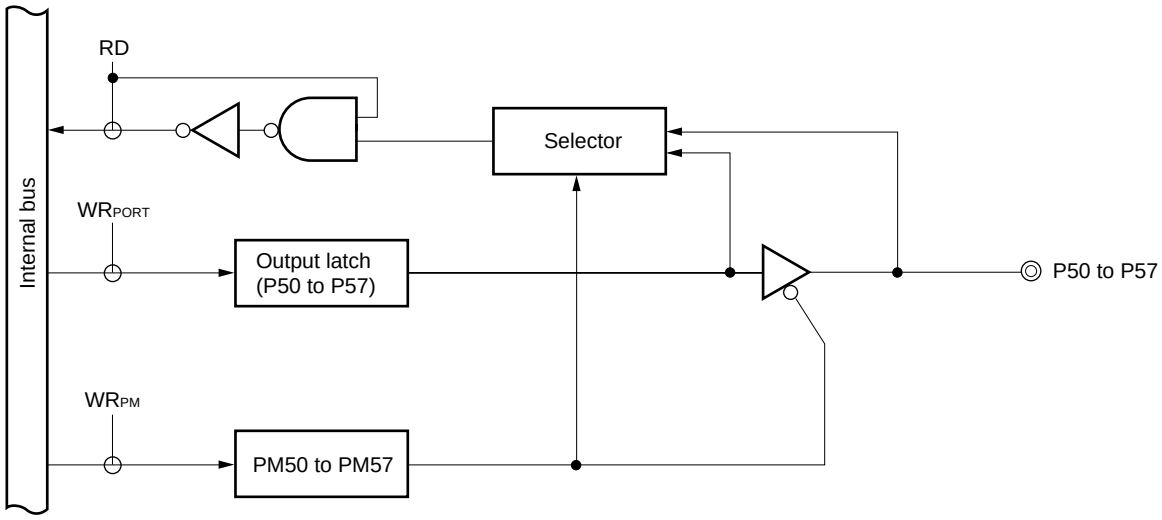
4.2.5 Port 5

Port 5 is an 8-bit I/O port with an output latch. Input or output mode can be specified for port 5 in 1-bit units using port mode register 5 (PM5).

Reset input sets port 5 to the input mode.

Figure 4-10 shows the block diagram of port 5.

Figure 4-10. Block Diagram of P50 to P57



PM: Port mode register

RD: Port 5 read signal

WR: Port 5 write signal

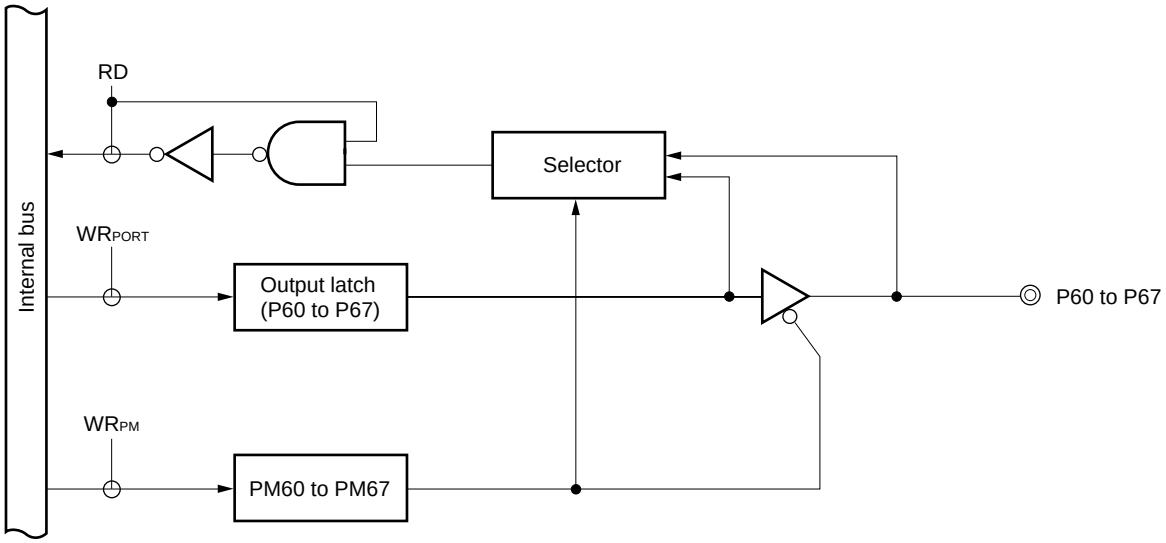
4.2.6 Port 6

Port 6 is an 8-bit I/O port with an output latch. Input or output mode can be specified for port 6 in 1-bit units using port mode register 6 (PM6).

Reset input sets port 6 to the input mode.

Figure 4-11 shows the block diagram of port 6.

Figure 4-11. Block Diagram of P60 to P67



PM: Port mode register
RD: Port 6 read signal
WR: Port 6 write signal

4.2.7 Port 7

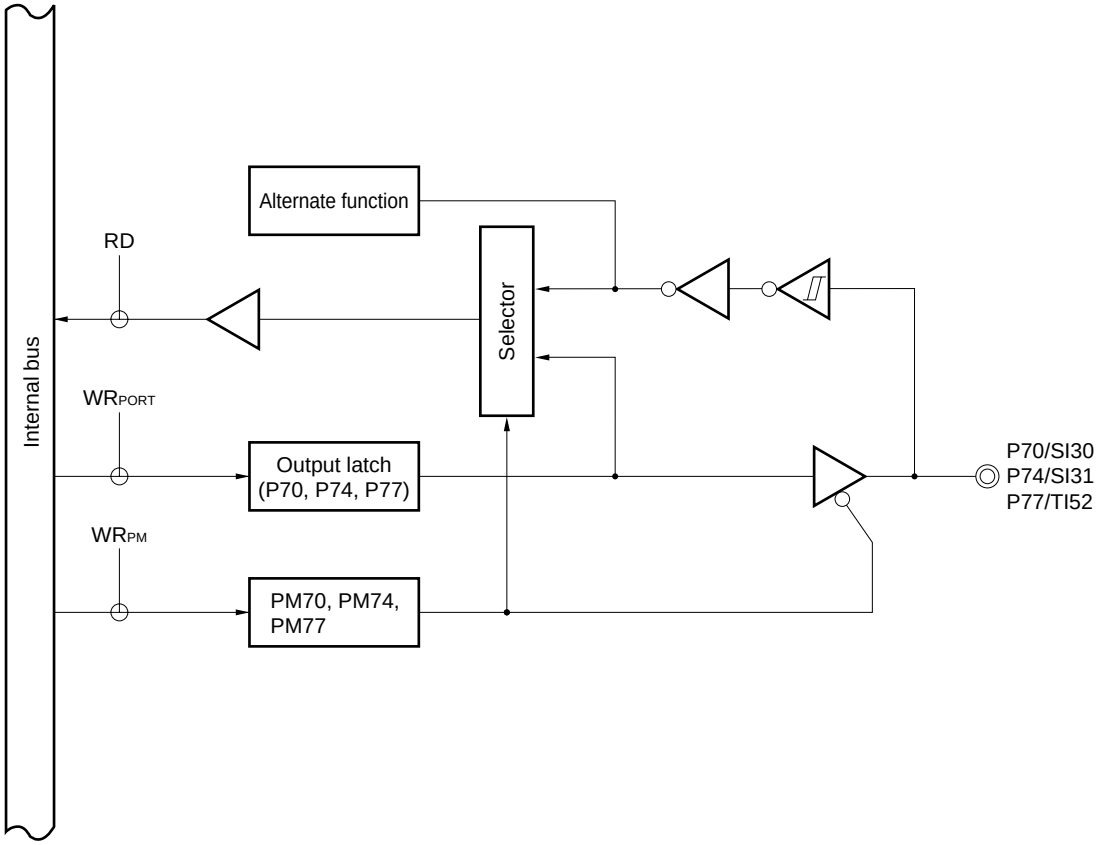
Port 7 is an 8-bit I/O port with an output latch. Input or output mode can be specified for port 7 in 1-bit units using port mode register 7 (PM7).

Alternate functions include serial interface data I/O, clock I/O, and timer input.

Reset input sets port 7 to the input mode.

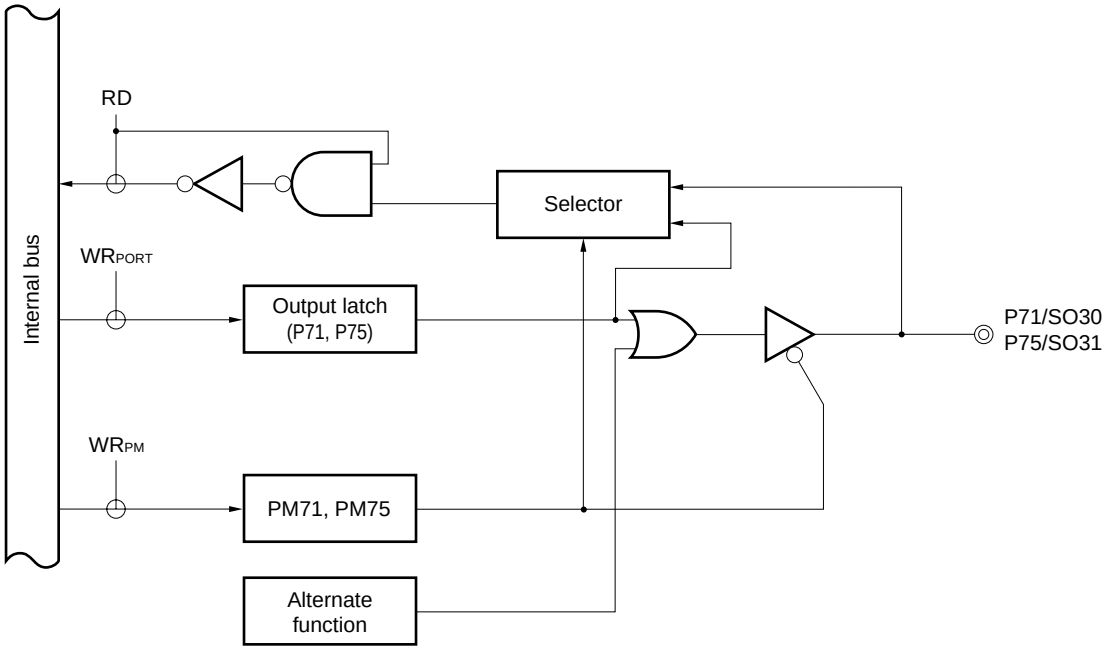
Figures 4-12 to 4-15 show the block diagrams of port 7.

Figure 4-12. Block Diagram of P70, P74, and P77



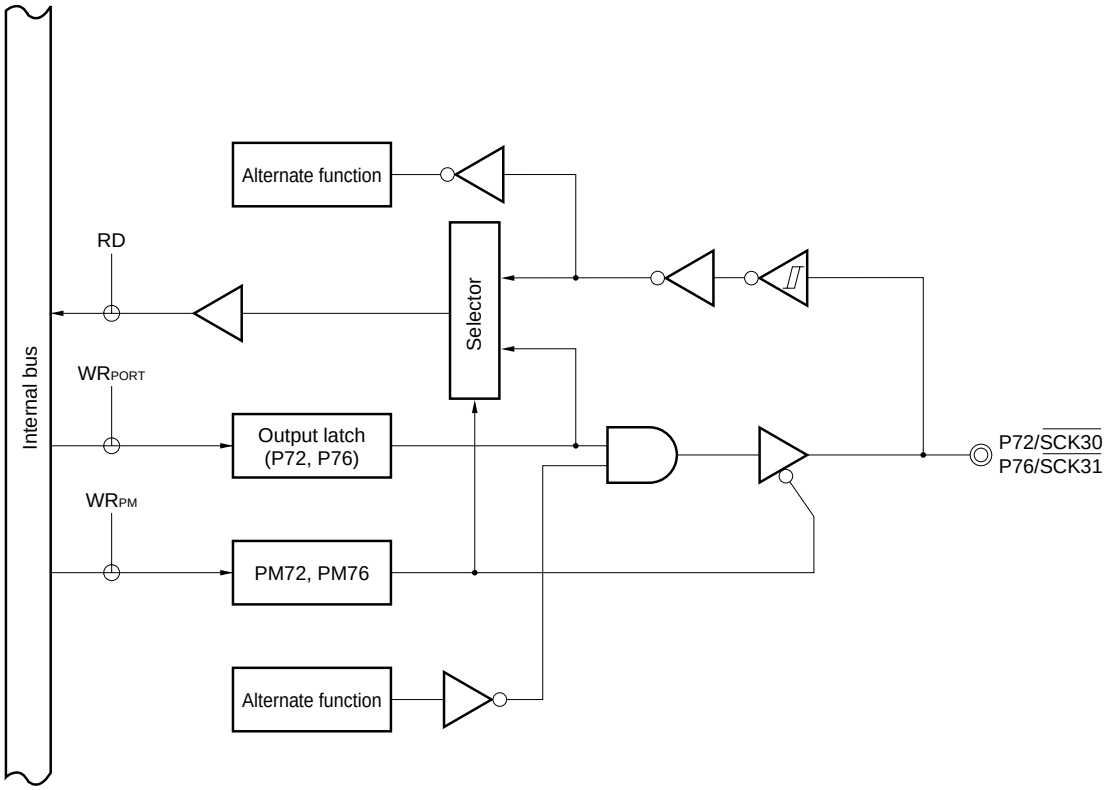
PM: Port mode register
RD: Port 7 read signal
WR: Port 7 write signal

Figure 4-13. Block Diagram of P71 and P75



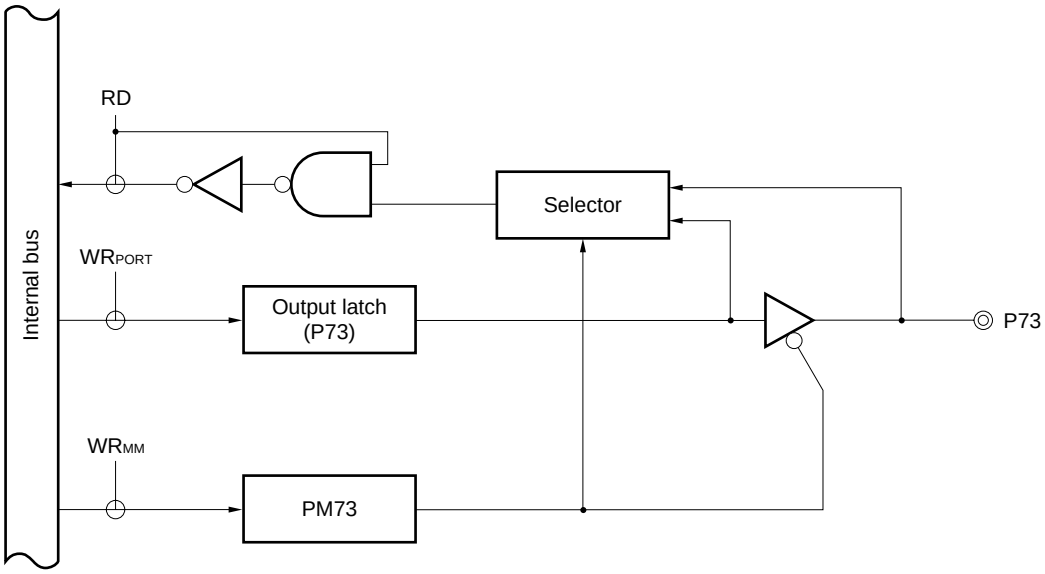
PM: Port mode register
RD: Port 7 read signal
WR: Port 7 write signal

Figure 4-14. Block Diagram of P72 and P76



PM: Port mode register
RD: Port 7 read signal
WR: Port 7 write signal

Figure 4-15. Block Diagram of P73



PM: Port mode register
RD: Port 7 read signal
WR: Port 7 write signal

4.2.8 Port 12

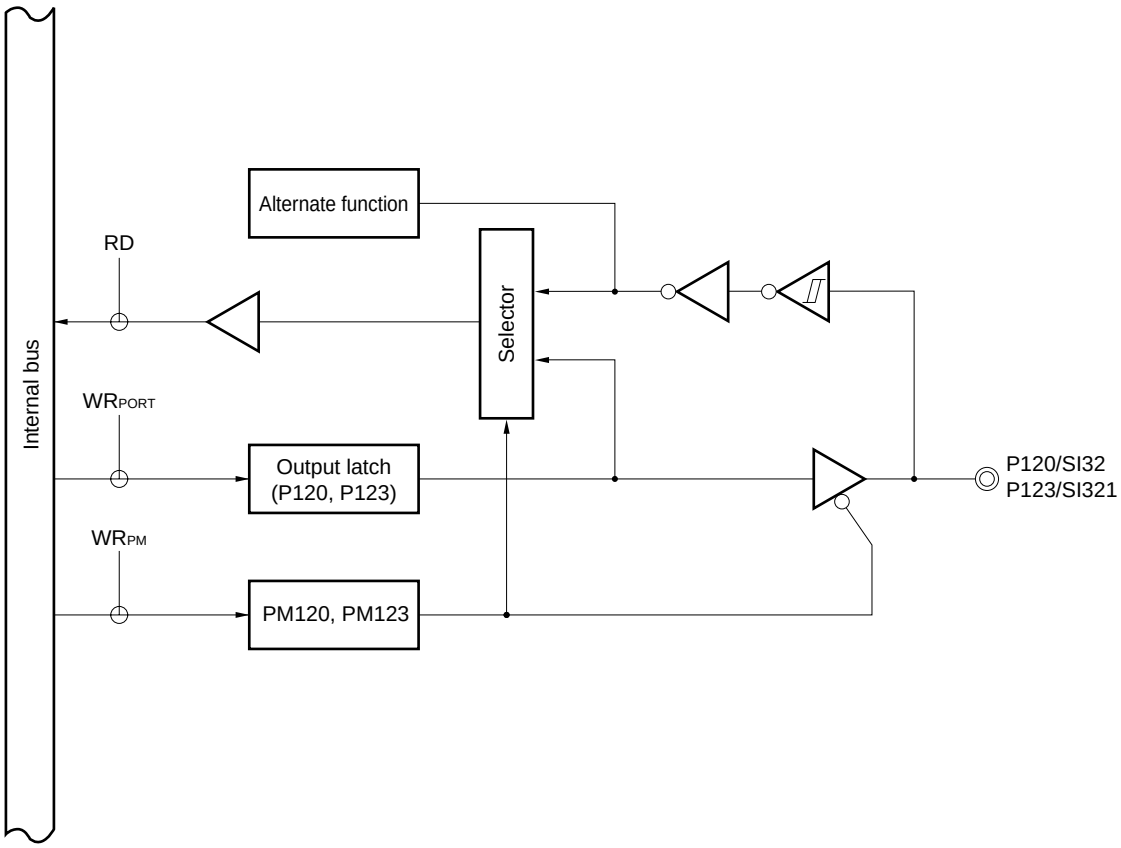
Port 12 is a 6-bit I/O port with an output latch. Input or output mode can be specified for port 12 in 1-bit units using port mode register 12 (PM12).

Alternate functions include serial interface data I/O and clock I/O.

Reset input sets port 12 to the input mode.

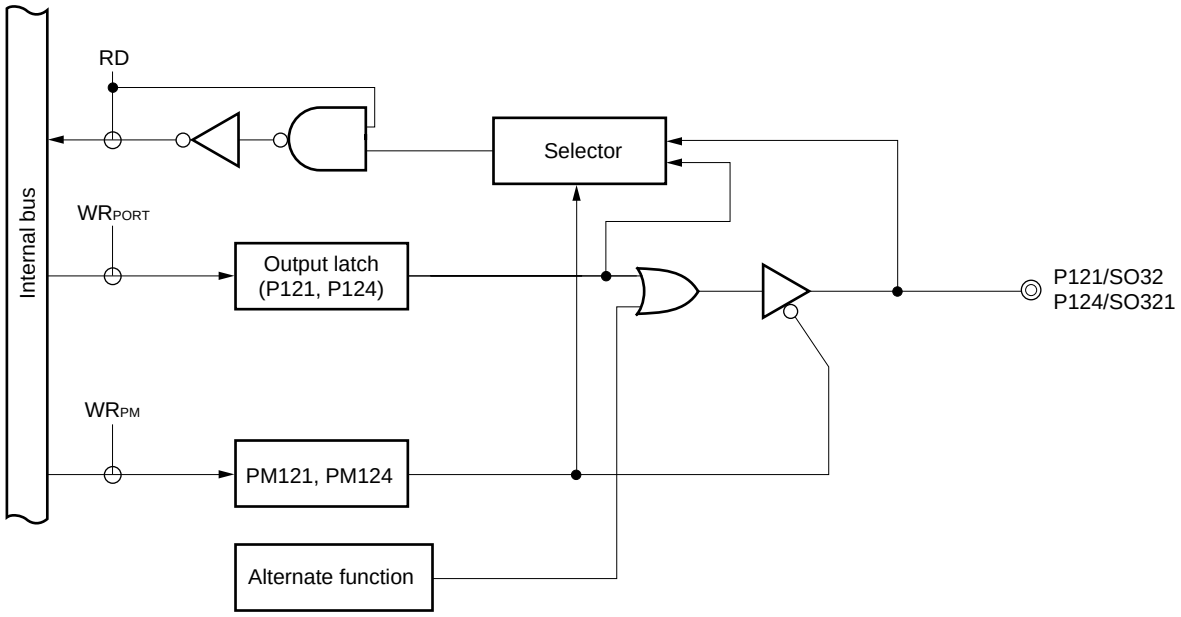
Figures 4-16 to 4-18 show the block diagrams of port 12.

Figure 4-16. Block Diagram of P120 and P123



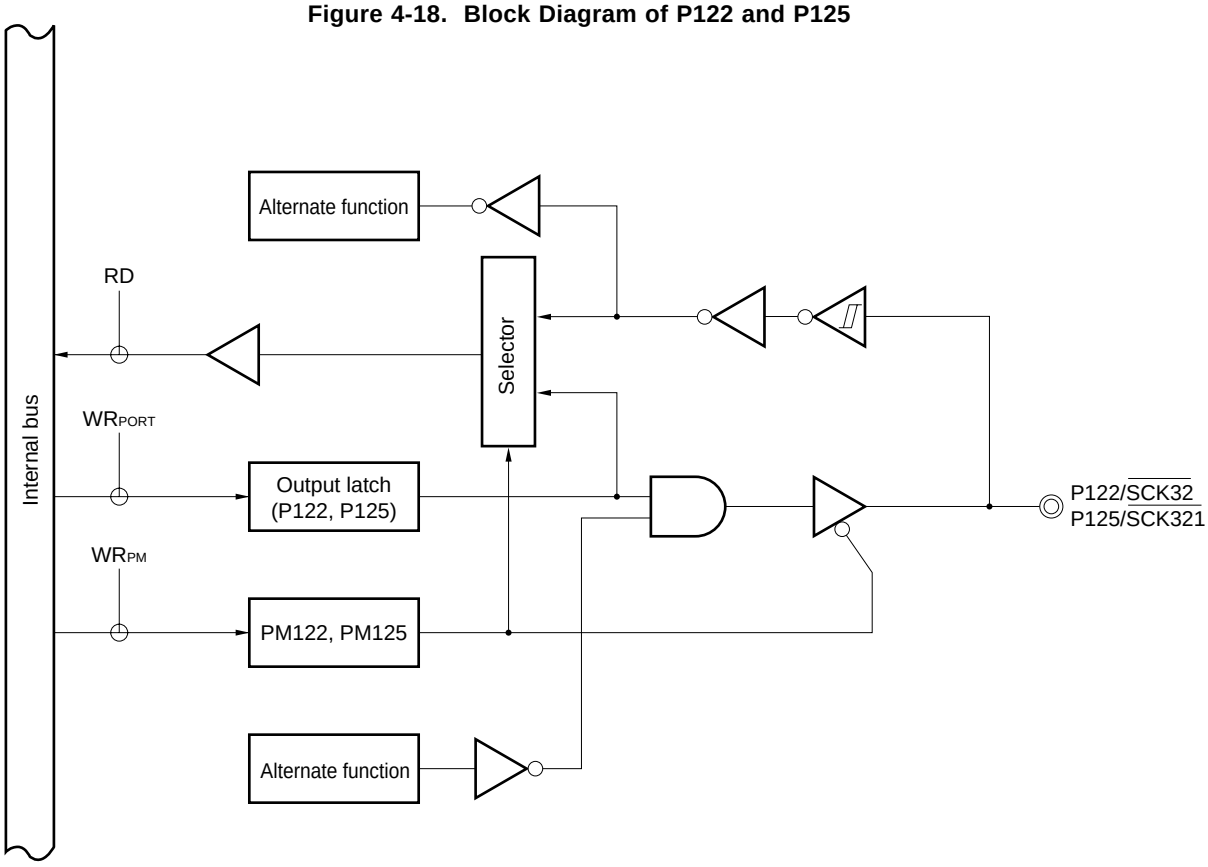
PM: Port mode register
RD: Port 12 read signal
WR: Port 12 write signal

Figure 4-17. Block Diagram of P121 and P124



PM: Port mode register
RD: Port 12 read signal
WR: Port 12 write signal

Figure 4-18. Block Diagram of P122 and P125



PM: Port mode register
RD: Port 12 read signal
WR: Port 12 write signal

4.2.9 Port 13

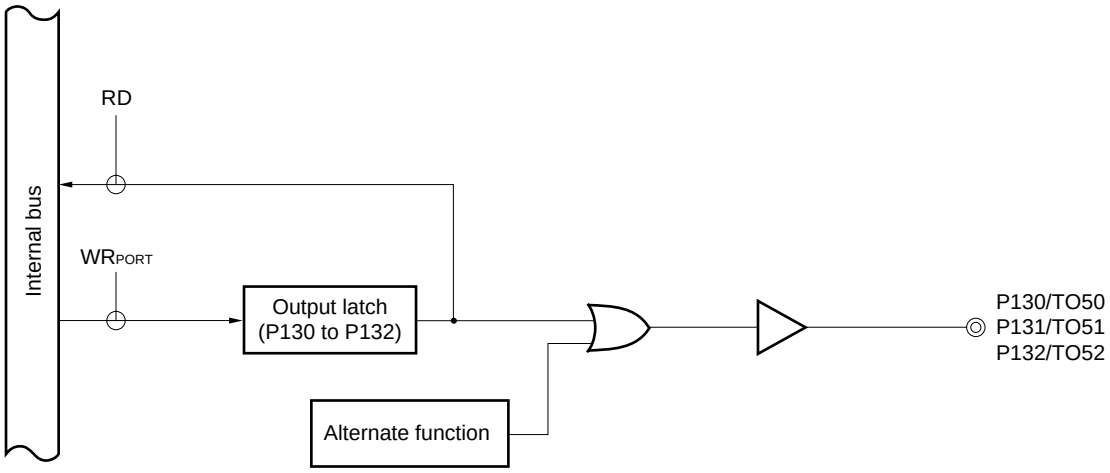
Port 13 is a 3-bit N-ch open-drain output port with an output latch.

The pins of this port are also used as timer output pins.

Reset input sets port 13 in the general-purpose output port mode.

The port 13 block diagram is shown in Figure 4-19.

Figure 4-19. Block Diagram of P130 to P132



RD: Port 13 read signal

WR: Port 13 write signal

4.3 Registers Controlling Port Functions

The following two types of registers control the ports.

- Port mode registers (PM0, PM3 to PM7, PM12)
- Pull-up resistor option register (PU4)

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

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

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

Reset input sets these registers to FFH.

When using a port pin as an alternate-function pin, set the values of the port mode registers and the output latches as shown in Table 4-3.

Cautions 1. P10 to P17 are input-only pins, and P130 to P132 are output-only pins.

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

★ **Table 4-3. Port Mode Register and Output Latch Settings When Using Alternate Functions**

Pin Name	Alternate Functions		PM××	P××
	Name	I/O		
P00 to P04	INTP0 to INTP4	Input	1	×
P33	TI50	Input	1	×
P34	TI51	Input	1	×
P36	BEEP0	Output	0	0
P37	BUZ	Output	0	0
P70	SI30	Input	1	×
P71	SO30	Output	0	0
P72	$\overline{\text{SCK30}}$	Input	1	×
		Output	0	0
P74	SI31	Input	1	×
P75	SO31	Output	0	0
P76	$\overline{\text{SCK31}}$	Input	1	×
		Output	0	0
P77	TI52	Input	1	×
P120	SI32	Input	1	×
P121	SO32	Output	0	0
P122	$\overline{\text{SCK32}}$	Input	1	×
		Output	0	0
P123	SI321	Input	1	×
P124	SO321	Output	0	0
P125	$\overline{\text{SCK321}}$	Input	1	×
		Output	0	0
P130 to P132	TO50 to TO52	Output	—	0

Caution When using the above alternate function pins as an output port, be sure to set the output latch (P××) to 0.

Remark ×: Don't care
PM××: Port mode register
P××: Output latch of port

Figure 4-20. Format of Port Mode Registers

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FF24H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FF25H	FFH	R/W
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60	FF26H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FF27H	FFH	R/W
PM12	1	1	PM125	PM124	PM123	PM122	PM121	PM120	FF2CH	FFH	R/W

PMmn	Pmn pin input/output mode selection (m = 0, 3 to 7, 12 : n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

(2) Pull-up resistor option register 4 (PU4)

This register is used to specify the use of the internal pull-up resistors of port 4. A pull-up resistor can only be used internally for the bit specified by PU4.

PU4 can be set with a 1-bit or 8-bit memory manipulation instruction.

Reset input sets PU4 to 00H.

Figure 4-21. Format of Pull-up Resistor Option Register 4 (PU4)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40	FF34H	00H	R/W

PU4n	Selection of internal pull-up resistor for P4n (n = 0 to 7)
0	Internal pull-up resistor not used
1	Internal 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 I/O ports

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

4.4.2 Reading from I/O ports

(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 I/O ports

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

CHAPTER 5 CLOCK GENERATOR

5.1 Functions of Clock Generator

The clock generator generates the clock to be supplied to the CPU and peripheral hardware. This system clock oscillator is connected to 4.5 MHz crystal resonator. At this time, set bit 0 (DTSCK0) of the DTS system clock select register (DTSCK) to 1. Set the DTSCK0 flag after power application and reset by the $\overline{\text{RESET}}$ pin, and before using the basic timer, buzzer output control circuit, PLL frequency synthesizer, and frequency counter.

Oscillation can be stopped by executing the STOP instruction.

Figure 5-1. Format of DTS System Clock Select Register (DTSCK)

Symbol	7	6	5	4	3	2	1	<0>	Address	After reset	R/W
DTSCK	0	0	0	0	0	0	0	DTSCK0	FFAAH	00H	R/W

DTSCK0	Selects system clock
1	4.5 MHz
0	Setting prohibited

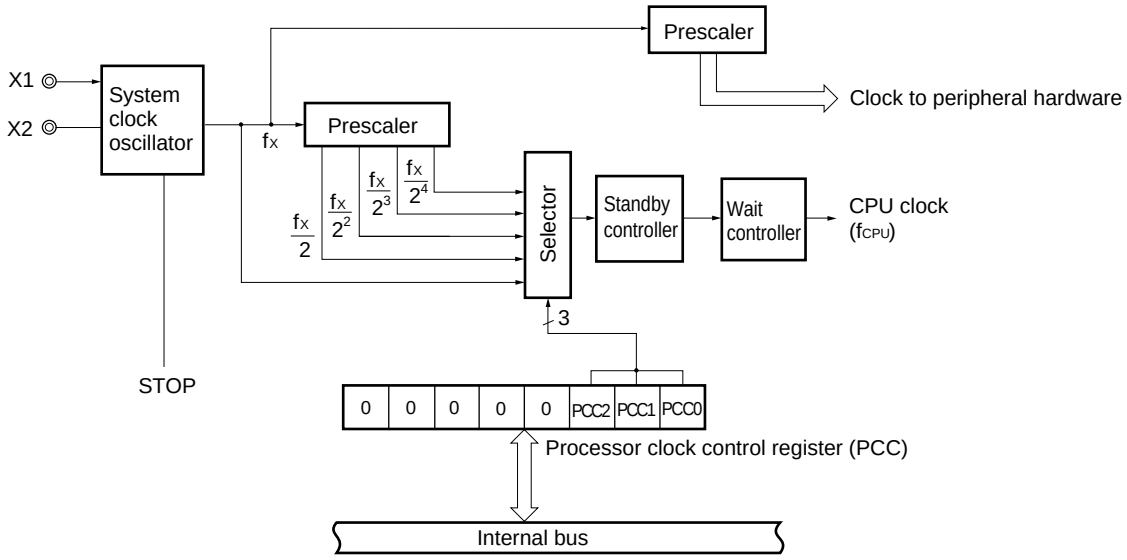
5.2 Configuration of Clock Generator

The clock generator consists of the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control register	Processor clock control register (PCC)
Oscillator	System clock oscillator

Figure 5-2. Block Diagram of Clock Generator



5.3 Register Controlling Clock Generator

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

PCC sets the CPU clock.

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

Reset input sets PCC to 04H.

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

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

R/W	PCC2	PCC1	PCC0	CPU clock (f_{CPU}) selection
	0	0	0	f_X (0.45 μs)
	0	0	1	$f_X/2$ (0.89 μs)
	0	1	0	$f_X/2^2$ (1.78 μs)
	0	1	1	$f_X/2^3$ (3.56 μs)
	1	0	0	$f_X/2^4$ (7.11 μs)
	Other than above			Setting prohibited

Note Bits 3 to 7 are read only.

Remarks 1. f_X: System clock oscillation frequency

2. ( ): Minimum instruction execution time: $2/f_{CPU}$ at f_X = 4.5 MHz operation

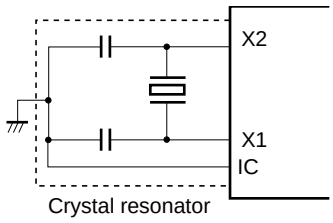
5.4 System Clock Oscillator

5.4.1 System clock oscillator

The system clock oscillator oscillates with a crystal resonator (4.5 MHz TYP.) connected to the X1 and X2 pins. Figure 5-4 shows an external circuit of the system clock oscillator.

Figure 5-4. External Circuit of System Clock Oscillator

Crystal oscillation



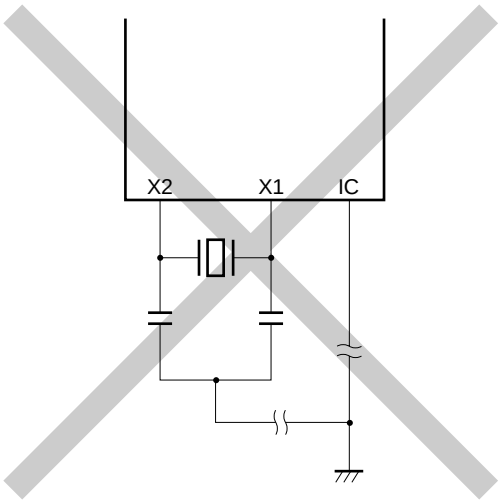
Caution When using a system clock oscillator, wire as follows in the area enclosed by the broken lines in Figure 5-4 to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as GND. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

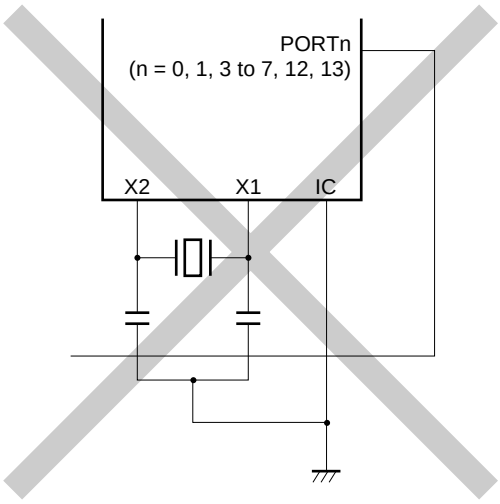
Figure 5-5 shows examples of incorrectly connected resonators.

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

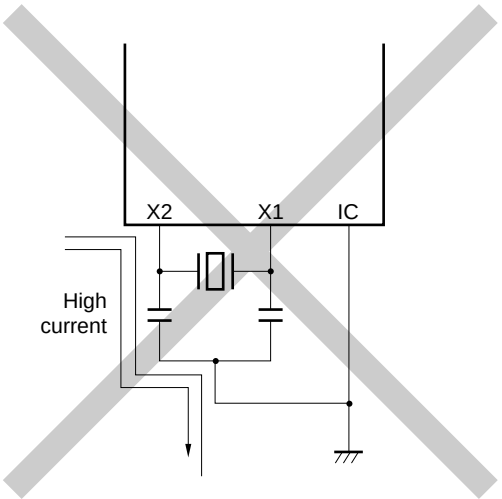
(a) Wiring of connection circuits is too long



(b) Signal lines cross each other



(c) High fluctuating current is near a signal lines



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

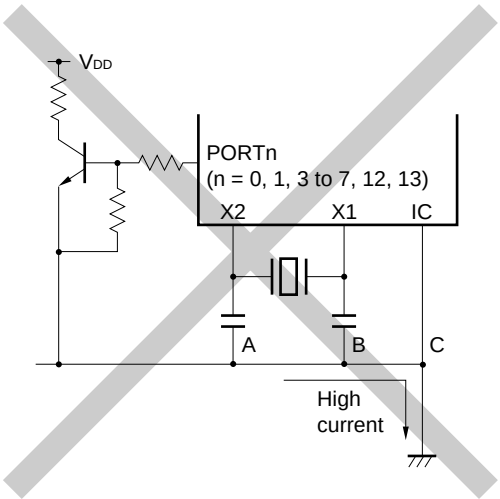
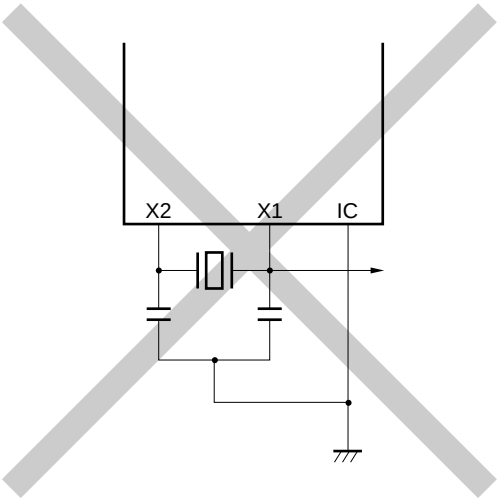


Figure 5-5. Examples of Incorrect Resonator Connection (2/2)

(e) Signals are fetched



5.4.2 Divider

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

5.5 Clock Generator Operations

The clock generator generates the following types of clocks and controls the CPU operating mode, such as the standby mode.

- System clock f_X
- CPU clock f_{CPU}
- Clock to peripheral hardware

The following clock generator functions and operations are determined by the processor clock control register (PCC).

(a) Upon generation of the $\overline{\text{RESET}}$ signal, the lowest speed mode of the system clock (7.11 μs when operated at 4.5 MHz) is selected (PCC = 04H). System clock oscillation stops while a low level is applied to the $\overline{\text{RESET}}$ pin.

(b) One of the five CPU clock types (0.45, 0.89, 1.78, 3.56, 7.11 μs at 4.5 MHz) can be selected by setting PCC.

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

(d) The system clock is divided and supplied to the peripheral hardware. The peripheral hardware also stops if the system clock is stopped.

5.6 Changing System Clock and CPU Clock Settings

5.6.1 Time required for switching between system clock and CPU clock

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

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

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

Set Values Before Switching			Set Values 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 with the preswitched CPU clock.

CHAPTER 6 8-BIT TIMER/EVENT COUNTERS 50 TO 53

6.1 Functions of 8-Bit Timer/Event Counters 50 to 53

8-bit timer/event counters 50 to 53 have the following two modes.

- Mode in which an 8-bit timer/event counter is used alone (single mode)
- Mode in which the two timer/event counters are cascaded (cascade mode with a resolution of 16 bits)

These two modes are explained below.

(1) Mode in which an 8-bit timer/event counter is used alone (single mode)

The timer/event counter operates as an 8-bit timer/event counter.
In this mode, the following functions can be used.

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

Caution Timer 53 can be used only as an interval timer since it does not include timer input and output pins.

(2) Mode in which the two timer/event counters are cascaded (cascade mode with a resolution of 16 bits)

By connecting timer 50 or timer 52 as a lower timer and timer 51 or timer 53 as a higher timer in cascade, they operate as a 16-bit timer/event counter.
In this mode, the following functions can be used:

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

Figures 6-1 to 6-4 show the block diagrams of 8-bit timer/event counters 50 to 53.

Figure 6-1. Block Diagram of 8-Bit Timer/Event Counter 50

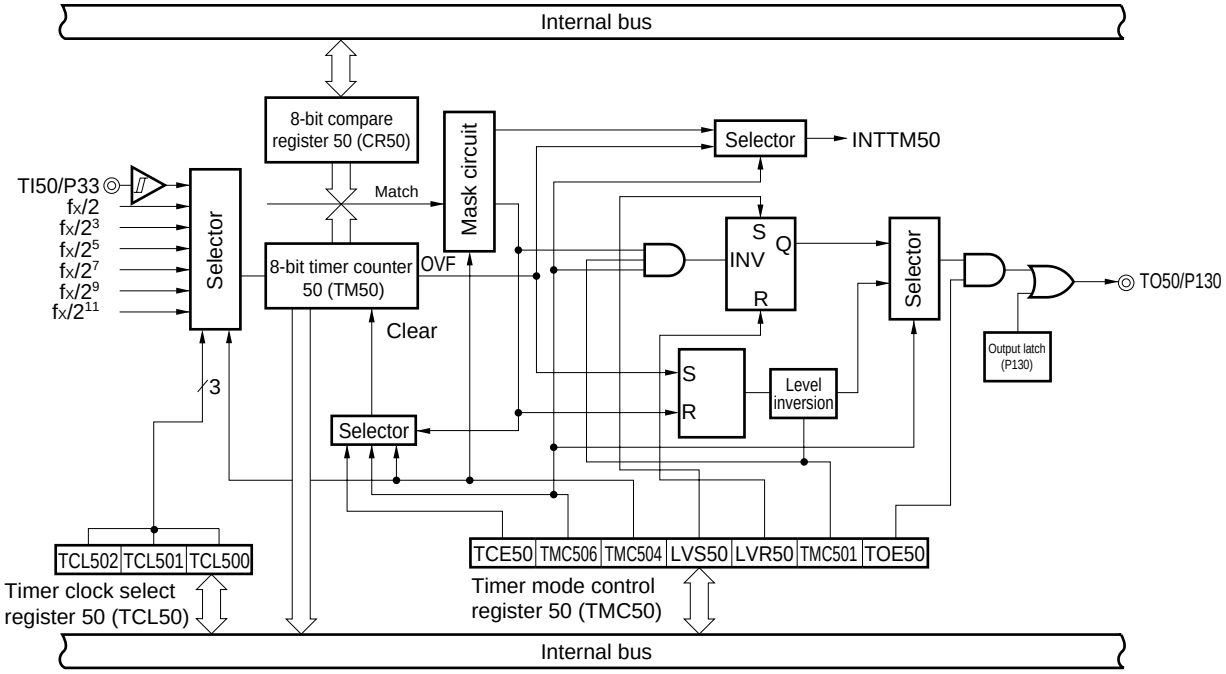


Figure 6-2. Block Diagram of 8-Bit Timer/Event Counter 51

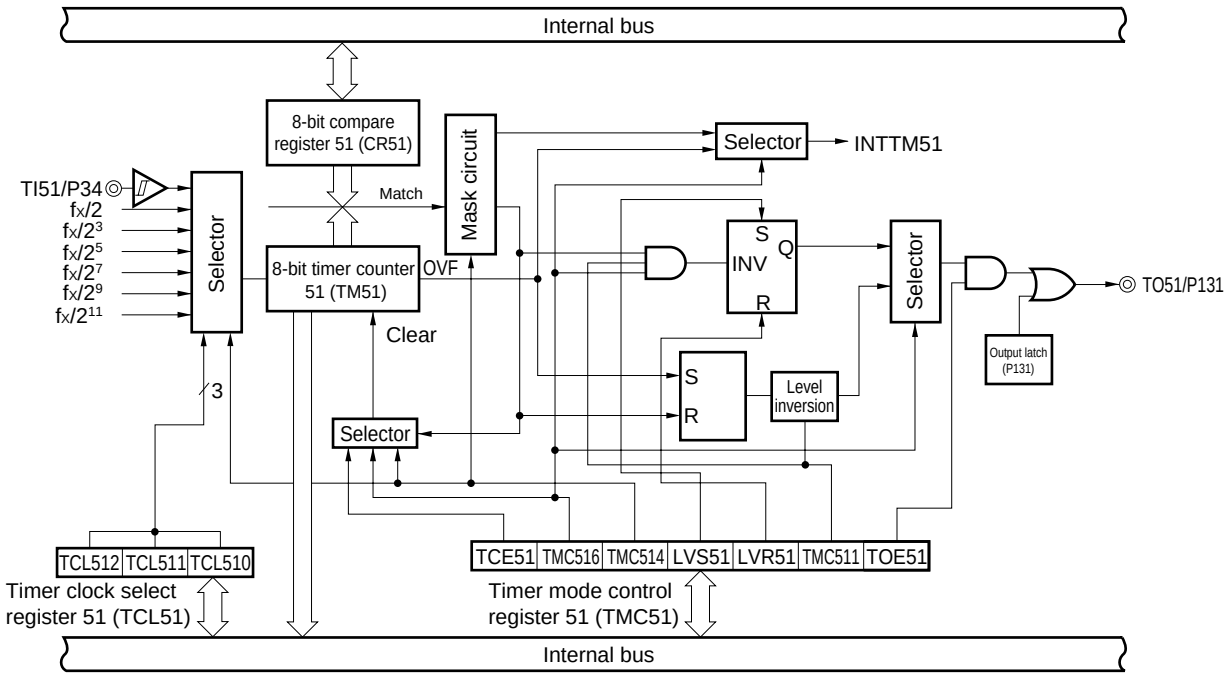


Figure 6-3. Block Diagram of 8-Bit Timer/Event Counter 52

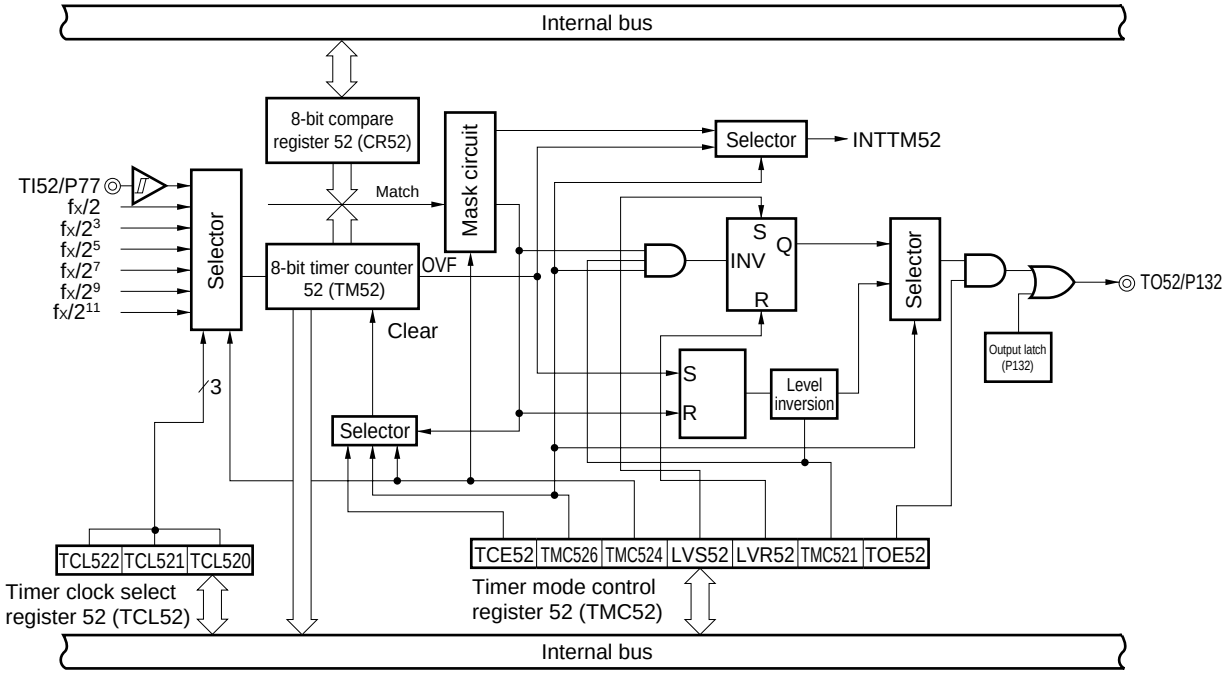
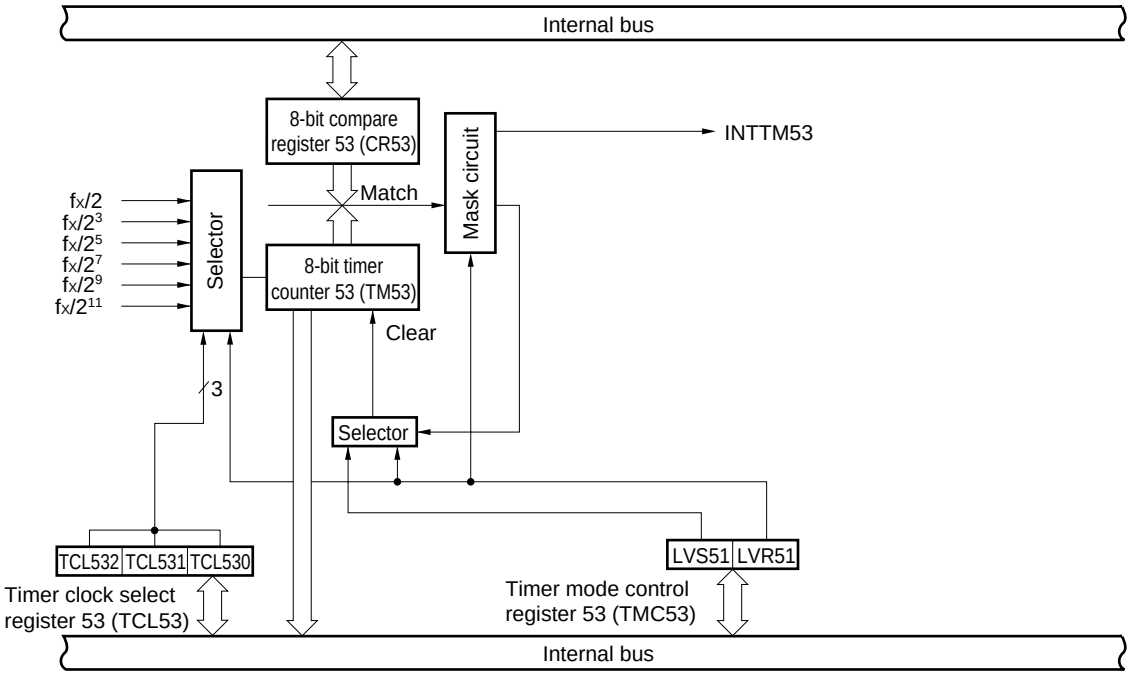


Figure 6-4. Block Diagram of 8-Bit Timer 53



6.2 Configuration of 8-Bit Timer/Event Counters 50 to 53

8-bit timer/event counters 50 to 53 consist of the following hardware.

Table 6-1. Configuration of 8-Bit Timer/Event Counters 50 to 53

Item	Configuration
Timer registers	8-bit timer counters 50, 51, 52, and 53 (TM50 to TM53)
Registers	8-bit compare registers 50, 51, 52, and 53 (CR50 to CR53)
Timer outputs	3 lines (TO50 to TO52)
Control registers	• Timer clock select registers 50, 51, 52, and 53 (TCL50 to TCL53) • 8-bit timer mode control registers 50, 51, 52, and 53 (TMC50 to TMC53)

(1) 8-bit timer counters 50, 51, 52, and 53 (TM50 to TM53)

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

The counter is incremented at the rising edge of the count clock.

TM50 and TM51 or TM52 and TM53 can be cascaded and used as a 16-bit timer.

When TM50 and TM51 are cascaded and used as a 16-bit timer, its value can be read using a 16-bit memory manipulation instruction. However, because TM50 and TM51 are connected with the internal 8-bit bus, they are read one at a time. Therefore, read the value of TM50 and TM51 when used as a 16-bit timer two times for comparison, taking changes in the values during counting into consideration.

When TM52 and TM53 are cascaded and used as a 16-bit timer, its value can be read using a 16-bit memory manipulation instruction. However, because TM52 and TM53 are connected with the internal 8-bit bus, they are read one at a time. Therefore, read the value of TM52 and TM53 when used as a 16-bit timer two times for comparison, taking changes in the values during counting into consideration.

If the count value is read while the timer is operating, stop input of the count clock, and read the count value at that point. The count value is cleared to 00H in the following cases.

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

<2> Clearing TCE5n

<3> Match between TM5n and CR5n in mode in which the timer is cleared and started on match between TM5n and CR5n

Caution When TM50 and TM51 or TM52 and TM53 are cascaded, the value of the timer is cleared to 00H even if the least significant bit (TCE50 or TCE52) of timer mode control register 50 (TMC50) or 52 (TMC52) is cleared.

Remark n = 0 to 3

(2) 8-bit compare registers 50, 51, 52, and 53 (CR50 to CR53)

The value set to CR5n is always compared with the value of 8-bit timer counter 5n (TM5n). When the value of a compare register matches the count value of the corresponding counter, an interrupt request (INTTM5n) is generated (in a mode other than PWM mode).

If TM50 and TM51 are cascaded and used as a 16-bit timer, CR50 and CR51 operate together as a 16-bit compare register. The 16-bit counter value and 16-bit compare register value are compared, and when the two values match, an interrupt request (INTTM50) is generated. At this time, the interrupt request INTTM51 is also generated. Therefore, mask INTTM51 when using TM50 and TM51 in the cascade mode.

If TM52 and TM53 are cascaded and used as a 16-bit timer, CR52 and CR53 operate together as a 16-bit compare register. The 16-bit counter value and 16-bit compare register value are compared, and when the two values match, an interrupt request (INTTM52) is generated. At this time, the interrupt request INTTM53 is also generated. Therefore, mask INTTM53 when using TM52 and TM53 in the cascade mode.

Caution When TM50 and TM51 or TM52 and TM53 are cascaded, be sure to change the CR5n setting value after stopping the timer operation of cascaded TM5n.

Remark n = 0 to 3

6.3 Registers Controlling 8-Bit Timer/Event Counters 50 to 53

The following two types of registers control the 8-bit timer/event counters 50 to 53.

- Timer clock select registers 50 to 53 (TCL50 to TCL53)
- 8-bit timer mode control registers 50 to 53 (TMC50 to TMC53)

(1) Timer clock select registers 50 to 52 (TCL50 to TCL52)

These registers select the count clock of 8-bit timer counter 5n (TM5n) and the valid edge of the TI5n input.

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

Reset input clears TCL50 to TCL52 to 00H.

Remark n = 0 to 2

Figure 6-5. Format of Timer Clock Select Registers 50 to 52 (TCL50 to TCL52)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL50	0	0	0	0	0	TCL502	TCL501	TCL500	FF84H	00H	R/W
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510	FF87H	00H	R/W
TCL52	0	0	0	0	0	TCL522	TCL521	TCL520	FF74H	00H	R/W

TCL5n2	TCL5n1	TCL5n0	Count clock selection
0	0	0	Falling edge of TI5n
0	0	1	Rising edge of TI5n
0	1	0	$f_X/2$ (2.25 MHz)
0	1	1	$f_X/2^3$ (563 kHz)
1	0	0	$f_X/2^5$ (141 kHz)
1	0	1	$f_X/2^7$ (35.2 kHz)
1	1	0	$f_X/2^9$ (8.79 kHz)
1	1	1	$f_X/2^{11}$ (2.20 kHz)

Cautions 1. Before changing the data of TCL5n, be sure to stop the timer operation.

2. Be sure to set bits 3 to 7 to 0.

Remarks 1. In the cascade mode, the setting of bits TCL50 or TCL52 of the lower timer (TM50 or TM52) is valid, and the setting of bits TCL51 or TCL53 of the higher timer (TM51 or TM53) is invalid.

2. n = 0 to 2

3. f_X: System clock oscillation frequency

4. ( ): f_X = 4.5 MHz

(2) Timer clock select register 53 (TCL53)

This register selects the count clock of 8-bit timer counter 53 (TM53).

TCL53 is set with an 8-bit memory manipulation instruction.

Reset input clears TCL53 to 00H.

Figure 6-6. Format of Timer Clock Select Register 53 (TCL53)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL53	0	0	0	0	0	TCL532	TCL531	TCL530	FF77H	00H	R/W

TCL532	TCL531	TCL530	Count clock selection
0	0	0	Setting prohibited
0	0	1	Setting prohibited
0	1	0	$f_X/2$ (2.25 MHz)
0	1	1	$f_X/2^3$ (563 kHz)
1	0	0	$f_X/2^5$ (141 kHz)
1	0	1	$f_X/2^7$ (35.2 kHz)
1	1	0	$f_X/2^9$ (8.79 kHz)
1	1	1	$f_X/2^{11}$ (2.20 kHz)

Cautions 1. Before changing the data of TCL53, be sure to stop the timer operation.

2. Be sure to reset bits 3 to 7 to 0.

Remarks 1. In the cascade mode, the setting of bit TCL53 of the higher timer (TM53) is invalid.

2. f_X: System clock oscillation frequency

3. ( ): f_X = 4.5 MHz

(3) 8-bit timer mode control registers 50 to 52 (TMC50 to TMC52)

The TMC5n register is used for the following.

<1> Controlling count operation of 8-bit timer counter 5n (TM5n)
<2> Selecting operation mode of 8-bit timer counter 5n (TM5n)
<3> Selecting single mode or cascade mode
<4> Setting status of timer output F/F (flip-flop)
<5> Controlling timer F/F or selecting active level in PWM (free-running) mode
<6> Controlling timer output

TMC5n can be set with a 1-bit or 8-bit memory manipulation instruction.

Reset input clears TMC5n to 00H.

Remark n = 0 to 2

Figure 6-7. Format of 8-Bit Timer Mode Control Registers 50 to 52 (TMC50 to TMC52)

Symbol	<7>	6	5	4	<3>	<2>	1	<0>	Address	After reset	R/W
TMC50	TCE50	TMC506	0	TMC504	LVS50	LVR50	TMC501	TOE50	FF85H	00H	R/W
TMC51	TCE51	TMC516	0	TMC514	LVS51	LVR51	TMC511	TOE51	FF88H	00H	R/W
TMC52	TCE52	TMC526	0	TMC524	LVS52	LVR52	TMC521	TOE52	FF75H	00H	R/W

TCE5n	Control of count operation of TM5n
0	Clears counter to 0 and disables count operation (disables prescaler)
1	Starts count operation

TMC5n6	Selection of operating mode of TM5n
0	Mode of clearing and starting TM5n on match between TM5n and CR5n
1	PWM (free-running) mode

TMC5n4	Selection of single mode or cascade mode
0	Single mode
1**Note**	Cascade mode (connected to lower timer)

LVS5n	LVR5n	Setting status of timer output F/F
0	0	Not affected
0	1	Resets timer output F/F to 0
1	0	Sets timer output F/F to 1
1	1	Setting prohibited

TMC5n1	Other than PWM mode (TMC5n6 = 0)	PWM mode (TMC5n6 = 1)
	Control of timer F/F	Selection of active level
0	Disables inversion operation	High active
1	Enables inversion operation	Low active

TOE5n	Control of timer output
0	Disables output (port mode)
1	Enables output

Note Since the higher timer settings become valid, the lower timer TMC504/TMC524 settings become invalid.

Caution Be sure to reset bit 4 (TMC5n4) to 0.

Remarks 1. The PWM output becomes inactive when TCE5n = 0 in the PWM mode.

2. LVS5n and LVR5n are 0 when read after data has been set.

3. n = 0 to 2

(4) 8-bit timer mode control register 53 (TMC53)

The TMC53 register is used for the following.

<1> Controlling count operation of 8-bit timer counter 53 (TM53)
<2> Selecting single mode or cascade mode

TMC53 can be set with a 1-bit or 8-bit memory manipulation instruction.
Reset input clears TMC53 to 00H.

Figure 6-8. Format of 8-Bit Timer Mode Control Register 53 (TMC53)

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
TMC53	TCE53	0	0	TMC534	0	0	0	0	FF78H	00H	R/W

TCE53	Control of count operation of TM53
0	Clears counter to 0 and disables count operation (disables prescaler)
1	Starts count operation

TMC534	Selection of single mode or cascade mode
0	Single mode
1	Cascade mode (connected to lower timer (TM52))

6.4 Operations of 8-Bit Timer/Event Counters 50 to 53

6.4.1 Operation as interval timer (8-bit)

The 8-bit timer/event counter operates as an interval timer that repeatedly generates an interrupt request at the interval specified by the count value set in advance in 8-bit compare register 5n (CRn).

When the count value of 8-bit timer counter 5n (TM5n) matches the value set in CR5n, the value of TM5n is cleared to 0. TM5n continues counting and an interrupt request signal (INTTM5n) is generated.

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

For the operation if the value of the compare register is changed while the timer count operation, refer to **(2)** in **6.5 Notes on 8-Bit Timer/Event Counters 50 to 53**.

[Setting]

<1> Set each register.

- TCL5n: Select a count clock.
- CR5n: Compare value
- TMC5n: Select a mode in which TM5n is cleared and started on match between TM5n and CR5n (TMC5n = 0000×××0B: × = Don't care).

<2> The count operation is started when TEC5n is set to 1.

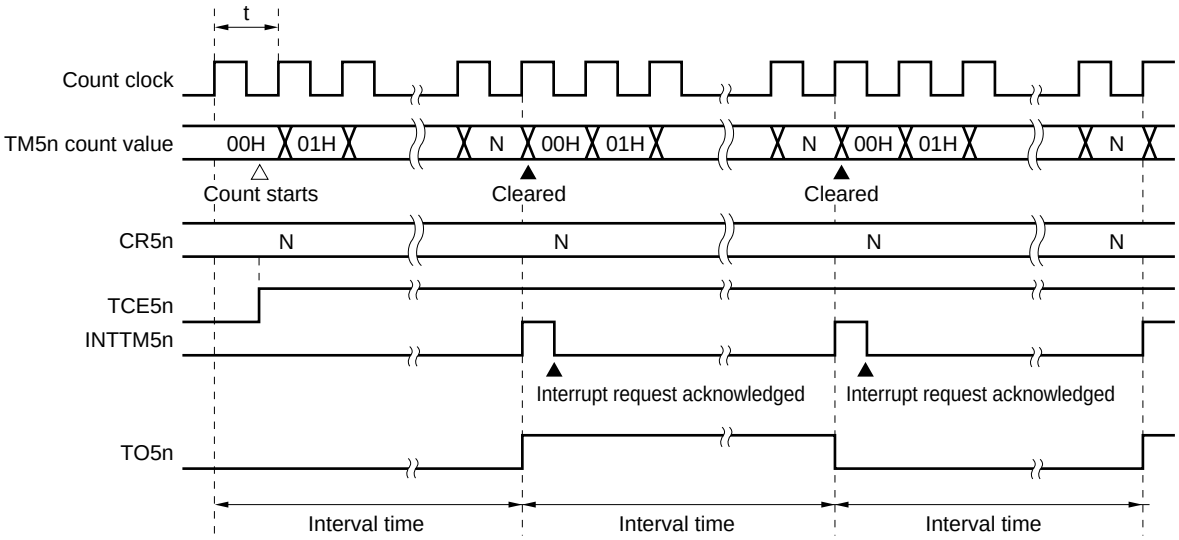
<3> INTTM5n is generated if the values of TM5n and CR5n match (TM5n is cleared to 00H).

<4> After that, INTTM5n is repeatedly generated at fixed intervals. To stop the count operation, clear TCE5n to 0.

Remark n = 0 to 3

Figure 6-9. Timing of Interval Timer Operation (1/3)

(a) Basic operation

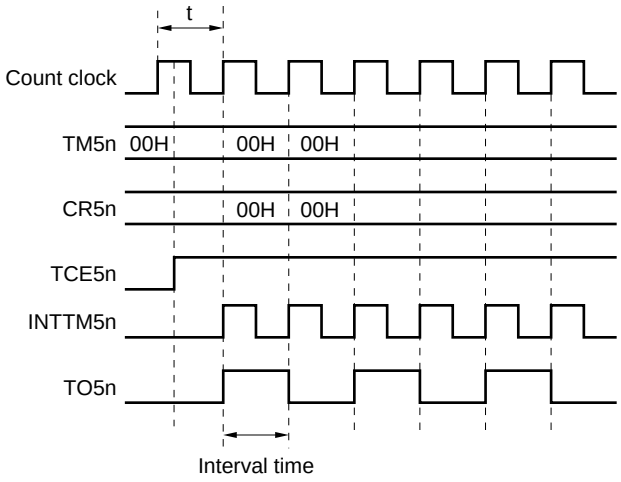


Remarks 1. Interval time = (N + 1) × t: N = 00H to FFH

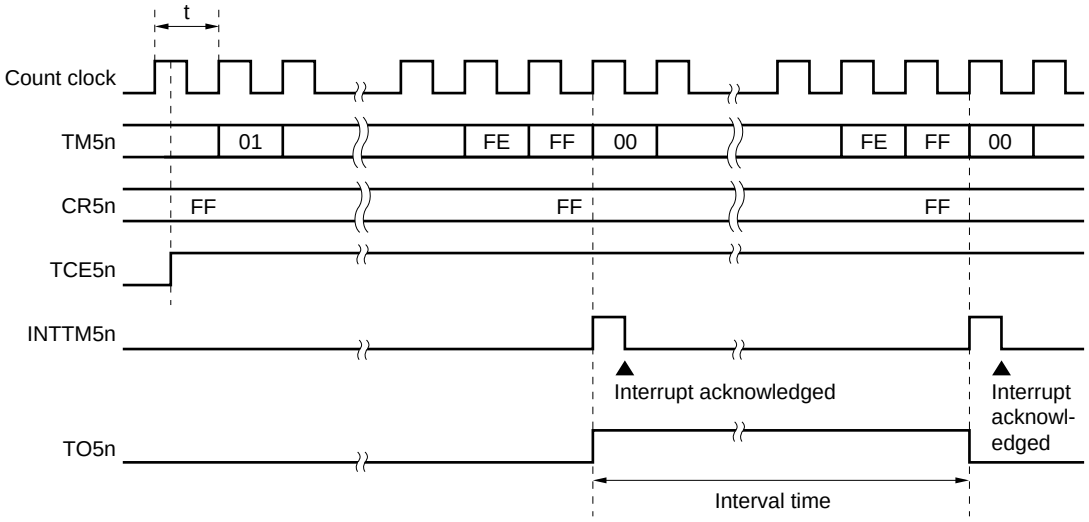
2. n = 0 to 3

Figure 6-9. Timing of Interval Timer Operation (2/3)

(b) When CR5n = 00H



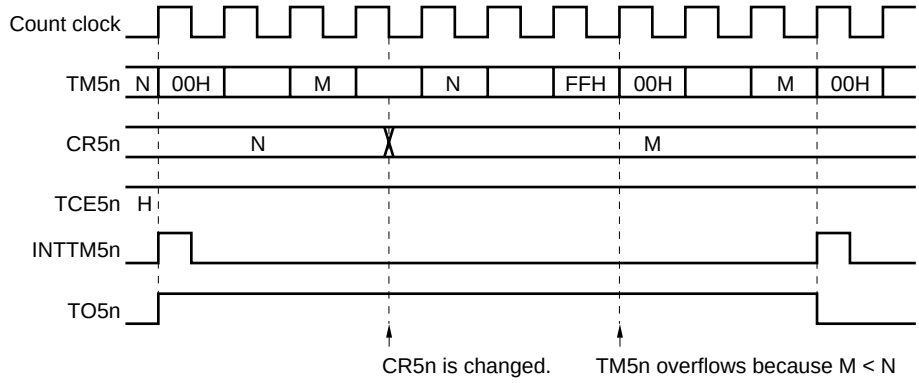
(c) When CR5n = FFH



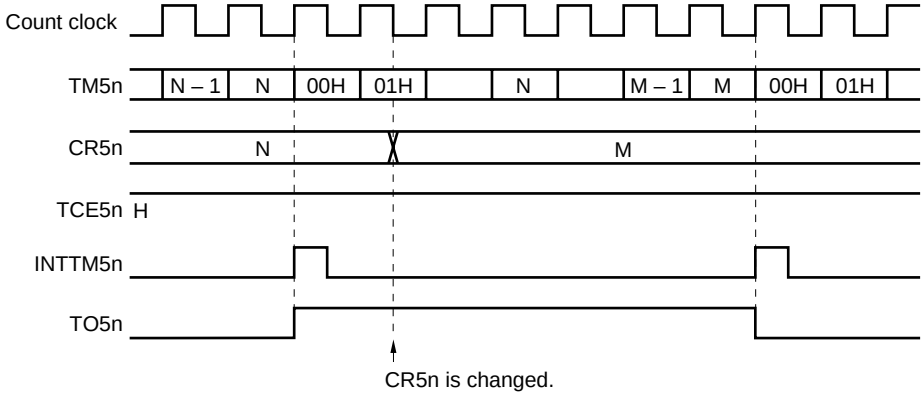
n = 0 to 3

Figure 6-9. Timing of Interval Timer Operation (3/3)

(d) Operation when CR5n is changed (M < N)



(e) Operation when CR5n is changed (M > N)



n = 0 to 3

6.4.2 Operation as external event counter (timers 50 to 52)

The external event counter counts the number of clock pulses input from an external source to the TI5n pin using 8-bit timer counter 5n (TM5n).

Each time the valid edge specified by timer clock select register 5n (TCL5n) has been input to TI5n, the value of TM5n is incremented. As the valid edge, either the rising or falling edge can be selected.

When the count value of TM5n matches the value of 8-bit compare register 5n (CR5n), TM5n is cleared to 0, and an interrupt request signal (INTTM5n) is generated.

After that, each time the value of TM5n matches the value of CR5n, INTTM5n is generated.

[Setting]

<1> Set each register.
- TCL5n: Select the valid edge of TI5n input.
- CR5n: Compare value
- TMC5n: Select a mode in which TM5n is cleared and started on match between TM5n and CR5n.

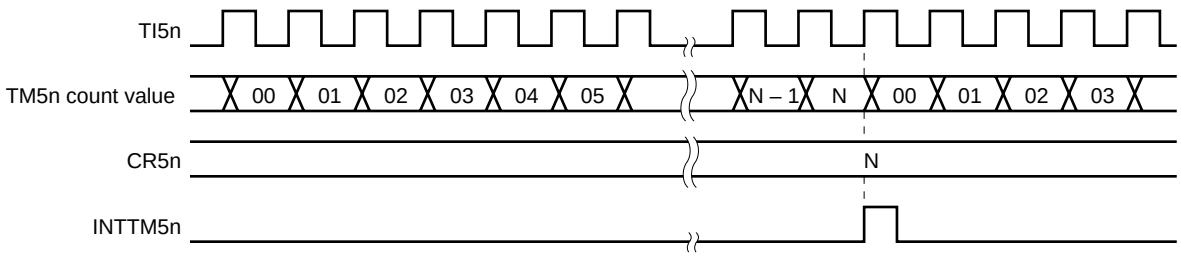
<2> The count operation is started when TEC5n is set to 1.

<3> INTTM5n is generated if the values of TM5n and CR5n match (TM5n is cleared to 00H).

<4> After that, INTTM5n is generated each time the value of TM5n matches the value of CR5n. To stop the count operation, clear TCE5n to 0.

Remark n = 0 to 2

Figure 6-10. Operation Timing of External Event Counter (with Rising Edge Specified)



n = 0 to 2

6.4.3 Square wave output operation (8-bit resolution) (timers 50 to 52)

8-bit timer/event counter TM5n can be used to output a square wave with any frequency at time interval specified by the value set in advance in 8-bit compare register 5n (CR5n).

When bit 0 (TOE5n) of 8-bit timer mode control register 5n (TMC5n) is set to 1, the output status of TO5n is inverted at the interval specified by the count value set in advance to CR5n. In this way, a square wave (duty factor = 50%) of any frequency can be output.

[Setting]

<1> Set each register.

- Reset the port latch and port mode register to “0”.
- TCL5n: Select a count clock.
- CR5n: Compare value
- TMC5n: Mode in which TM5n is cleared and started on match between TM5n and CR5n

LVS5n	LVR5n	Sets Status of Timer Output F/F
1	0	High-level output
0	1	Low-level output

Enable inverting the timer F/F.

Enable the timer output → TOE5n = 1.

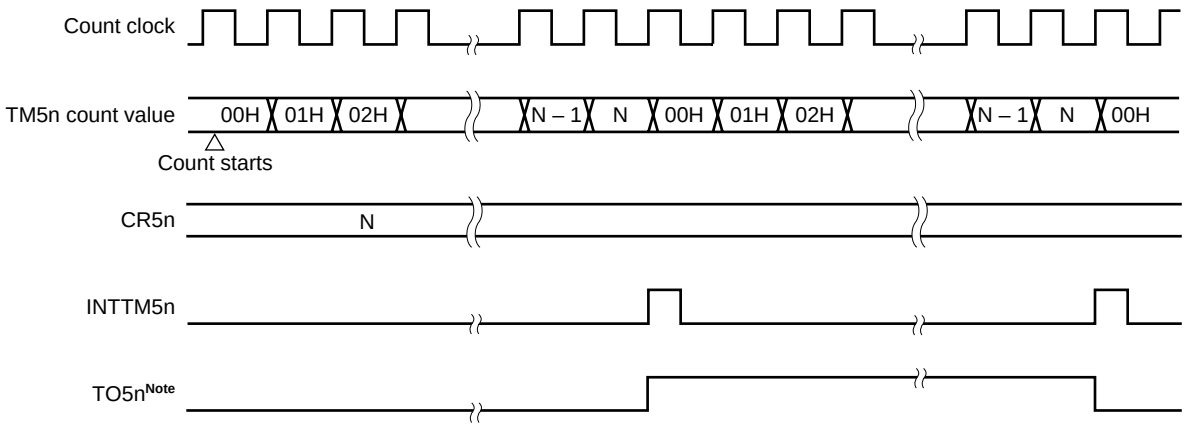
<2> When TCE5n is set to 1, the count operation is started.

<3> When the value of TM5n matches the value of CR5n, the timer output F/F is inverted.
In addition, INTTM5n is generated, and TM5n is cleared to 00H.

<4> After that, the timer output F/F is inverted at fixed intervals, and a square wave is output from TO5n.

Remark n = 0 to 2

Figure 6-11. Timing of Square Output Operation



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

Remark n = 0 to 2

6.4.4 8-bit PWM output operation (timers 50 to 52)

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

A pulse with a duty factor determined by the value set in 8-bit compare register 5n (CR5n) is output from TO5n. Set the active level width of the PWM pulse to CR5n. The active level is selected by bit 1 (TMC5n) of TMC5n. The count clock can be selected by bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock select register n (TCL5n). PWM output can be enabled or disabled by bit 0 (TOE5n) of TMC5n.

Caution The value of CR5n can be rewritten only once in one cycle in the PWM mode.

Remark n = 0 to 2

(1) Basic operation of PWM output

[Setting]

<1> Set port latches (P130 and P131) to 0.
<2> Select the active level width using the 8-bit compare register (CR5n).
<3> Select the count clock by using timer clock select register 5n (TCL5n).
<4> Select the active level using bit 1 (TMC5n1) of TMC5n.
<5> When bit 7 (TCE5n) of TMC5n is set to 1, the count operation is started.
To stop the count operation, reset TCE5n to 0.

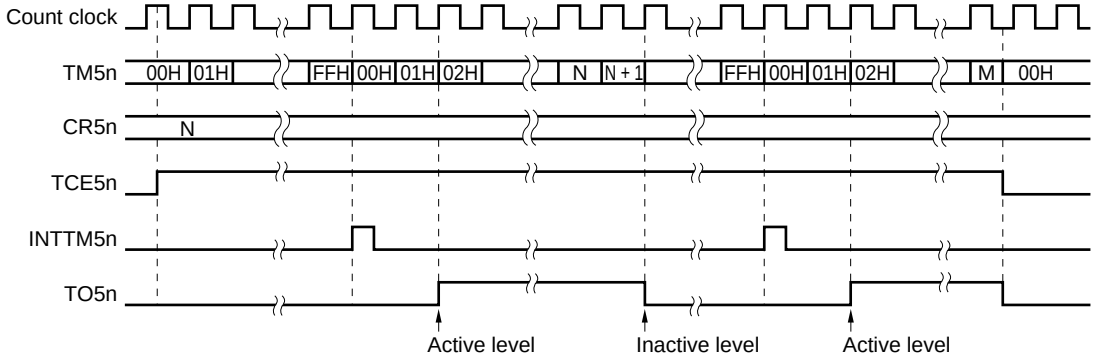
[Operation of PWM output]

<1> When the count operation is started, the PWM output (output from TO5n) remains inactive until an overflow occurs.
<2> When an overflow occurs, the active level set in step <1> above is output. This active level is output until the value of CR5n matches the count value of 8-bit timer counter 5n (TM5n).
<3> The PWM output remains inactive after CR5n and the count value of TM5n match, until an overflow occurs again.
<4> After that, <2> and <3> are repeated until the count operation is stopped.
<5> When the count operation is stopped because TCE5n is cleared to 0, PWM output becomes inactive.

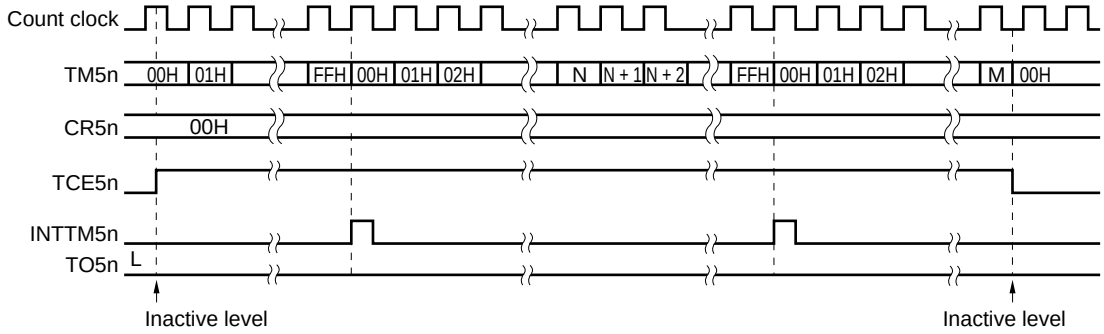
Remark n = 0 to 2

Figure 6-12. Operation Timing of PWM Output

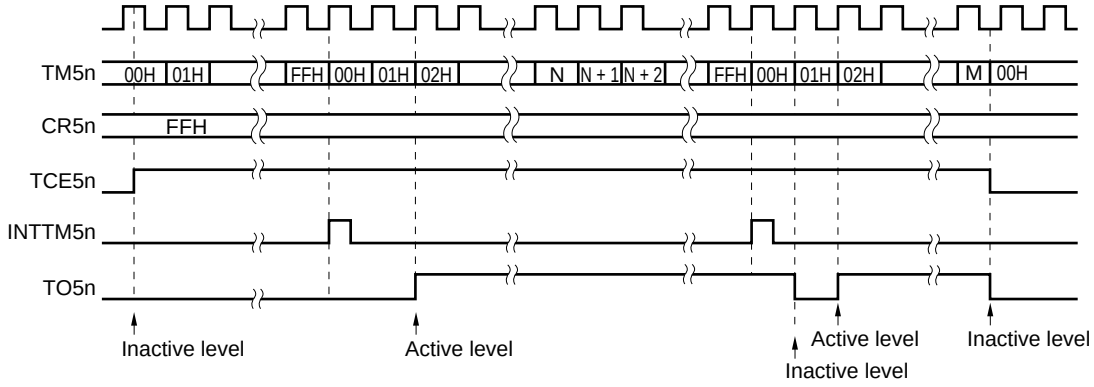
(a) Basic operation (when active level = H)



(b) When CR5n = 0



(c) When CR5n = FFH

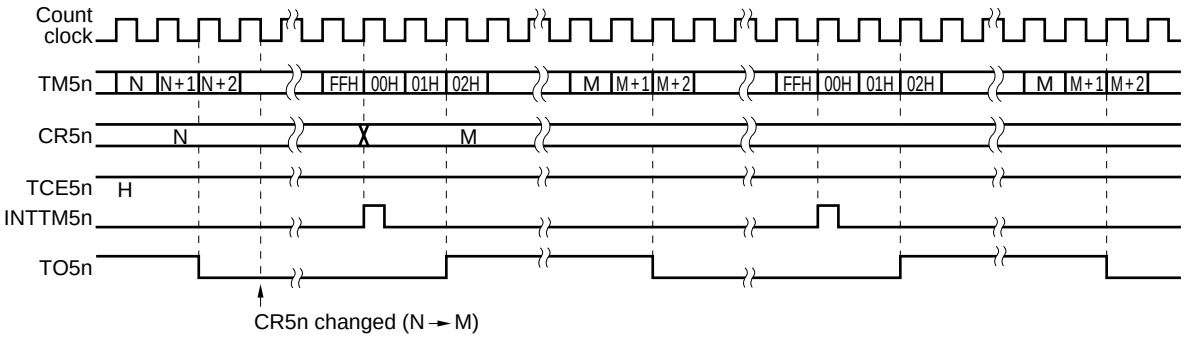


n = 0 to 2

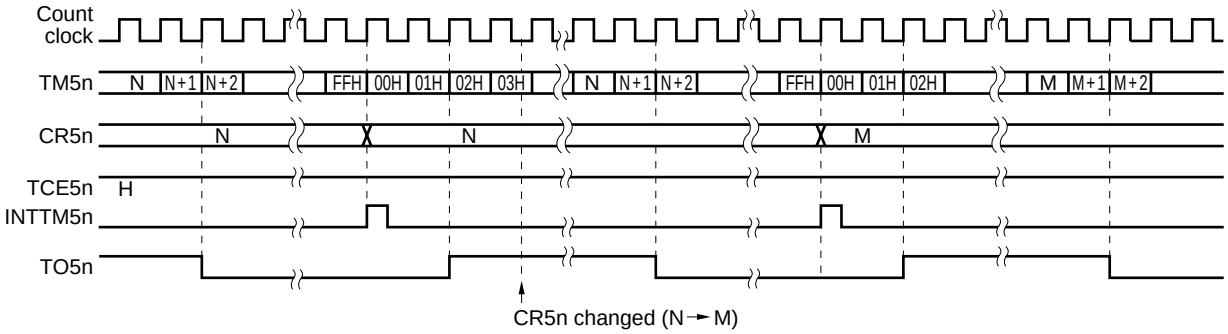
(2) Operation when CR5n is changed

Figure 6-13. Timing of Operation When CR5n Is Changed

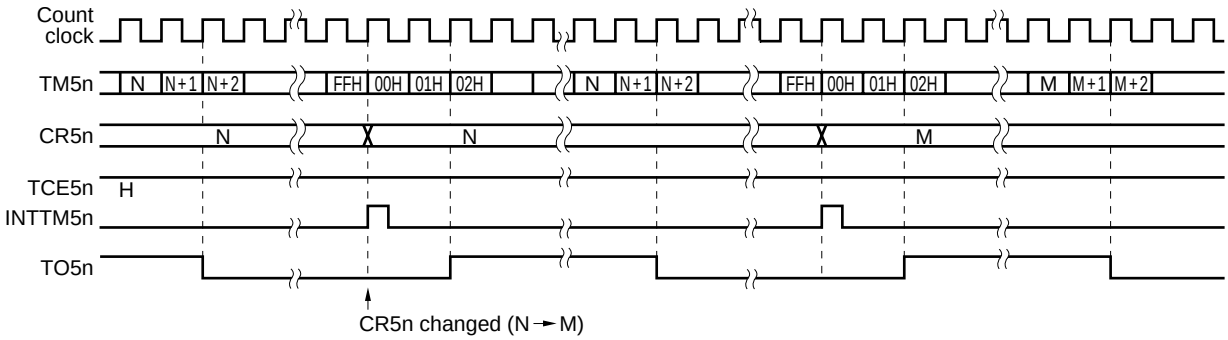
(a) If value of CR5n is changed from N to M before overflow of TM5n



(b) If value of CR5n is changed from N to M after overflow of TM5n



(c) If value of CR5n is changed from N to M for duration of 2 clocks immediately after overflow of TM5n



n = 0 to 2

Caution The value of CR5n can be changed only once in one cycle in the PWM mode.

6.4.5 Interval timer operation (16-bit)

When using the 8-bit timer/counters as a 16-bit timer, be sure to use a combination of timers 50 and 51 or timers 52 and 53. The following section describes the case when using timers 50 and 51. When using timers 52 and 53, read “50” as “52” and “51” as “53”.

The 8-bit timer/event counters are used together in 16-bit timer/counter mode when bit 4 (TMC514) of 8-bit timer mode control register 51 (TM51) is set to 1.

In this mode, the 8-bit timer/event counters are used as a 16-bit interval timer that repeatedly generates an interrupt request at intervals specified by the count value set in advance in the 8-bit compare registers (CR50 and CR51).

At this time, CR50 serves as the lower 8 bits of the 16-bit compare register, and CR51 serves as the higher 8 bits.

[Setting]

<1> Set each register.

- TCL50: Select the count clock for TM50.
  The count clock for TM51, which is cascaded, does not have to be set.
- CR50 and CR51: Compare values. (Each compare value can be set in a range of 00H to FFH.)
- TMC50 and TMC51: Select a mode in which the interval timer is cleared and started on match between TM50 and CR50 (or between TM51 and CR51).

  (TM50 → TMC50 = 0000×××0B ×: Don’t care
  TM51 → TMC51 = 0001×××0B ×: Don’t care)

<2> The count operation is started by setting TCE51 of TMC51 to 1 first, and then TCE50 of TMC50 to 1.

<3> If the value of cascaded timer TM50 matches the value of CR50, INTTM50 of TM50 is generated (TM50 and TM51 are cleared to 00H).

<4> After that, INTTM50 is repeatedly generated at fixed intervals.

Cautions 1. Be sure to set the compare registers (CR50 and CR51) after stopping the timer operation.

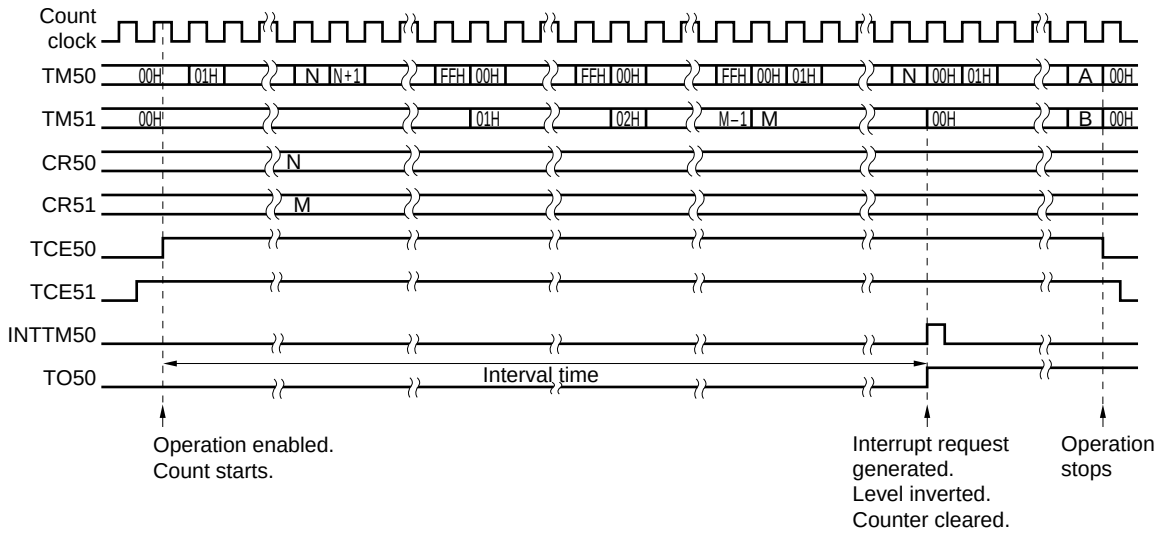
2. Even if the 8-bit timers/counters are cascaded, INTTM51 of TM51 is generated when the count value of TM51 matches CR51. Be sure to mask TM51 to disable this interrupt.

3. Set TCE50 and TCE51 in the order of TM51 and TM50.

4. Counting can be restarted or stopped by setting or resetting TCE50 of TM50 to 1 or 0.

Figure 6-14 shows a timing example in the 16-bit resolution cascade mode.

Figure 6-14. Operation Timing of 16-Bit Resolution Cascade Mode (Timers 50 and 51)

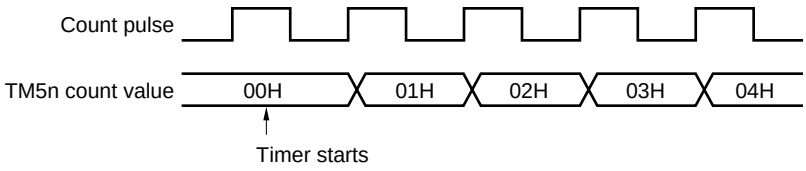


6.5 Notes on 8-Bit Timer/Event Counters 50 to 53

(1) Error on starting timer

An error of up to 1 clock occurs after the timer has been started until a match signal is generated. This is because 8-bit timer counter 5n (TM5n) is started asynchronously with the count pulse.

Figure 6-15. Start Timing of 8-Bit Timer Counter

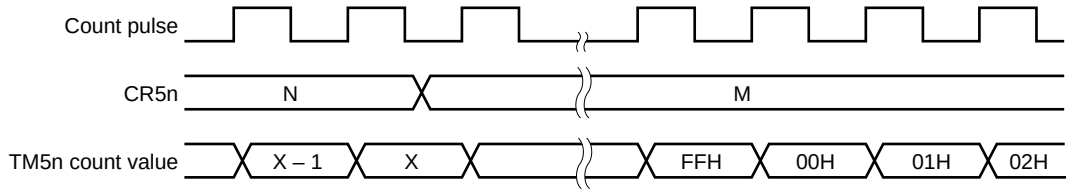


n = 0 to 3

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

If a new value of 8-bit compare register 5n (CR5n) is less than the value of 8-bit timer counter 5n (TM5n), counting continues, and TM5n overflows and starts counting from 0. If the new value of CR5n (M) is less than the old value (N), therefore, it is necessary to restart the timer after changing CR5n.

Figure 6-16. Timing After Changing Compare Register Value During Timer Count Operation



Caution Be sure to clear TCE5n to 0 to set the STOP status, except when TI5n input is selected.

Remarks 1. N > X > M

2. n = 0 to 3

(3) Reading TM5n (n = 0 to 3) during timer operation

When TM5n is read during operation, the count clock is temporarily stopped. Therefore, select a count clock with a high/low level longer than two cycles of the CPU clock. For example, when the CPU clock (f_{CPU}) is f_X, the count clock to be selected should be $f_X/4$ or less in order that TM5n can be read.

Remark n = 0 to 3

CHAPTER 7 BASIC TIMER

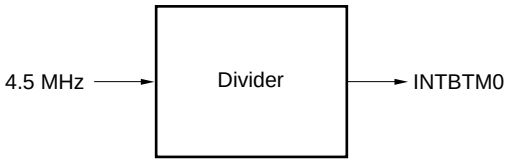
The basic timer is used for time management during program execution.

7.1 Function of Basic Timer

The basic timer generates an interrupt request signal (INTBTM0) at time intervals of 100 ms.

7.2 Configuration of Basic Timer

Figure 7-1. Block Diagram of Basic Timer



Caution Use the basic timer after setting bit 0 (DTSCK0) of the DTS system clock select register (DETSCK) to 1 after power application, and after reset by the $\overline{\text{RESET}}$ pin (refer to 5.1 Functions of Clock Generator).
The first interrupt request signal (INTBTM0) after the DTSCK0 flag has been set is generated within 100 to 140 ms. The second signal and those that follow are generated at intervals of 100 ms.

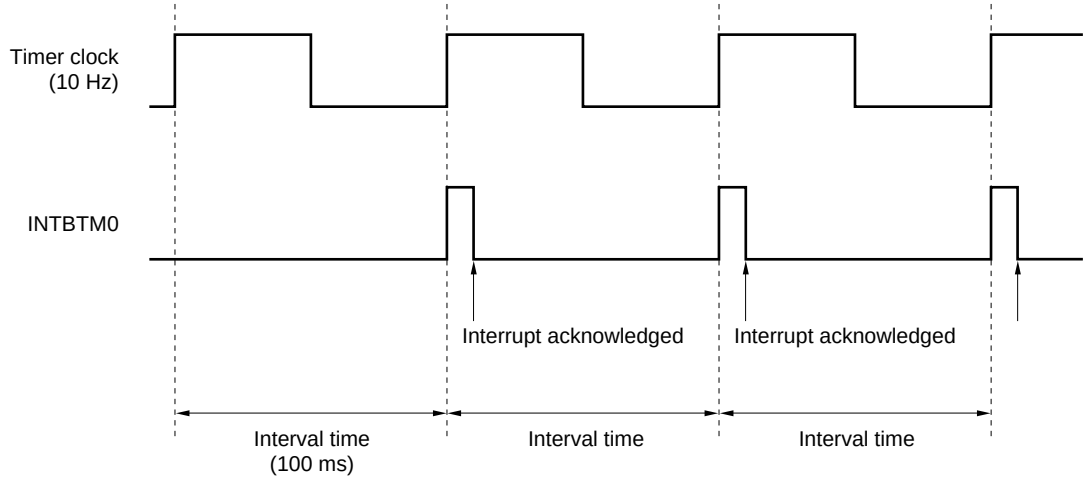
7.3 Operation of Basic Timer

An example of the operation of the basic timer is shown below.

In this example, the basic timer operates as an interval timer that repeatedly generates an interrupt at time intervals of 100 ms. Interrupt request signal (INTBTM0) is generated every 100 ms.

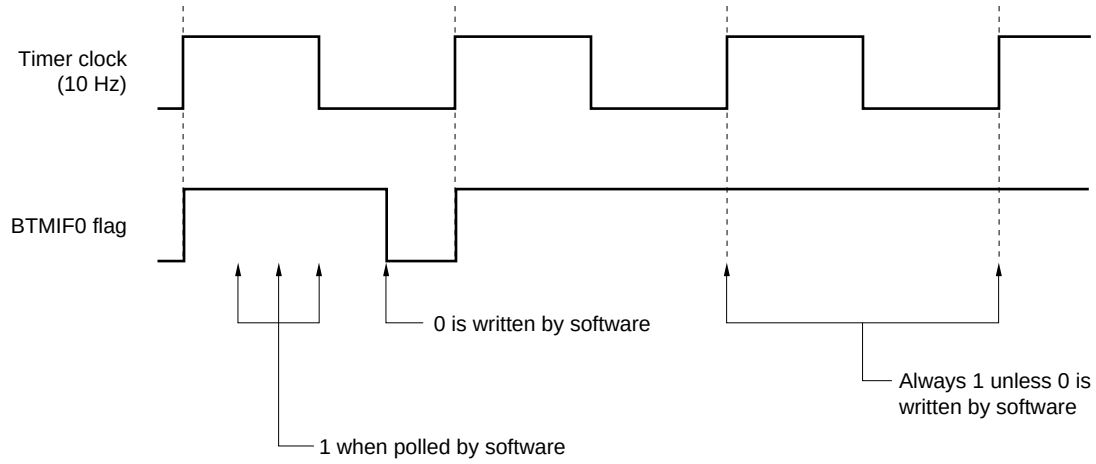
The timer clock frequency is 10 Hz.

Figure 7-2. Operation Timing of Basic Timer



By polling the interrupt request flag (BTMIF0) of this basic timer by software, time management can be carried out. Note that BTMIF0 is not a Read & Reset flag.

Figure 7-3. Operating Timing to Poll BTMIF0 Flag



For the registers controlling the basic timer, refer to **CHAPTER 12 INTERRUPT FUNCTIONS**.

CHAPTER 8 WATCHDOG TIMER

8.1 Functions of Watchdog Timer

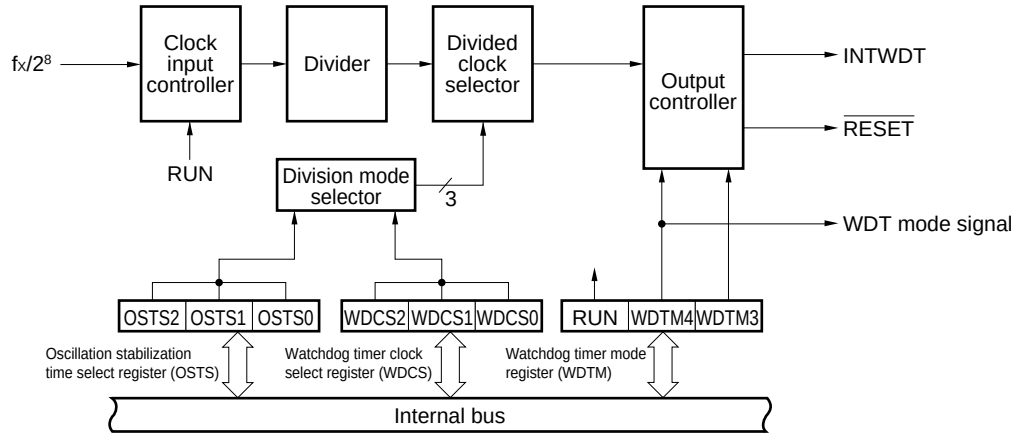
The watchdog timer has the following functions.

- Watchdog timer
- Interval timer
- Selecting oscillation stabilization time

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

Figure 8-1 shows a block diagram.

Figure 8-1. Block Diagram of Watchdog Timer



(1) Watchdog timer mode

An inadvertent program loop is detected. Upon detection of the inadvertent program loop, a non-maskable interrupt request or reset can be generated.

Table 8-1. Watchdog Timer Inadvertent Program Loop Detection Times

Inadvertent Program Loop Detection Time
$2^{12}/f_X$ (910 μs)
$2^{13}/f_X$ (1.82 ms)
$2^{14}/f_X$ (3.64 ms)
$2^{15}/f_X$ (7.28 ms)
$2^{16}/f_X$ (14.6 ms)
$2^{17}/f_X$ (29.1 ms)
$2^{18}/f_X$ (58.3 ms)
$2^{20}/f_X$ (233 ms)

Remarks 1. f_X: System clock oscillation frequency

2. ( ): f_X = 4.5 MHz

(2) Interval timer mode

Interrupt requests are generated at the preset time intervals.

Table 8-2. Interval Time

Interval Time
$2^{12}/f_X$ (910 μs)
$2^{13}/f_X$ (1.82 ms)
$2^{14}/f_X$ (3.64 ms)
$2^{15}/f_X$ (7.28 ms)
$2^{16}/f_X$ (14.6 ms)
$2^{17}/f_X$ (29.1 ms)
$2^{18}/f_X$ (58.3 ms)
$2^{20}/f_X$ (233 ms)

Remarks 1. f_X: System clock oscillation frequency

2. ( ): f_X = 4.5 MHz

8.2 Configuration of Watchdog Timer

The watchdog timer consists of the following hardware.

Table 8-3. Configuration of Watchdog Timer

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

8.3 Registers Controlling Watchdog Timer

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

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

(1) Watchdog timer clock select register (WDCS)

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

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

Reset input clears WDCS to 00H.

Figure 8-2. Format of Watchdog Timer Clock Select Register (WDCS)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
WDCS	0	0	0	0	0	WDCS2	WDCS1	WDCS0	FF42H	00H	R/W

WDCS2	WDCS1	WDCS0	Watchdog timer/interval timer overflow time
0	0	0	$2^{12}/f_X$ (910 μs)
0	0	1	$2^{13}/f_X$ (1.82 ms)
0	1	0	$2^{14}/f_X$ (3.64 ms)
0	1	1	$2^{15}/f_X$ (7.28 ms)
1	0	0	$2^{16}/f_X$ (14.6 ms)
1	0	1	$2^{17}/f_X$ (29.1 ms)
1	1	0	$2^{18}/f_X$ (58.3 ms)
1	1	1	$2^{20}/f_X$ (233 ms)

Remarks 1. f_X: System clock oscillation frequency

2. ( ): f_X = 4.5 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.

Reset input clears WDTM to 00H.

Figure 8-3. Format of Watchdog Timer Mode Register (WDTM)

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0	FFF9H	00H	R/W

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

WDTM4	WDTM3	Watchdog timer operating mode selection[Note 2]
0	×	Interval timer mode[Note 3] (Maskable interrupt occurs upon generation of an overflow.)
1	0	Watchdog timer mode 1 (Non-maskable interrupt occurs upon generation of an overflow.)
1	1	Watchdog timer mode 2 (Reset operation is activated upon generation of an overflow.)

Notes 1. Once set to 1, RUN cannot be cleared to 0 by software. Therefore, use $\overline{\text{RESET}}$ input to clear RUN to 0.

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

3. WDTM starts interval timer operation at a time RUN is set to 1.

Caution When RUN is set to 1 so that the watchdog timer is cleared, the actual overflow time is up to 0.5% shorter than the time set by the timer clock select register (WDCS).

Remark ×: Don't care

(3) Oscillation stabilization time select register (OSTS)

This register is used to select the time required for oscillation to stabilize after the $\overline{\text{RESET}}$ signal has been input or the STOP mode has been released.

★ This register is set with an 8-bit memory manipulation instruction.

Reset input sets OSTS to 04H. Therefore, it takes $2^{17}/f_X$ to release the STOP mode by $\overline{\text{RESET}}$ input.

Figure 8-4. Format of Oscillation Stabilization Time Select Register (OSTS)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFAH	04H	R/W

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time
0	0	0	$2^{12}/f_X$ (910 μs)
0	0	1	$2^{14}/f_X$ (3.64 ms)
0	1	0	$2^{15}/f_X$ (7.28 ms)
0	1	1	$2^{16}/f_X$ (14.6 ms)
1	0	0	$2^{17}/f_X$ (29.1 ms)
Other than above			Setting prohibited

Remarks 1. f_X: System clock oscillation frequency

2. ( ): f_X = 4.5 MHz

8.4 Operations of Watchdog Timer

8.4.1 Watchdog timer operation

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

The watchdog timer count clock (inadvertent program loop detection time interval) can be selected with bits 0 to 2 (WDCS0 to WDCS2) of timer clock select register 2 (WDCS). A watchdog timer count operation is started by setting bit 7 (RUN) of WDTM to 1. After the watchdog timer count operation starts, set RUN to 1 within the set inadvertent program loop time interval.

The watchdog timer can be cleared and counting started by setting RUN to 1. If RUN is not set to 1 and the inadvertent program loop detection time has elapsed, a system reset or a non-maskable interrupt request is generated according to the value of WDTM bit 3 (WDTM3).

The watchdog timer continues operating in the HALT mode but 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 inadvertent program loop detection time may be shorter than the set time by a maximum of 0.5%.

Table 8-4. Watchdog Timer Inadvertent Program Loop Detection Time

Inadvertent Program Loop Detection Time
$2^{12}/f_X$ (910 μs)
$2^{13}/f_X$ (1.82 ms)
$2^{14}/f_X$ (3.64 ms)
$2^{15}/f_X$ (7.28 ms)
$2^{16}/f_X$ (14.6 ms)
$2^{17}/f_X$ (29.1 ms)
$2^{18}/f_X$ (58.3 ms)
$2^{20}/f_X$ (233 ms)

Remarks 1. f_X: System clock oscillation frequency

2. ( ): f_X = 4.5 MHz

8.4.2 Interval timer operation

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

The count clock (interval time) can be selected by using bits 0 to 2 (WDCS0 to WDCS2) of the watchdog timer clock select register (WDCS). By setting bit 7 (RUN) of WDTM to 1, the watchdog timer starts operating as an interval timer.

When the watchdog timer operates as an interval timer, the interrupt mask flag (WDTMK) and priority specification flag (WDTPR) are validated and the maskable request interrupt (INTWDT) can be generated. Among maskable interrupt requests, the INTWDT default has the highest priority.

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

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

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

Table 8-5. Interval Timer Interval Time

Interval Time
$2^{12}/f_X$ (910 μs)
$2^{13}/f_X$ (1.82 ms)
$2^{14}/f_X$ (3.64 ms)
$2^{15}/f_X$ (7.28 ms)
$2^{16}/f_X$ (14.6 ms)
$2^{17}/f_X$ (29.1 ms)
$2^{18}/f_X$ (58.3 ms)
$2^{20}/f_X$ (233 ms)

Remarks 1. f_X: System clock oscillation frequency

2. ( ): f_X = 4.5 MHz

CHAPTER 9 BUZZER OUTPUT CONTROLLER

9.1 Functions of Buzzer Output Controllers

The μPD178054 Subseries has the following two types of buzzer output controllers.

- BEEP0
- BUZ

BEEP0 outputs a square wave of the buzzer frequency selected by BEEP clock select register 0 (BEEPCL0) from the BEEP0/P36 pin.

BUZ outputs a square wave of the buzzer frequency selected by the clock output select register (CKS) from the BUZ/P37 pin.

Figures 9-1 and 9-2 show the block diagrams of BEEP0 and BUZ.

Figure 9-1. Block Diagram of BEEP0

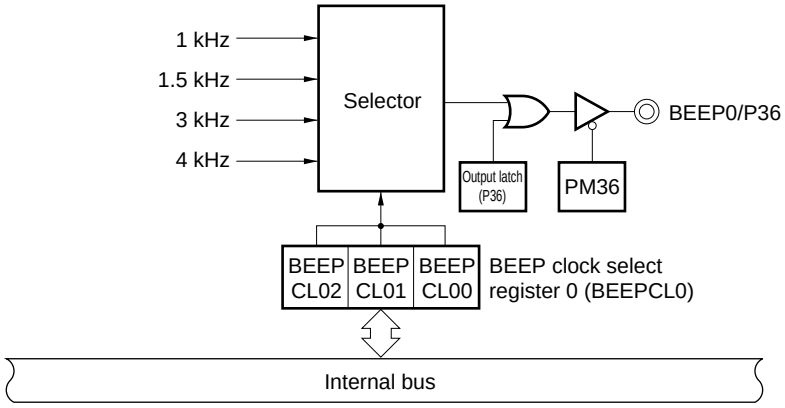
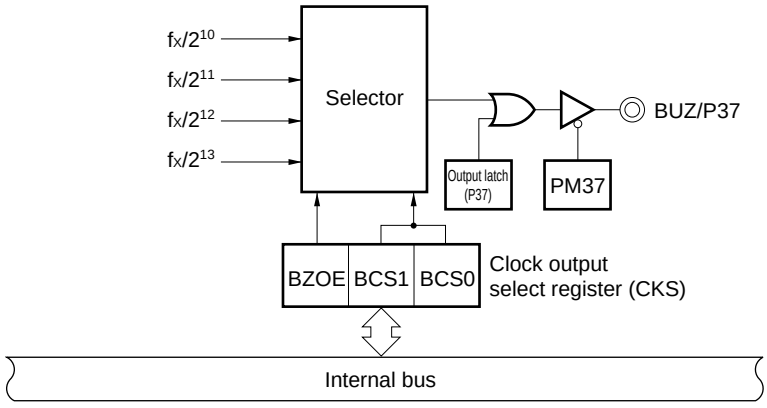


Figure 9-2. Block Diagram of BUZ



Remark f_X: System clock frequency

9.2 Configuration of Buzzer Output Controllers

The buzzer output controllers consist of the following hardware.

Table 9-1. Configuration of Buzzer Output Controllers

(1) BEEP0

Item	Configuration
Control register	BEEP clock select register 0 (BEEPCL0)

(2) BUZ

Item	Configuration
Control register	Clock output select register (CKS)

9.3 Registers Controlling Buzzer Output Controllers

9.3.1 BEEP0

BEEP0 is controlled by the following register.

- BEEP clock select register 0 (BEEPCL0)

(1) BEEP clock select register 0 (BEEPCL0)

This register selects the frequency of the buzzer output.

BEEPCL0 is set with a 1-bit or 8-bit memory manipulation instruction.

Reset input clears this register to 00H.

Figure 9-3. Format of BEEP Clock Select Register 0 (BEEPCL0)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BEEPCL0	0	0	0	0	0	BEEPCL02	BEEPCL01	BEEPCL00	FF41H	00H	R/W

BEEPCL02	BEEPCL01	BEEPCL00	Selection of frequency of BEEP0 output
0	×	×	Disables buzzer output (port function)
1	0	0	1 kHz
0	0	1	3 kHz
1	1	0	4 kHz
1	1	1	1.5 kHz

Caution The selected clock may not be correctly output during the period of 1 cycle immediately after the output clock has been changed.

9.3.2 BUZ

BUZ is controlled by the following register.

- Clock output select register (CKS)

(1) Clock output select register (CKS)

This register enables/disables buzzer output and sets the clock of the buzzer output.

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

Reset input clears this register to 00H.

Figure 9-4. Format of Clock Output Select Register (CKS)

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CKS	BZOE	BCS1	BCS0	0	0	0	0	0	FF40H	00H	R/W

BZOE	Enables/disables output of BUZ
0	Low-level output
1	Enables buzzer output

BCS1	BCS0	Selects output clock of BUZ
0	0	$f_X/2^{10}$ (4.39 kHz)
0	1	$f_X/2^{11}$ (2.20 kHz)
1	0	$f_X/2^{12}$ (1.10 kHz)
1	1	$f_X/2^{13}$ (549 Hz)

Remarks 1. f_X: System clock frequency

2. ( ): f_X = 4.5 MHz

9.4 Operation of Buzzer Output Controllers

The buzzer frequency is output by the following procedure.

(1) BEEP0

<1> Select a buzzer output frequency using bits 0 to 2 (BEEPCL00 to BEEPCL02) of BEEP clock select register 0 (BEEPCL0).

<2> Set the output latch of P36 to 0.

<3> Set bit 6 (PM36) of the port mode register 3 to 0 (set the output mode).

(2) BUZ

<1> Select a buzzer output frequency by using bits 5 and 6 (BCS0 and BCS1) of the clock output select register (CKS) (disable buzzer output).

<2> Set bit 7 (BZOE) of CKS to 1 and enable buzzer output.

<3> Set the output latch of P37 to 0.

<4> Set bit 7 (PM37) of the port mode register 3 to 0 (set output mode).

CHAPTER 10 A/D CONVERTER

10.1 Functions of A/D Converter

The A/D converter converts analog inputs into digital values and consists of 6 channels (ANI0 to ANI5) with an 8-bit resolution.

The conversion method is based on successive approximation and the conversion result is held in 8-bit A/D conversion result register 3 (ADCR3).

Conversion is started by setting A/D converter mode register 3.

Select one analog input channel from ANI0 to ANI5 and carry out A/D conversion.

When A/D conversion is complete, the next A/D conversion is started immediately. Each time an A/D conversion operation ends, an interrupt request (INTAD3) is generated.

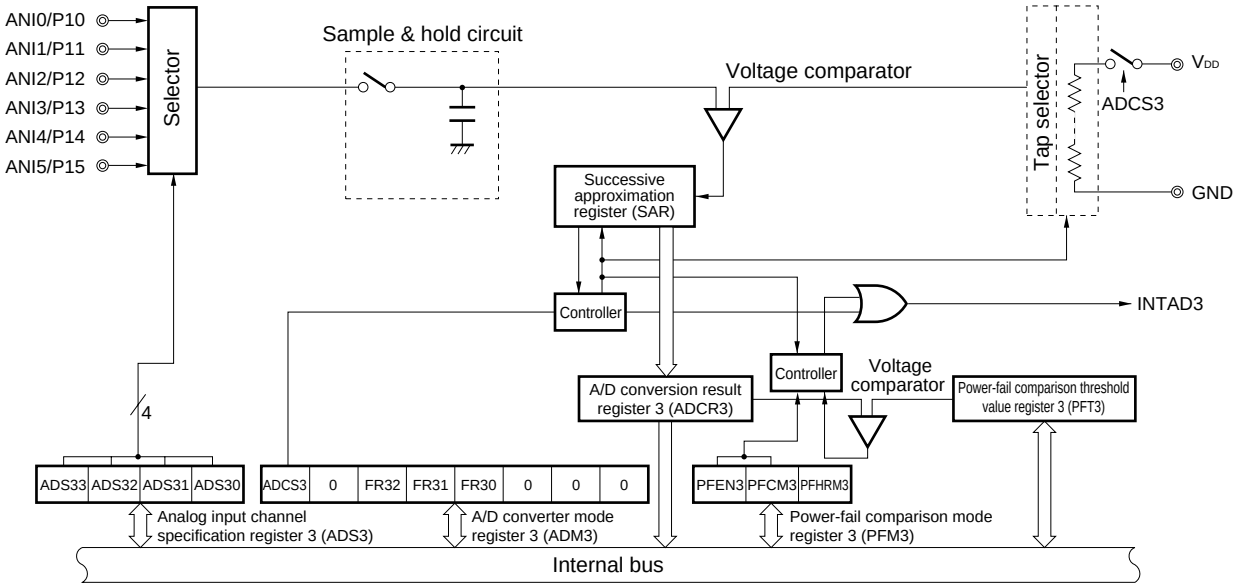
10.2 Configuration of A/D Converter

The A/D converter consists of the following hardware.

Table 10-1. Configuration of A/D Converter

Item	Configuration
Analog inputs	6 channels (ANI0 to ANI5)
Control registers	A/D converter mode register 3 (ADM3) Analog input channel specification register 3 (ADS3) Power-fail comparison mode register 3 (PFM3)
Registers	Successive approximation register (SAR) A/D conversion result register 3 (ADCR3) Power-fail comparison threshold value register 3 (PFT3)

Figure 10-1. Block Diagram of A/D Converter



(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 (termination of A/D conversion), the SAR contents are transferred to the A/D conversion result register.

(2) A/D conversion result register 3 (ADCR3)

This register is an 8-bit register to store the A/D conversion result. Each time A/D conversion terminates, the conversion result is loaded from the successive approximation register (SAR).

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

Reset input makes ADCR undefined.

Caution When data is written to A/D converter mode register 3 (ADM3) and analog input channel specification register 3 (ADS3), the contents of ADCR3 may be undefined. Read the result of conversion after conversion has been completed and before writing data to ADM3 and ADS3; otherwise the correct conversion result may not be read.

(3) Power-fail comparison threshold value register 3 (PFT3)

This register sets a threshold value to be compared with the value of A/D conversion result register 3 (ADCR3).

PFT3 is read or written with an 8-bit memory manipulation instruction.

(4) Sample & hold circuit

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

(5) Voltage comparator

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

(6) Resistor string

The resistor string is connected between V_{DD} and GND, and generates a voltage to be compared to the analog input.

(7) ANI0 to ANI5 pins

These are the 6-channel analog input pins through which analog signals to undergo A/D conversion are input to the A/D converter.

Cautions 1. Use the ANI0 to ANI5 input voltages within the specified range. If a voltage higher than V_{DD} or lower than GND is applied (even if within the absolute maximum ratings), the converted value of the corresponding channel becomes undefined and may adversely affect the converted values of other channels.

2. The analog input pins (ANI0 to ANI5) are also used as input port pins (P10 to P15). When one of ANI0 to ANI5 is selected for A/D conversion, do not execute an input instruction to port 1; otherwise the conversion resolution may drop.

If a digital pulse is applied to the pin adjacent to the pin executing A/D conversion, the A/D conversion value may not be obtained as expected due to coupling noise. Do not apply a pulse to the pin adjacent to the pin executing A/D conversion.

10.3 Registers Controlling A/D Converter

The following three registers control the A/D converter.

- A/D converter mode register 3 (ADM3)
- Analog input channel specification register 3 (ADS3)
- Power-fail comparison mode register 3 (PFM3)

(1) A/D converter mode register 3 (ADM3)

This register selects the conversion time of the analog input to be converted and starts or stops the conversion operation.

ADM3 is set with a 1-bit or 8-bit memory manipulation instruction.

Reset input clears this register to 00H.

Figure 10-2. Format of A/D Converter Mode Register 3 (ADM3)

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
ADM3	ADCS3	0	FR32	FR31	FR30	0	0	0	FF12H	00H	R/W

ADCS3	Control of A/D conversion operation
0	Stops conversion operation
1	Enables conversion operation

FR32	FR31	FR30	Selection of conversion time
0	0	0	288/f_X (64.0 μs)
0	0	1	240/f_X (53.3 μs)
0	1	0	192/f_X (42.7 μs)
1	0	0	144/f_X (32.0 μs)
1	0	1	120/f_X (26.7 μs)
1	1	0	96/f_X (21.3 μs)
Other than above			Setting prohibited

Cautions 1. The conversion result is undefined immediately after bit 7 (ADCS3) has been set to 1.

2. To change the data of bits 3 to 5 (FR30 to FR32), stop the A/D conversion operation.

Remarks 1. f_X: System clock oscillation frequency

2. ( ): f_X = 4.5 MHz

(2) Analog input channel specification register 3 (ADS3)

This register specifies the input channel of the analog voltage to be converted.

ADS3 is set with an 8-bit memory manipulation instruction.

Reset input clears this register to 00H.

Figure 10-3. Format of Analog Input Channel Specification Register 3 (ADS3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ADS3	0	0	0	0	ADS33	ADS32	ADS31	ADS30	FF13H	00H	R/W

ADS33	ADS32	ADS31	ADS30	Specification of analog input channel
0	0	0	0	ANI0
0	0	0	1	ANI1
0	0	1	0	ANI2
0	0	1	1	ANI3
0	1	0	0	ANI4
0	1	0	1	ANI5
Other than above				Setting prohibited

(3) Power-fail comparison mode register 3 (PFM3)

PFM3 is set with a 1-bit or 8-bit memory manipulation instruction.

Reset input clears this register to 00H.

Figure 10-4. Format of Power-Fail Comparison Mode Register 3 (PFM3)

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	After reset	R/W
PFM3	PFEN3	PFCM3	PFHRM3	0	0	0	0	0	FF16H	00H	R/W

PFEN3	Enable/disable of power-fail comparison
0	Disables power-fail comparison
1	Enables power-fail comparison

PFCM3	Selection of power-fail comparison mode
0	Generates interrupt request (INTAD) when ADCR3 ≥ PFT
1	Generates interrupt request (INTAD) when ADCR3 < PFT

Note PFHRM3	Selection of power-fail HALT repeat mode
0	Disables power-fail HALT repeat mode
1	Enables power-fail HALT repeat mode

Note When bit 5 (PFHRM3) is set to 1, power-fail comparison manipulation is enabled in the HALT mode in which A/D conversion is repeated until an interrupt request (INTAD3) is generated (this bit is reset to 0 when INTAD3 is generated).

10.4 Operations of A/D Converter

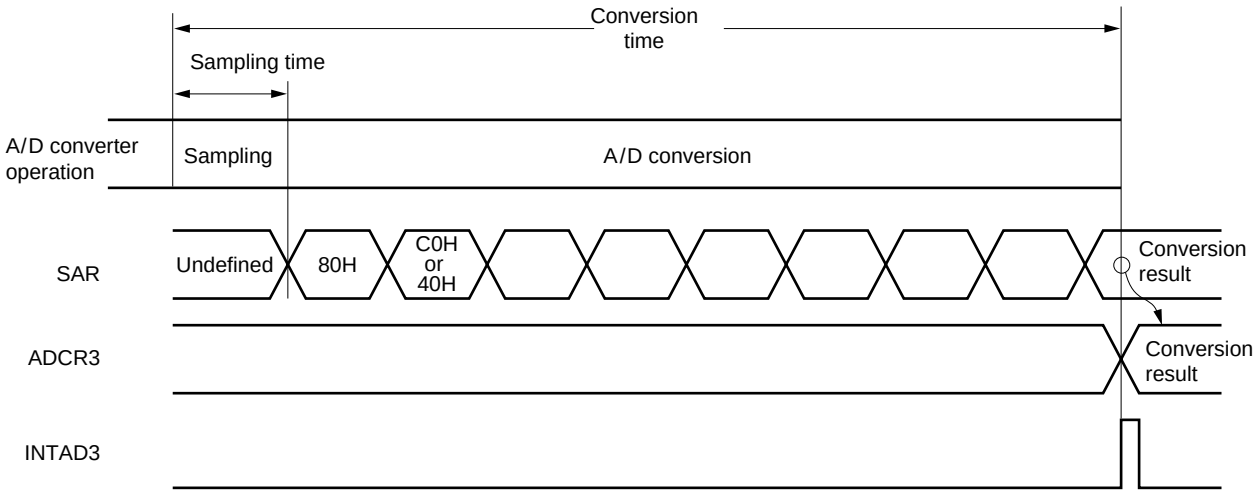
10.4.1 Basic operations of A/D converter

(1) Select one channel for A/D conversion with A/D converter analog input channel specification register 3 (ADS3).
(2) Sample the voltage input to the selected analog input channel with the sample & hold circuit.
(3) Sampling for the specified period of time sets the sample & hold circuit to the hold state so that the circuit holds the input analog voltage until termination of A/D conversion.
(4) Bit 7 of the successive approximation register (SAR) is set and the tap selector sets the series resistor string voltage tap to (1/2) V_{DD}.
(5) The voltage difference between the series resistor string voltage tap and analog input is compared with a voltage comparator. If the analog input is greater than (1/2) V_{DD}, the MSB of SAR remains set. If the input is smaller than (1/2) V_{DD}, the MSB is reset.
(6) Next, bit 6 of SAR is automatically set and the operation proceeds to the next comparison. In this case, the series resistor string voltage tap is selected according to the preset value of bit 7 as described below.
    - Bit 7 = 1: (3/4) V_{DD}
    - Bit 7 = 0: (1/4) V_{DD}

    The voltage tap and analog input voltage are compared and bit 6 of SAR is manipulated with the result as follows.
    - Analog input voltage ≥ Voltage tap: Bit 6 = 1
    - Analog input voltage < Voltage tap: Bit 6 = 0
(7) Comparison of this sort continues up to bit 0 of SAR.
(8) Upon completion of the comparison of 8 bits, any valid digital resultant value remains in SAR and the resultant value is transferred to and latched in A/D conversion result register 3 (ADCR3).
    At the same time, the A/D conversion termination interrupt request (INTAD3) can also be generated.

Caution The value immediately after A/D conversion has been started may not satisfy the ratings.

Figure 10-5. A/D Converter Basic Operation



A/D conversion operations are performed continuously until bit 7 (ADCS3) of the ADM is reset (0) by software.

If a write to ADM3 or ADS3 is performed during an A/D conversion operation, the conversion operation is initialized, and if the ADCS3 bit is set (1), conversion starts again from the beginning.

After reset input, the value of ADCR3 is undefined.

10.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI5) and the A/D conversion result (the value stored in A/D conversion result register 3 (ADCR3) is shown by the following expression.

$$ADCR3 = INT\left(\frac{V_{IN}}{V_{DD}} \times 256 + 0.5\right)$$

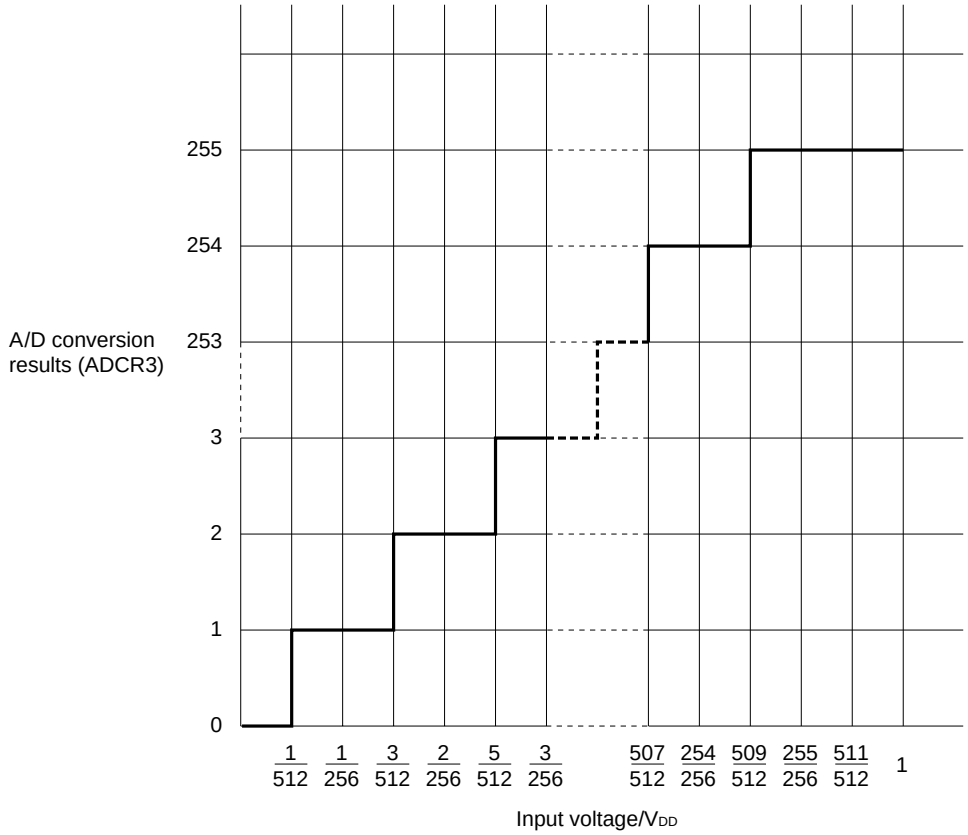
or

$$(ADCR3 - 0.5) \times \frac{V_{DD}}{256} \leq V_{IN} < (ADCR3 + 0.5) \times \frac{V_{DD}}{256}$$

Remark INT( ): Function which returns integer parts of value in parentheses.
V_{IN}: Analog input voltage
V_{DD}: V_{DD} pin voltage
ADCR3: A/D conversion result register 3 (ADCR3) value

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

Figure 10-6. Relationship Between Analog Input Voltage and A/D Conversion Result



10.4.3 A/D converter operating mode

The A/D converter has the following two modes:

- A/D conversion operation mode: In this mode, the voltage applied to the analog input pin selected from ANI0 to ANI5 is converted into a digital signal. The result of the A/D conversion is stored in A/D conversion result register 3 (ADCR3), and at the same time, an interrupt request signal (INTAD3) is generated.
- Power-fail comparison mode: The digital value resulting from A/D conversion is compared with the value assigned to power-fail comparison threshold value register 3 (PFT3) is compared. If the result of the comparison matches the condition set by bit 6 (PFCM3) of power-fail comparison mode register 3 (PFM3), an interrupt request signal (INTAD3) is generated.

(1) A/D conversion operation mode

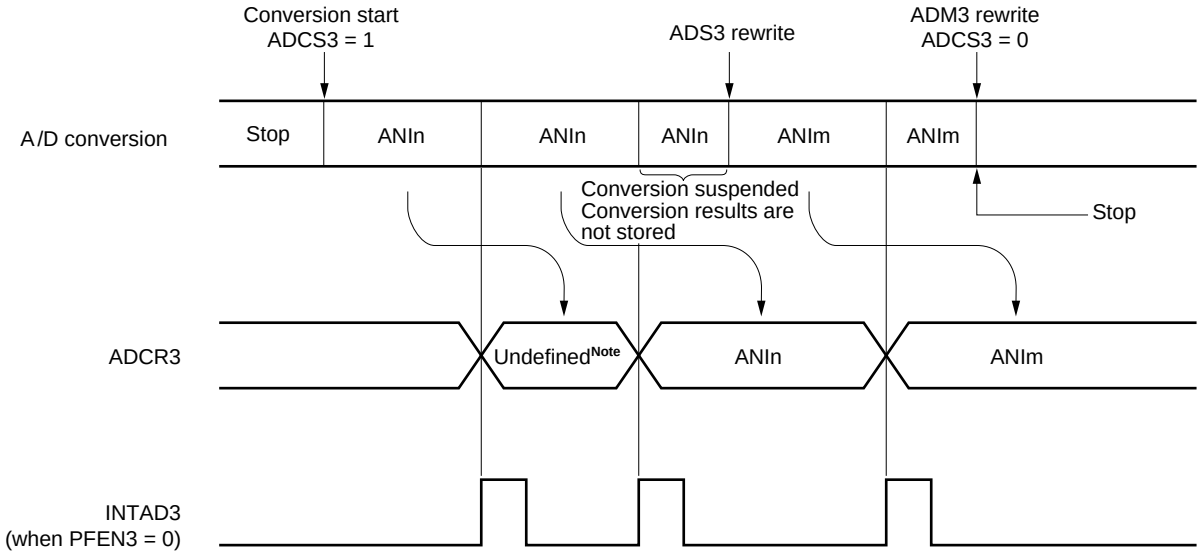
When bit 7 (ADCS3) of A/D converter mode register 3 (ADM3) is set to 1, the A/D conversion starts on the voltage applied to the analog input pins specified with bits 0 to 3 (ADS30 to ADS33) of ADS3.

Upon termination of the A/D conversion, the conversion result is stored in A/D conversion result register 3 (ADCR3) and the interrupt request signal (INTAD3) is generated. After one A/D conversion operation is started and terminated, the next A/D conversion operation starts immediately. The A/D conversion operation continues repeatedly until new data is written to ADM3.

If data is written to ADCS3 again during A/D conversion, the converter suspends its A/D conversion operation and starts A/D conversion on the newly written data.

If data with ADCS3 set to 0 is written to ADM3 during A/D conversion, the A/D conversion operation stops immediately.

Figure 10-7. A/D Conversion Operation



Remarks 1. n = 0, 1, ... , 5

2. m = 0, 1, ... , 5

Note The conversion result is illegal immediately after bit 7 (ADCS3) of A/D converter mode register 3 (ADM3) has been set to 1 (to enable conversion).

Caution Reset bit 5 (PFHRM3) of power-fail comparison mode register 3 (PFM3) to 0.

(2) Power-fail comparison mode

In the power-fail comparison mode, the digital value converted from analog input is compared in units of 8 bits.

If the result of the comparison matches the condition set by bit 6 (PFCM3) of power-fail comparison mode register 3 (PFM3), an interrupt request (INTAD3) is generated.

Moreover, the power-fail comparison mode can be used in the HALT mode. At this time, the HALT mode can be released by generating the interrupt request signal (INTAD3) as a result of comparison (however, the A/D operation must be executed before the HALT instruction is executed).

To set the power-fail comparison mode, set bit 7 (PEEN3) of PFM3 to 1, set bit 6 (PFCM3) to the generation condition of INTAD, and assign the threshold value to be compared with the value of A/D conversion result register 3 (ADCR3) to power-fail comparison threshold value register 3 (PFT3).

By setting bit 7 (ADCS3) of A/D converter mode register 3 (ADM3) to 1, the voltage applied to the analog input pin specified by ADS3 is converted into a digital signal. When the A/D conversion has been completed, the result of the conversion is stored in ADCR3. This conversion result is compared with the value set in PFT3 and if the result of the comparison matches the condition set by bit 6 (PFCM3) of PFM3, an interrupt request signal (INTAD3) is generated.

Figure 10-8. Power-Fail Comparison Threshold Value Register 3 (PFT3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PFT3	PFT37	PFT36	PFT35	PFT34	PFT33	PFT32	PFT31	PFT30	FF15H	00H	R/W

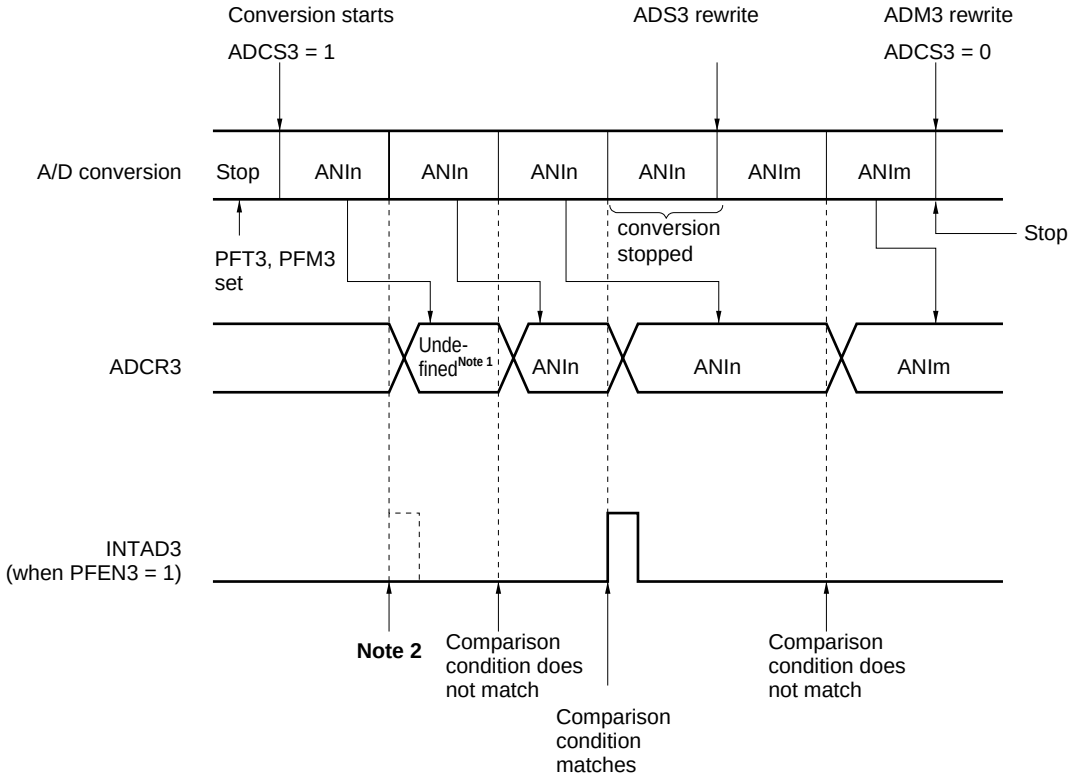
Remark Bit 7 (PFT37) is the MSB, and bit 0 (PFT30) is the LSB.
For the setting value, refer to **10.4.2 Input voltage and conversion results**.

Cautions

1. **In the power-fail comparison mode, the first result (A/D conversion result and interrupt request (INTAD)) of the A/D conversion (started by setting bit 7 (ADCS3) of A/D converter mode register 3 (ADM3) to 1) is not correct.**
2. **When executing A/D conversion in the HALT mode using the power-fail HALT repeat mode, clear the interrupt request flag (ADIF) after the first conversion has been completed immediately after bit 7 (ADCS3) of ADM3 has been set to 1, and bit 5 (PFHRM3) of power-fail comparison mode register 3 (PFM3) has been set to 1, before executing the HALT instruction.**
3. **To set the power-fail comparison mode in the HALT mode, be sure to set bit 5 (PFHRM3) of PFM3 to 1 before executing the HALT instruction; otherwise comparison cannot be performed correctly because the conversion result in the HALT mode is not stored in A/D conversion result register 3 (ADCR3). If bit 5 (PFHRM3) of PFM3 is set in the normal operating mode (other than HALT mode), the A/D conversion is not performed correctly. Therefore, be sure to clear this bit to 0 in the normal mode.**

Figure 10-9. A/D Conversion Operation in Power-Fail Comparison Mode (1/3)

(1) In normal mode (other than HALT mode)



Notes 1. The conversion data is undefined immediately after bit 7 (ADCS3) of A/D converter mode register 3 (ADM3) is set to 1 (to start conversion).

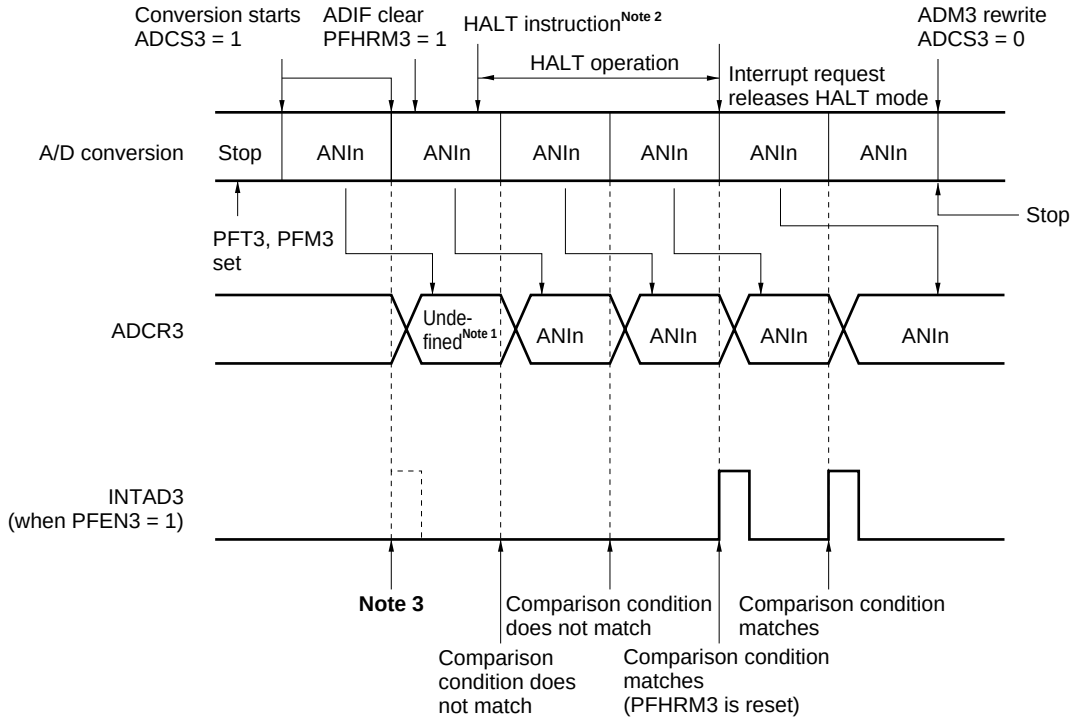
2. The first result of the A/D conversion (A/D conversion result and interrupt request) is not correct. Do not use this result because there is a possibility that it will be determined that the comparison condition has matched even if it has not.

Caution Set power-fail comparison threshold value register 3 (PFT3) and power-fail comparison mode register 3 (PFM3) before starting conversion. Be sure to reset bit 5 (PFHRM3) of PFM3 to 0 (to disable HALT repeat mode setting).

Remark n = 0, 1, ... 5
m = 0, 1, ... 5

Figure 10-9. A/D Conversion Operation in Power-Fail Comparison Mode (2/3)

(2) In HALT repeat mode (when generation of interrupt (INTAD3) is used to release HALT mode)



Notes 1. The conversion data is undefined immediately after bit 7 (ADCS3) of A/D converter mode register 3 (ADM3) is set to 1 (to start conversion).

2. When executing A/D conversion in the HALT mode by using the power-fail comparison mode, clear the interrupt request flag (ADIF) after the first conversion has been completed immediately after bit 7 (ADCS3) of ADM3 has been set to 1, and bit 5 (PFHRM3) of power-fail comparison mode register 3 (PFM3) has been set to 1, before executing the HALT instruction.

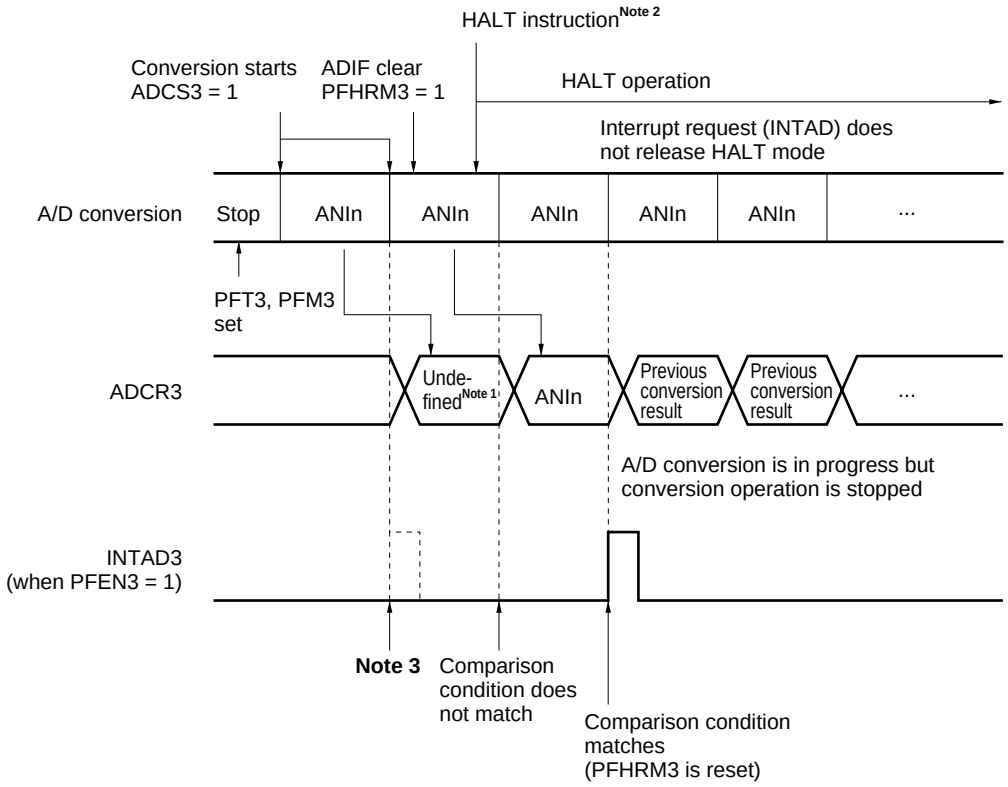
3. The first result of the A/D conversion (A/D conversion result and interrupt request) is not correct. Do not use this result because there is a possibility that it will be determined that the comparison condition has matched even if it has not.

Caution Be sure to set bit 5 (PFHRM3) of PFM3 to 1 (to enable the HALT repeat mode setting).

Remark n = 0, 1, ... 5

Figure 10-9. A/D Conversion Operation in Power-Fail Comparison Mode (3/3)

(3) In HALT repeat mode (when generation of interrupt (INTAD3) is not used to release HALT mode)



Notes 1. The conversion data is undefined immediately after bit 7 (ADCS3) of A/D converter mode register 3 (ADM3) is set to 1 (to start conversion).

2. When executing A/D conversion in the HALT mode by using the power-fail HALT repeat mode, clear the interrupt request flag (ADIF) after the first conversion has been completed immediately after bit 7 (ADCS3) of ADM3 has been set to 1, and bit 5 (PFHRM3) of power-fail comparison mode register 3 (PFM3) has been set to 1, before executing the HALT instruction.

3. The first result of the A/D conversion (A/D conversion result and interrupt request) is not correct. Do not use this result because there is a possibility that it will be determined that the comparison condition has matched even if it has not.

Caution Be sure to set bit 5 (PFHRM3) of PFM3 to 1 (to enable the HALT repeat mode setting).

Remark n = 0, 1, ... 5

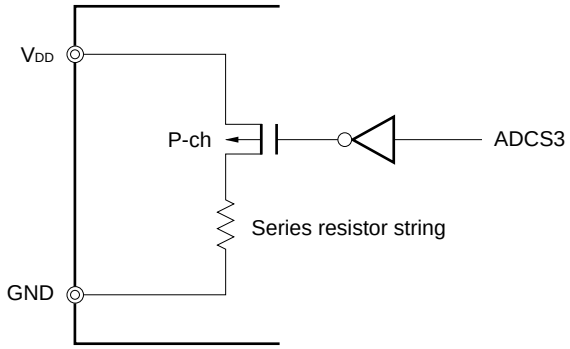
10.5 Notes on A/D Converter

(1) Current consumption in standby mode

The A/D converter is stopped in the standby mode. At this time, the current consumption can be reduced by stopping the conversion operation (by resetting bit 7 (ADCS3) of A/D converter mode register 3 (ADM3) to 0).

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

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



(2) Input range of ANI0 to ANI5

The input voltages of ANI0 to ANI5 should be within the specified range. In particular, if a voltage above V_{DD} or below GND is input (even if within the absolute maximum rating range), the conversion value for that channel will be undefined. The conversion values of the other channels may also be affected.

(3) Conflicting operations

<1> Conflict between writing A/D conversion result register 3 (ADCR3) on completion of conversion and reading ADCR3 by an instruction

Reading ADCR3 takes precedence. After ADCR3 has been read, a new conversion result is written to ADCR3.

<2> Conflict between writing ADCR3 on completion of conversion and writing A/D converter mode register 3 (ADM3) or writing analog input channel specification register 3 (ADS3)

Writing ADM3 or ADS3 takes precedence. ADCR3 is not written. Nor is the conversion completion interrupt request signal (INTAD3) generated.

(4) ANI0 to ANI5

The analog input pins ANI0 to ANI5 also function as input port (P10 to P15) pins.

When A/D conversion is performed with any of pins ANI0 to ANI5 selected, be sure not to execute a PORT1 input instruction while conversion is in progress, as this may reduce the conversion resolution.

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

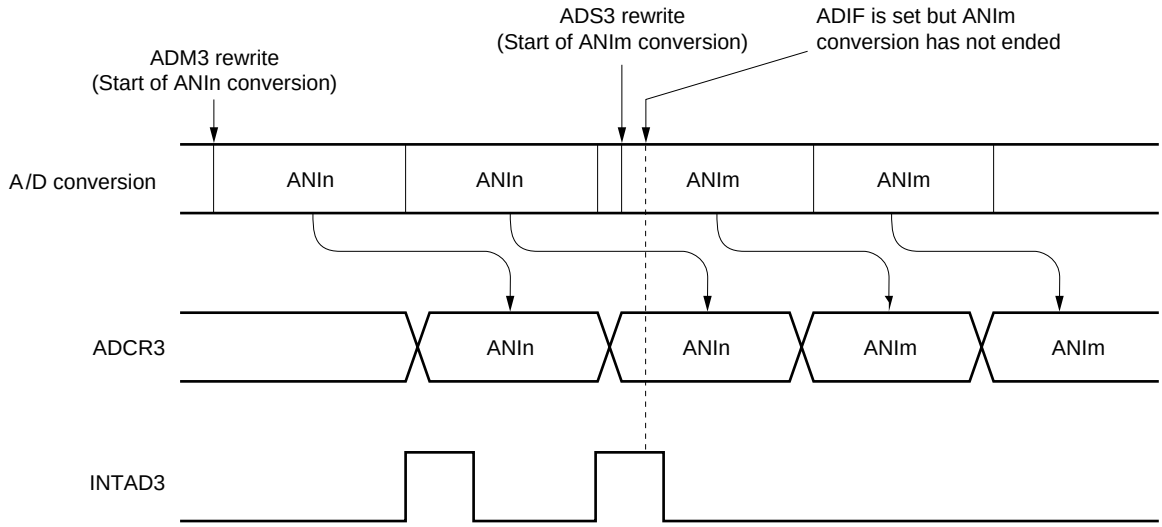
(5) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if analog input channel specification register 3 (ADS3) is changed.

Caution is therefore required since, if an analog input pin is changed during A/D conversion, the A/D conversion result and conversion end interrupt request flag for the pre-change analog input may be set just before the ADS3 rewrite, and when ADIF is read immediately after the ADM rewrite, ADIF may be set despite the fact that the A/D conversion for the post-change analog input has not ended.

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

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



Remarks 1. n = 0, 1, ..., 5

2. m = 0, 1, ..., 5

(6) Conversion result immediately after starting A/D conversion

The first A/D conversion result value is undefined immediately after the A/D conversion operation has been started. Poll the A/D conversion completion interrupt request (INTAD3) and discard the first conversion result.

(7) Reading A/D conversion result register 3 (ADCR3)

If data is written to A/D converter mode register 3 (ADM3) and analog input channel specification register 3 (ADS3), the contents of ADCR3 can be undefined. Read the conversion value before writing ADM3 and ADS3 after the conversion operation has been completed; otherwise the correct conversion result may not be read.

11.1 Functions of Serial Interfaces SIO30 to SIO32

The serial interface SIO3n has the following two modes.

(1) Operation stop mode

This mode is used when serial transfer is not performed. For details, refer to **11.4.1 Operation stop mode**.

(2) 3-wire serial I/O mode (MSB first)

In this mode, 8-bit data is transferred by using three lines: serial clock ($\overline{\text{SCK3n}}$), serial output (SO3n), and serial input (SI3n) lines.

Because transmission and reception can be executed simultaneously in this mode, the processing time of data transfer can be shortened.

The first bit of the 8-bit data to be transferred is the MSB.

The 3-wire serial I/O mode is useful for connecting a peripheral I/O or display controller with a clocked serial interface. For details, refer to **11.4.2 3-wire serial I/O mode**.

Figures 11-1 to 11-3 show the block diagrams of the serial interface SIO3n.

Remark n = 0 to 2

Figure 11-1. Block Diagram of Serial Interface SIO30

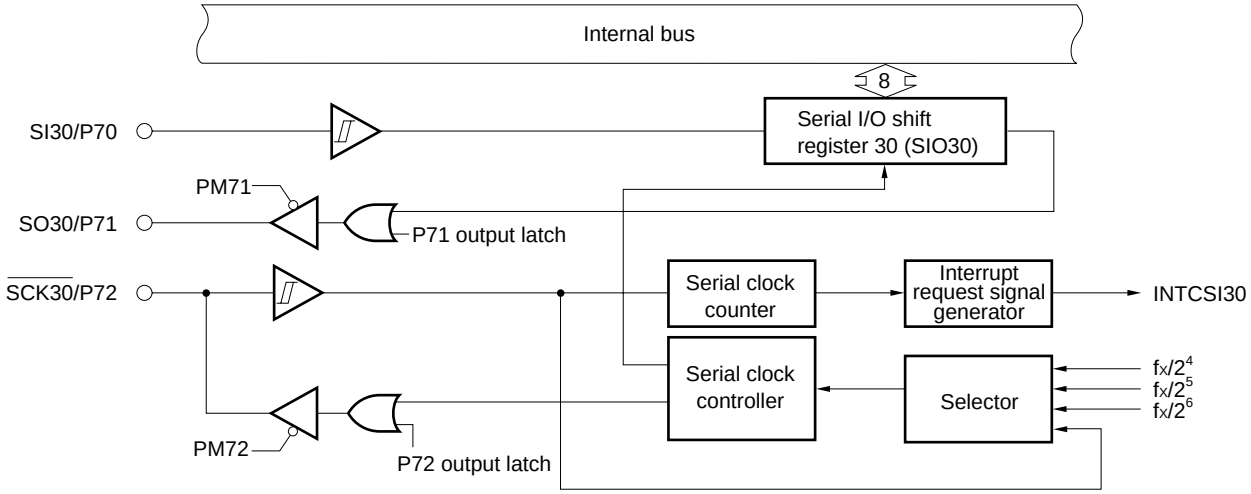


Figure 11-2. Block Diagram of Serial Interface SIO31

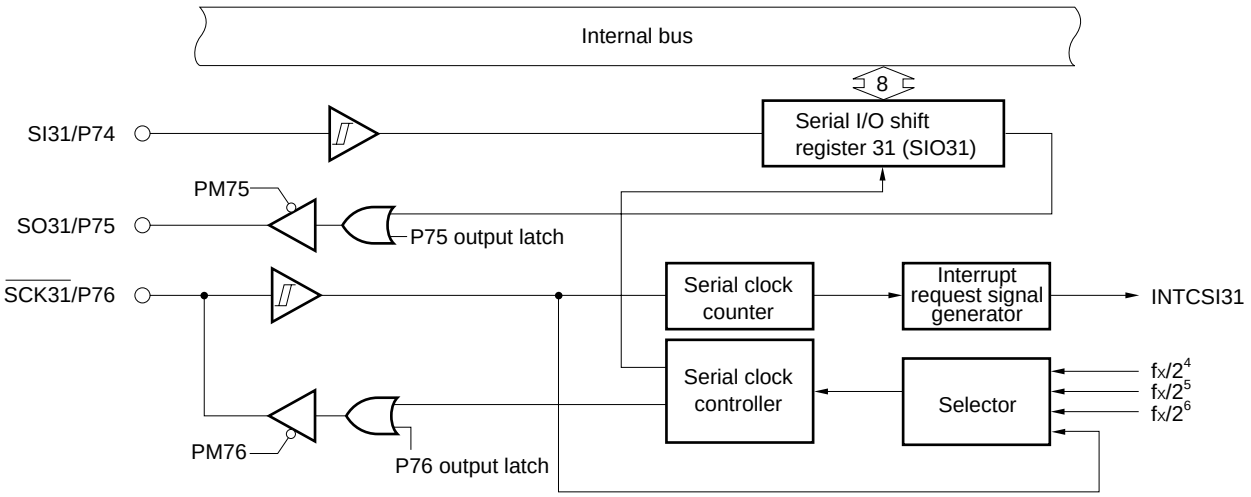
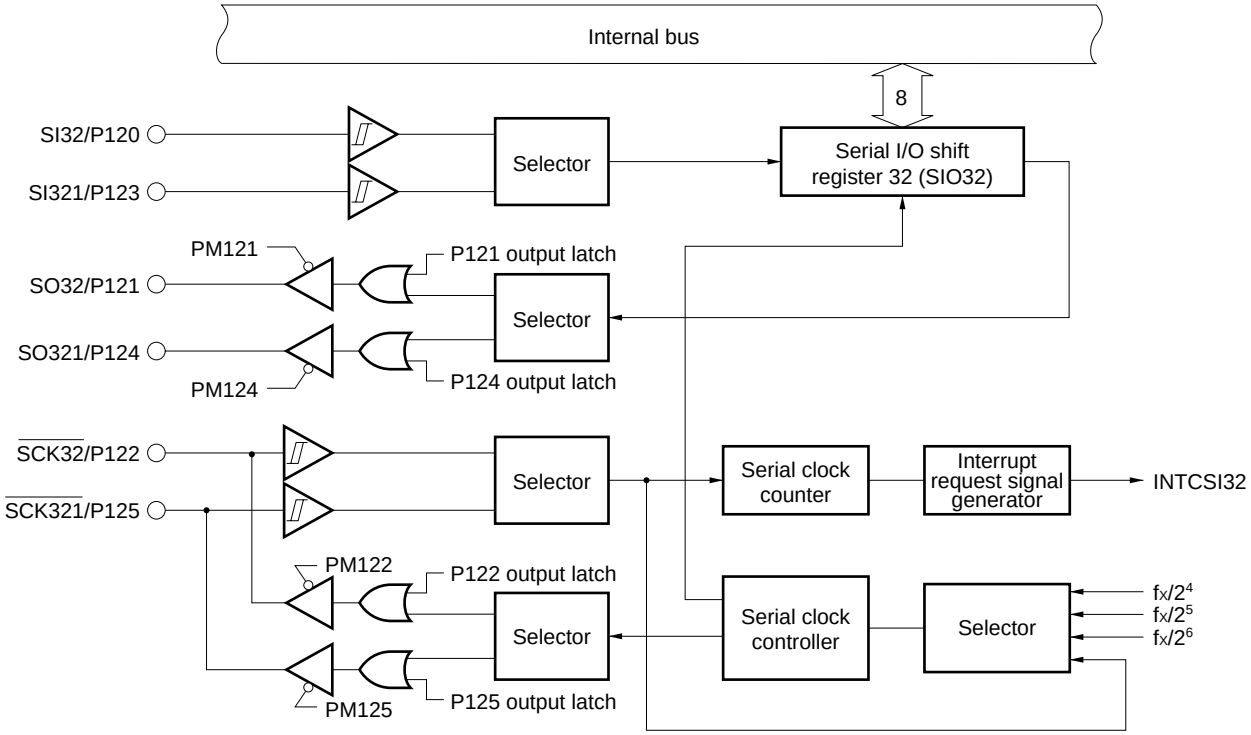


Figure 11-3. Block Diagram of Serial Interface SIO32



11.2 Configuration of Serial Interfaces SIO30 to SIO32

The serial interface SIO3n consists of the following hardware.

Table 11-1. Configuration of Serial Interfaces SIO30 to SIO32

Item	Configuration
Register	Serial I/O shift registers 30 to 32 (SIO30 to SIO32)
Control registers	Serial operating mode registers 30 to 32 (CSIM30 to CSIM32) Serial port select register 32 (SIO32SEL)

(1) Serial I/O shift registers 30 to 32 (SIO30 to SIO32)

These 8-bit registers convert parallel data into serial data and transmit or receive the serial data (shift operation) in synchronization with a serial clock.

SIO3n is set with an 8-bit memory manipulation instruction.

Serial operation is started by writing or reading data to or from SIO3n when bit 7 (CSIE3n) of serial operating mode register 3n (CSIM3n) is 1.

Data written to SIO3n is output to a serial output line (SO3n) for transmission.

Data is read to SIO3n from a serial input line (SI3n) for reception.

Reset input makes the values of these registers undefined.

Caution Do not execute access other than that for the transfer start trigger to SIO3n during a transfer operation (the read operation is disabled when MODE3n = 0, and the write operation is disabled when MODE3n = 1).

Remark n = 0 to 2

11.3 Registers Controlling Serial Interfaces SIO30 to SIO32

The following registers control the serial interface SIO3n.

- **Serial operating mode registers 30 to 32 (CSIM30 to CSIM32)**
- **Serial port select register 32 (SIO32SEL)**

(1) Serial operating mode register 30 to 32 (CSIM30 to CSIM32)

These registers select the serial clock of SIO3n and an operating mode, and enable or disable the operation.
CSIM3n is set with a 1-bit or 8-bit memory manipulation instruction.
Reset input clears these registers to 00H.

Figure 11-4. Format of Serial Operating Mode Registers 30 to 32 (CSIM30 to CSIM32)

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM30	CSIE30	0	0	0	0	MODE30	SCL301	SCL300	FF6FH	00H	R/W
CSIM31	CSIE31	0	0	0	0	MODE31	SCL311	SCL310	FF6DH	00H	R/W
CSIM32	CSIE32	0	0	0	0	MODE32	SCL321	SCL320	FF6BH	00H	R/W

CSIE3n	Enable/disable of SIO3n operation		
	Shift register operation	Serial counter	Port
0	Disables operation	Cleared	Port function**Note 1**
1	Enables operation	Enables counter operation	Serial function + port function**Note 2**

MODE3n	Transfer operation mode flag		
	Operating mode	Transfer start trigger	SO3n output
0	Transmit or transmit/receive mode	SIO3n write	Serial output
1	Receive only mode	SIO3n read	Fixed to low level**Note 3**

SCL3n1	SCL3n0	Clock selection
0	0	External clock input to $\overline{\text{SCK3n}}$
0	1	$f_X/2^4$ (281 kHz)
1	0	$f_X/2^5$ (141 kHz)
1	1	$f_X/2^6$ (70.3 kHz)

Notes 1. The SI3n, SO3n, and $\overline{\text{SCK3n}}$ pins can be used as port pins when CSIE3n = 0 (when SIO3n operation is stopped).

2. When CSIE3n = 1 (when SIO3n operation is enabled), the SI3n pin can be used as a port pin if only the transmission function is used, and the SO3n pin can be used as a port pin in the receive mode.

3. The SO3n pin can be used as a port pin.

Remarks 1. f_X: System clock oscillation frequency

2. ( ): f_X = 4.5 MHz

Caution Set the port mode register (PM××) as follows in the 3-wire serial I/O mode. Set the output latch to 0.

Serial Type / Operation Mode	Serial Interface SIO30	Serial Interface SIO31	Serial Interface SIO32	
			S32SEL0 = 0	S32SEL0 = 1
Serial clock output (master transmission or reception)	PM72 = 0 (set P72/$\overline{\text{SCK30}}$ pin to output mode)	PM76 = 0 (set P76/$\overline{\text{SCK31}}$ pin to output mode)	PM122 = 0 (set P122/$\overline{\text{SCK32}}$ pin to output mode)	PM125 = 0 (set P125/$\overline{\text{SCK321}}$ pin to output mode)
Serial clock input (slave transmission or reception)	PM72 = 1 (set P72/$\overline{\text{SCK30}}$ pin to input mode)	PM76 = 1 (set P76/$\overline{\text{SCK31}}$ pin to input mode)	PM122 = 1 (set P122/$\overline{\text{SCK32}}$ pin to input mode)	PM125 = 1 (set P125/$\overline{\text{SCK321}}$ pin to input mode)
In transmit or transmit/receive mode	PM71 = 0 (set P71/SO30 pin to output mode)	PM75 = 0 (set P75/SO31 pin to output mode)	PM121 = 0 (set P121/SO32 pin to output mode)	PM124 = 0 (set P124/SO321 pin to output mode)
In receive mode	PM70 = 1 (set P70/SI30 pin to input mode)	PM74 = 1 (set P74/SI31 pin to input mode)	PM120 = 1 (set P120/SI32 pin to input mode)	PM123 = 1 (set P123/SI321 pin to input mode)

(2) Serial port select register 32 (SIO32SEL)

This register selects the port used for serial interface SIO32.

SIO32SEL is set with a 1-bit or 8-bit memory manipulation instruction.

Reset input clears this register to 00H.

Figure 11-5. Format of Serial Port Select Register 32 (SIO32SEL)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
SIO32SEL	0	0	0	0	0	0	0	S32SEL0	FF69H	00H	R/W

S32SEL0	Serial interface SIO32 port selection		
	SI pin	SO pin	$\overline{\text{SCK}}$ pin
0Note 1	P120/SI32	P121/SO32	P122/$\overline{\text{SCK32}}$
1Note 2	P123/SI321	P124/SO321	P125/$\overline{\text{SCK321}}$

Notes 1. The P123/SI321, P124/SO321, P125/$\overline{\text{SCK321}}$ pins can be used as port pins.

2. The P120/SI32, P121/SO32, P122/$\overline{\text{SCK32}}$ pins can be used as port pins.

11.4 Operations of Serial Interfaces SIO30 to SIO32

This section explains the two modes of the serial interfaces SIO30 to SIO32.

11.4.1 Operation stop mode

In this mode, serial transfer is not performed.

The alternate-function pins used for the serial interface can be used as ordinary I/O port pins.

(1) Register setting

The operation stop mode is set using serial operating mode register 3n (CSIM3n).

CSIM3n is set with a 1-bit or 8-bit memory manipulation instruction.

Reset input clears this register to 00H.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM30	CSIE30	0	0	0	0	MODE30	SCL301	SCL300	FF6FH	00H	R/W
CSIM31	CSIE31	0	0	0	0	MODE31	SCL311	SCL310	FF6DH	00H	R/W
CSIM32	CSIE32	0	0	0	0	MODE32	SCL321	SCL320	FF6BH	00H	R/W

CSIE3n	Enable/disable of SIO3n operation		
	Shift register operation	Serial counter	Port
0	Disables operation	Cleared	Port function[Note 1]
1	Enables operation	Enables count operation	Serial function + port function[Note 2]

Notes 1. The SI3n, SO3n, and $\overline{\text{SCK3n}}$ pins can be used as port pins when CSIE3n = 0 (when SIO3n operation is stopped).

2. When CSIE3n = 1 (when SIO3n operation is enabled), the SI3n pin can be used as a port pin if only the transmission function is used, and the SO3n pin can be used as a port pin in the receive mode.

Remark n = 0 to 2

11.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connecting a peripheral I/O or display controller equipped with a clocked serial interface.

In this mode, communication is executed by using three lines: serial clock ($\overline{SCK3n}$), serial output (SO3n), and serial input (SI3n) lines.

(1) Register setting

The 3-wire serial I/O mode is set using serial operating mode register 3n (CSIM3n).

These registers are set with a 1-bit or 8-bit memory manipulation instruction.

Reset input clears these registers to 00H.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM30	CSIE30	0	0	0	0	MODE30	SCL301	SCL300	FF6FH	00H	R/W
CSIM31	CSIE31	0	0	0	0	MODE31	SCL311	SCL310	FF6DH	00H	R/W
CSIM32	CSIE32	0	0	0	0	MODE32	SCL321	SCL320	FF6BH	00H	R/W

CSIE3n	Enable/disable of SIO3n operation		
	Shift register operation	Serial counter	Port
0	Disables operation	Cleared	Port function**Note 1**
1	Enables operation	Enables counter operation	Serial function + port function**Note 2**

MODE3n	Transfer operation mode flag		
	Operating mode	Transfer start trigger	SO3n output
0	Transmit or transmit/receive mode	SIO3n write	Serial output
1	Receive-only mode	SIO3n read	Fixed to low level**Note 3**

SCL3n1	SCL3n0	Clock selection
0	0	External clock input to $\overline{SCK3n}$
0	1	$f_X/2^4$ (281 kHz)
1	0	$f_X/2^5$ (141 kHz)
1	1	$f_X/2^6$ (70.3 kHz)

Notes 1. The SI3n, SO3n, and $\overline{SCK3n}$ pins can be used as port pins when CSIE3n = 0 (when SIO3n operation is stopped).

2. When CSIE3n = 1 (when SIO3n operation is enabled), the SI3n pin can be used as a port pin if only the transmission function is used, and the SO3n pin can be used as a port pin in the receive mode.

3. The SO3n pin can be used as a port pin.

Remarks 1. f_X: System clock oscillation frequency

2. ( ): f_X = 4.5 MHz

Caution Set the port mode register (PM××) as follows in the 3-wire serial I/O mode. Set the output latch to 0.

Operation Mode \ Serial Type	Serial Interface SIO30	Serial Interface SIO31	Serial Interface SIO32	
			S32SEL0 = 0	S32SEL0 = 1
Serial clock output (master transmission or reception)	PM72 = 0 (set P72/$\overline{\text{SCK30}}$ pin to output mode)	PM76 = 0 (set P76/$\overline{\text{SCK31}}$ pin to output mode)	PM122 = 0 (set P122/$\overline{\text{SCK32}}$ pin to output mode)	PM125 = 0 (set P125/$\overline{\text{SCK321}}$ pin to output mode)
Serial clock input (slave transmission or reception)	PM72 = 1 (set P72/$\overline{\text{SCK30}}$ pin to input mode)	PM76 = 1 (set P76/$\overline{\text{SCK31}}$ pin to input mode)	PM122 = 1 (set P122/$\overline{\text{SCK32}}$ pin to input mode)	PM125 = 1 (set P125/$\overline{\text{SCK321}}$ pin to input mode)
In transmit or transmit/receive mode	PM71 = 0 (set P71/SO30 pin to output mode)	PM75 = 0 (set P75/SO31 pin to output mode)	PM121 = 0 (set P121/SO32 pin to output mode)	PM124 = 0 (set P124/SO321 pin to output mode)
In receive mode	PM70 = 1 (set P70/SI30 pin to input mode)	PM74 = 1 (set P74/SI31 pin to input mode)	PM120 = 1 (set P120/SI32 pin to input mode)	PM123 = 1 (set P123/SI321 pin to input mode)

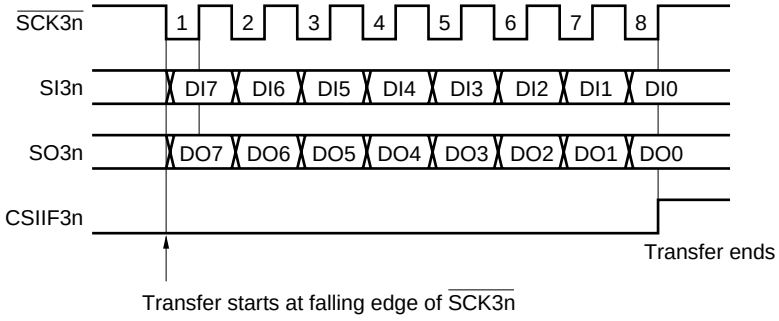
(2) Communication operation

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

The shift operation of serial I/O shift register 3n (SIO3n) is performed at the falling edge of the serial clock ($\overline{\text{SCK3n}}$). The transmit data is retained in SO3n latch and is output from the SO3n pin. The receive data input to the SI3n pin is latched to SIO3n at the falling edge of the serial clock.

When 8-bit data has been transferred, the operation of SIO3n is automatically stopped, and an interrupt request flag (CSIIF3n) is set.

Figure 11-6. Timing in 3-Wire Serial I/O Mode



(3) Starting transfer

Serial transfer is started by writing (or reading) the transfer data to serial I/O shift register 3n (SIO3n) when the following conditions are satisfied.

- Operation control bit of SIO3n (bit 7 (CSIE3n) of serial operation mode register 3n (CSIM3n)) = 1
- If the internal serial clock is stopped or $\overline{\text{SCK3n}}$ is high level after transfer of 8-bit serial data
- Transmit/receive mode
  Transfer is started if SIO3n is written when bit 7 (CSIE3n) of CSIM3n = 1, and bit 2 (MODE3n) = 0
- Receive mode
  Transfer is started if SIO3n is read when bit 7 (CSIE3n) of CSIM3n = 1, and bit 2 (MODE3n) = 1

Caution Serial transfer is not started even if 1 is written to CSIE3n after data is written to SIO3n.

On completion of transfer of the 8-bit data, serial transfer is automatically stopped, and an interrupt request flag (CSIIF3n) is set.

CHAPTER 12 INTERRUPT FUNCTIONS

12.1 Interrupt Function Types

The following three types of interrupt functions are used.

(1) Non-maskable interrupts

This type of interrupt is acknowledged unconditionally even if interrupts are disabled. It does not undergo interrupt priority control and is given top priority over all other interrupt requests.
It generates a standby release signal.
One interrupt source from the watchdog timer is incorporated as a non-maskable interrupt.

(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 specification flag register (PR).
Multiple interrupt servicing is possible if a high-priority interrupt is generated while a low-priority interrupt is being serviced. If two or more interrupts with the same priority are simultaneously generated, each interrupt has a predetermined priority (refer to **Table 12-1**).
A standby release signal is generated.
Maskable interrupts are provided for each product as follows.

- μPD178053, 178054, 178F054 Internal: 11, external: 5

(3) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even in an interrupt-disabled state. The software interrupt does not undergo interrupt priority control.

12.2 Interrupt Sources and Configuration

The μPD178053, 178054, and 178F054 have a total of 17 sources (non-maskable interrupt, maskable interrupt, software interrupt) (refer to **Table 12-1**).

Remark Either a non-maskable interrupt or a maskable interrupt (internal) can be selected for the watchdog timer interrupt source (INTWDT).

Table 12-1. Interrupt Sources

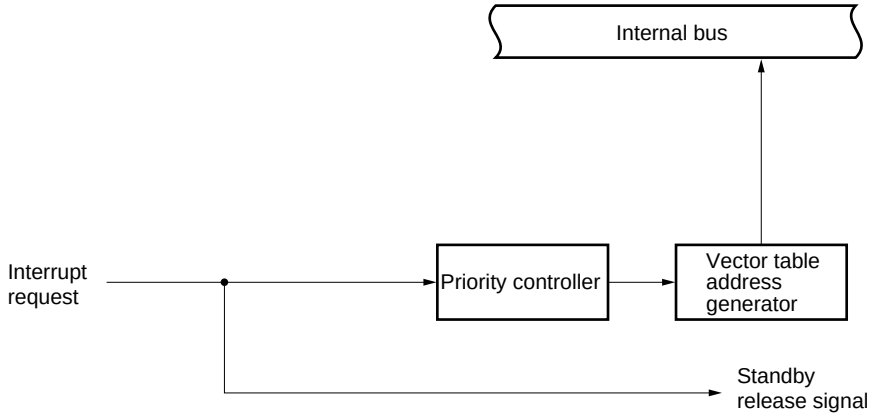
Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Non-maskable	–	INTWDT	Overflow of watchdog timer (when watchdog timer mode 1 is selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Overflow of watchdog timer (when interval timer mode is selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTP4			000EH	
	6	INTKY	Detection of key input of port 4	Internal	0010H	(B)
	7	INTCSI31	End of transfer by serial interface SIO31		0012H	
	8	INTBTM0	Generation of basic timer match signal		0014H	
	9	INTAD3	End of conversion by A/D converter		0016H	
	10	INTCSI32	End of transfer by serial interface SIO32		0018H	
	11	INTCSI30	End of transfer by serial interface SIO30		001AH	
	12	INTTM50	Generation of match signal of 8-bit timer/ event counter 50		001CH	
	13	INTTM51	Generation of match signal of 8-bit timer/ event counter 51		001EH	
	14	INTTM52	Generation of match signal of 8-bit timer/ event counter 52		0020H	
	15	INTTM53	Generation of match signal of 8-bit timer 53		0022H	
Software	–	BRK	Execution of BRK instruction	–	003EH	(D)

Notes 1. If two or more maskable interrupts occur at the same time, they are acknowledged or held pending according to their default priorities. The default priority 0 is the highest, and 15 is the lowest.

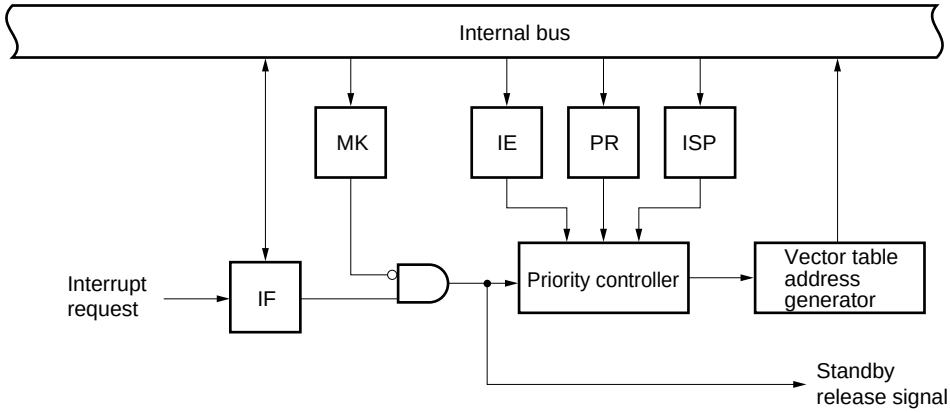
2. (A) to (D) under the heading Basic Configuration Type correspond to (A) to (D) in Figure 12-1.

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

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt

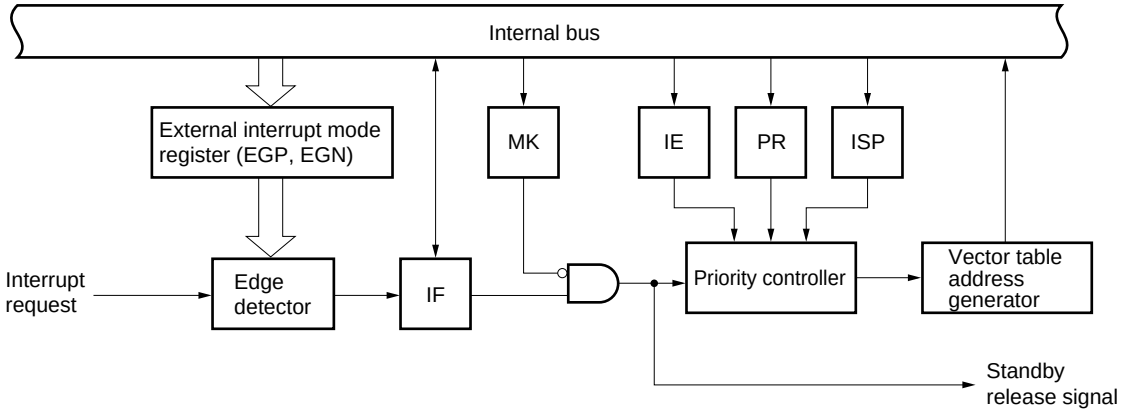
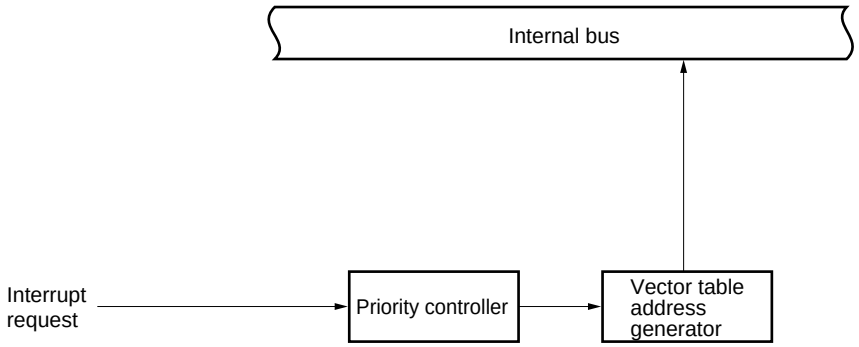


Figure 12-1. Basic Configuration of Interrupt Function (2/2)

(D) Software interrupt



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

12.3 Registers Controlling Interrupt Functions

The following six types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H)
- Interrupt mask flag register (MK0L, MK0H)
- Priority specification flag register (PR0L, PR0H)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)
- Program status word (PSW)

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

Table 12-2. Various Flags Corresponding to Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTWDT	WDTIF	IF0L	WDTMK	MK0L	WDTPR	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTP4	PIF4		PMK4		PPR4	
INTKY	KYIF		KYMK		KYPR	
INTCSI31	CSIIF31		CSIMK31		CSIPR31	
INTBTM0	BTMIF0	IF0H	BTMMK0	MK0H	BTMPR0	PR0H
INTAD3	ADIF		ADMK		ADPR	
INTCSI32	CSIIF32		CSIMK32		CSIPR32	
INTCSI30	CSIIF30		CSIMK30		CSIPR30	
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM51	TMIF51		TMMK51		TMPR51	
INTTM52	TMIF52		TMMK52		TMPR52	
INTTM53	TMIF53		TMMK53		TMPR53	

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

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 reset input.

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

Reset input clears these registers to 00H.

Figure 12-2. Format of Interrupt Request Flag Registers (IF0L, IF0H)

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
IF0L	CSIIF31	KYIF	PIF4	PIF3	PIF2	PIF1	PIF0	WDTIF	FFE0H	00H	R/W
	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>			
IF0H	TMIF53	TMIF52	TMIF51	TMIF50	CSIIF30	CSIIF32	ADIF	BTMIF0	FFE1H	00H	R/W

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

Cautions 1. WDTIF flag is R/W enabled only when a watchdog timer is used as an interval timer. If a watchdog timer is used in watchdog timer mode 1, set WDTIF flag to 0.

2. To operate the timers, serial interface, and A/D converter after the standby mode has been released, clear the interrupt request flag, because the interrupt request flag may be set by noise.

3. When an interrupt is acknowledged, the interrupt request flag is automatically cleared before entering the interrupt routine.

(2) Interrupt mask flag registers (MK0L, MK0H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing and to set standby clear enable/disable.

MK0L and MK0H are set with a 1-bit or 8-bit memory manipulation instruction. If MK0L and MK0H are used as a 16-bit register MK0, use a 16-bit memory manipulation instruction for the setting.

Reset input sets these registers to FFH.

Figure 12-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H)

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
MK0L	CSIMK31	KYMK	PMK4	PMK3	PMK2	PMK1	PMK0	WDTMK	FFE4H	FFH	R/W
	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>			
MK0H	TMMK53	TMMK52	TMMK51	TMMK50	CSIMK30	CSIMK32	ADMK	BTMMK0	FFE5H	FFH	R/W

××MK×	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Cautions 1. If the WDTMK flag is read when the watchdog timer is used in watchdog timer mode 1, MK0 value becomes undefined.

2. Because port 0 functions alternately as the external interrupt request input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, the interrupt mask flag should be set to 1 before using the output mode.

(3) Priority specification flag registers (PR0L, PR0H)

The priority specification flags are used to set the corresponding maskable interrupt priority orders.

PR0L and PR0H are set with a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H are used as a 16-bit register PR0, use a 16-bit memory manipulation instruction for the setting.

Reset input sets these registers to FFH.

Figure 12-4. Format of Priority Specification Flag Registers (PR0L, PR0H)

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
PR0L	CSIPR31	KYPR	PPR4	PPR3	PPR2	PPR1	PPR0	WDTPR	FFE8H	FFH	R/W
	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>			
PR0H	TMPR53	TMPR52	TMPR51	TMPR50	CSIPR30	CSIPR32	ADPR	BTMPR0	FFE9H	FFH	R/W

××PR×	Priority level selection
0	High priority level
1	Low priority level

Caution When the watchdog timer is used in watchdog timer mode 1, set the WDTPR flag to 1.

(4) External interrupt rising edge enable register (EGP), external interrupt falling edge enable register (EGN)

These registers set the valid edge for INTP0 to INTP4.

EGP and EGN are set with a 1-bit or 8-bit memory manipulation instructions.

Reset input clears these registers to 00H.

Figure 12-5. Format of External Interrupt Rising Edge Enable Register (EGP) and External Interrupt Falling Edge Enable Register (EGN)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
EGP	0	0	0	EGP4	EGP3	EGP2	EGP1	EGP0	FF48H	00H	R/W
	7	6	5	4	3	2	1	0			
EGN	0	0	0	EGN4	EGN3	EGN2	EGN1	EGN0	FF49H	00H	R/W

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 4)
0	0	Interrupt prohibited
0	1	Falling edge
1	0	Rising edge
1	1	Both falling and rising edges

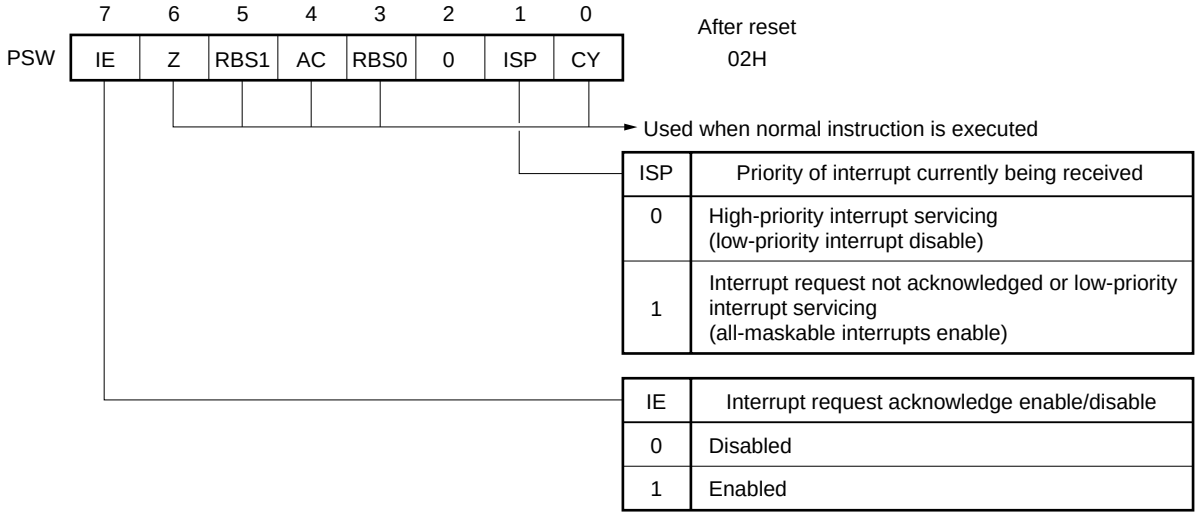
(5) Program status word (PSW)

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

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

Reset input sets PSW to 02H.

Figure 12-6. Configuration of Program Status Word (PSW)



ISP	Priority of interrupt currently being received
0	High-priority interrupt servicing (low-priority interrupt disable)
1	Interrupt request not acknowledged or low-priority interrupt servicing (all-maskable interrupts enable)

IE	Interrupt request acknowledge enable/disable
0	Disabled
1	Enabled

12.4 Interrupt Servicing Operations

12.4.1 Non-maskable interrupt request acknowledgement operation

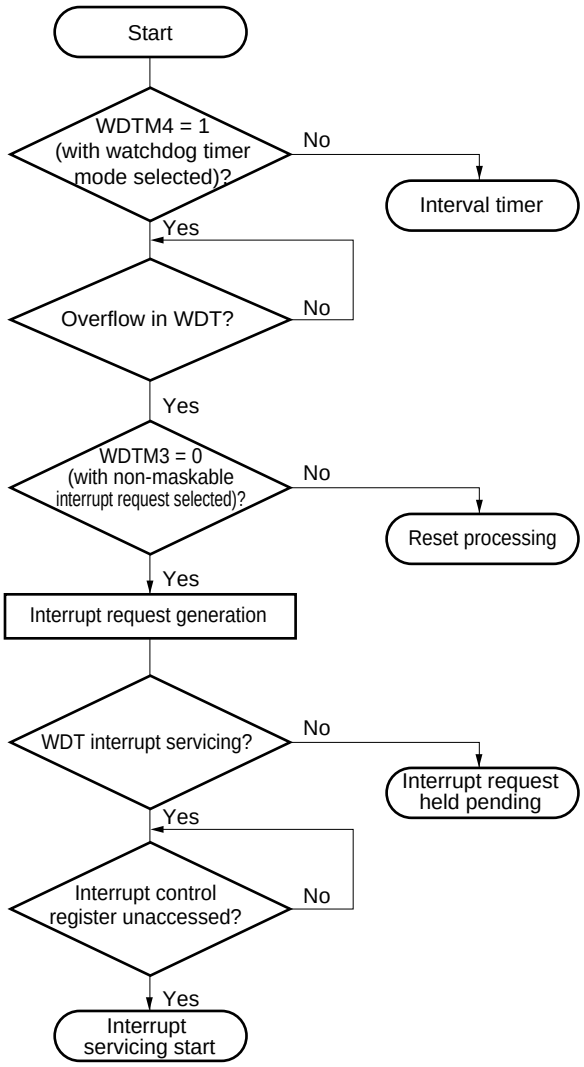
A non-maskable interrupt request is unconditionally acknowledged even if in an interrupt request acknowledgement disabled state. It does not undergo interrupt priority control and has the highest priority over all other interrupts.

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

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

Figure 12-7 shows the flowchart from generation of the non-maskable interrupt request to acknowledging it. Figure 12-8 shows the timing of acknowledging the non-maskable interrupt request, and Figure 12-9 shows the operation performed if a more than one non-maskable interrupt request occurs.

Figure 12-7. Flowchart from Generation of Non-Maskable Interrupt Request to Acknowledgement



WDTM: Watchdog timer mode register
WDT: Watchdog timer

Figure 12-8. Non-Maskable Interrupt Request Acknowledgement Timing

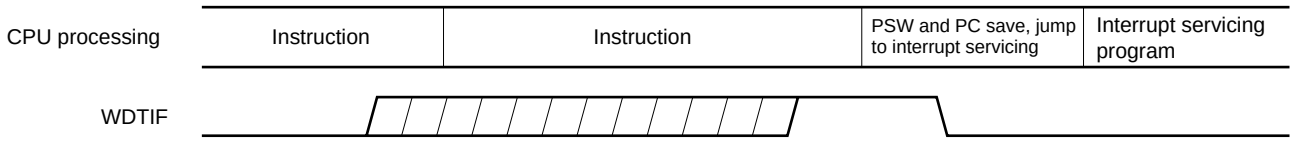
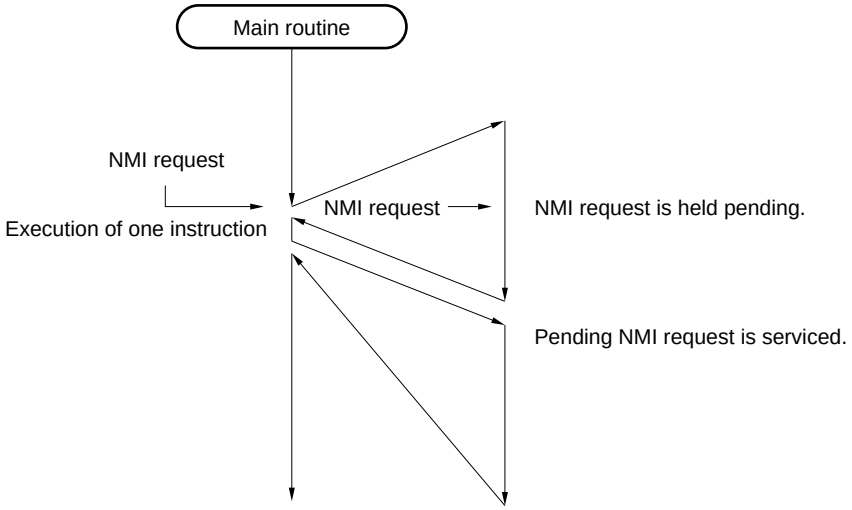
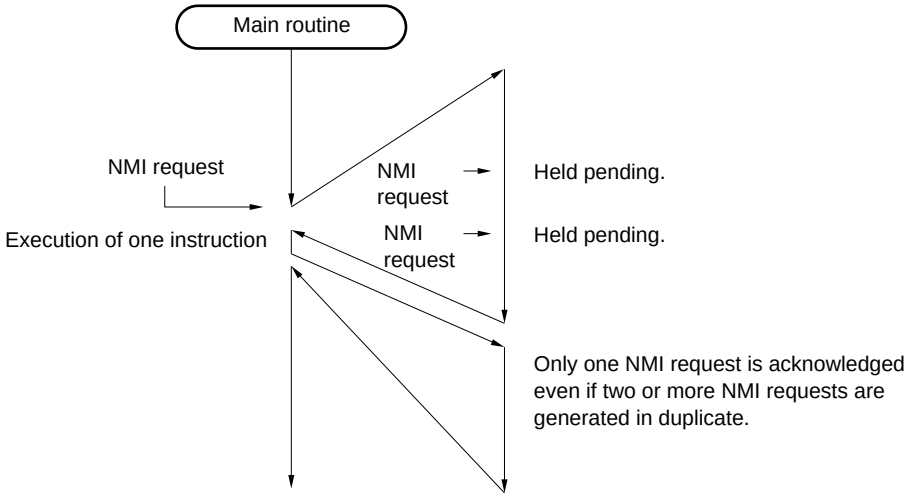


Figure 12-9. Non-Maskable Interrupt Request Acknowledgement Operation

(a) If a new non-maskable interrupt request is generated during non-maskable interrupt servicing program execution



(b) If two non-maskable interrupt requests are generated during non-maskable interrupt servicing program execution



12.4.2 Maskable interrupt request acknowledgement operation

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

Wait times from maskable interrupt request generation to interrupt request servicing are as follows.

For the interrupt acknowledge timing, refer to **Figures 12-11** and **12-12**.

Table 12-3. Times from Maskable Interrupt Request Generation to Interrupt 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: $\frac{1}{f_{CPU}}$ (f_{CPU}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request specified for higher priority with the priority specification flag is acknowledged first. If two or more requests are specified as the same priority by the priority specification flag, the default priorities apply.

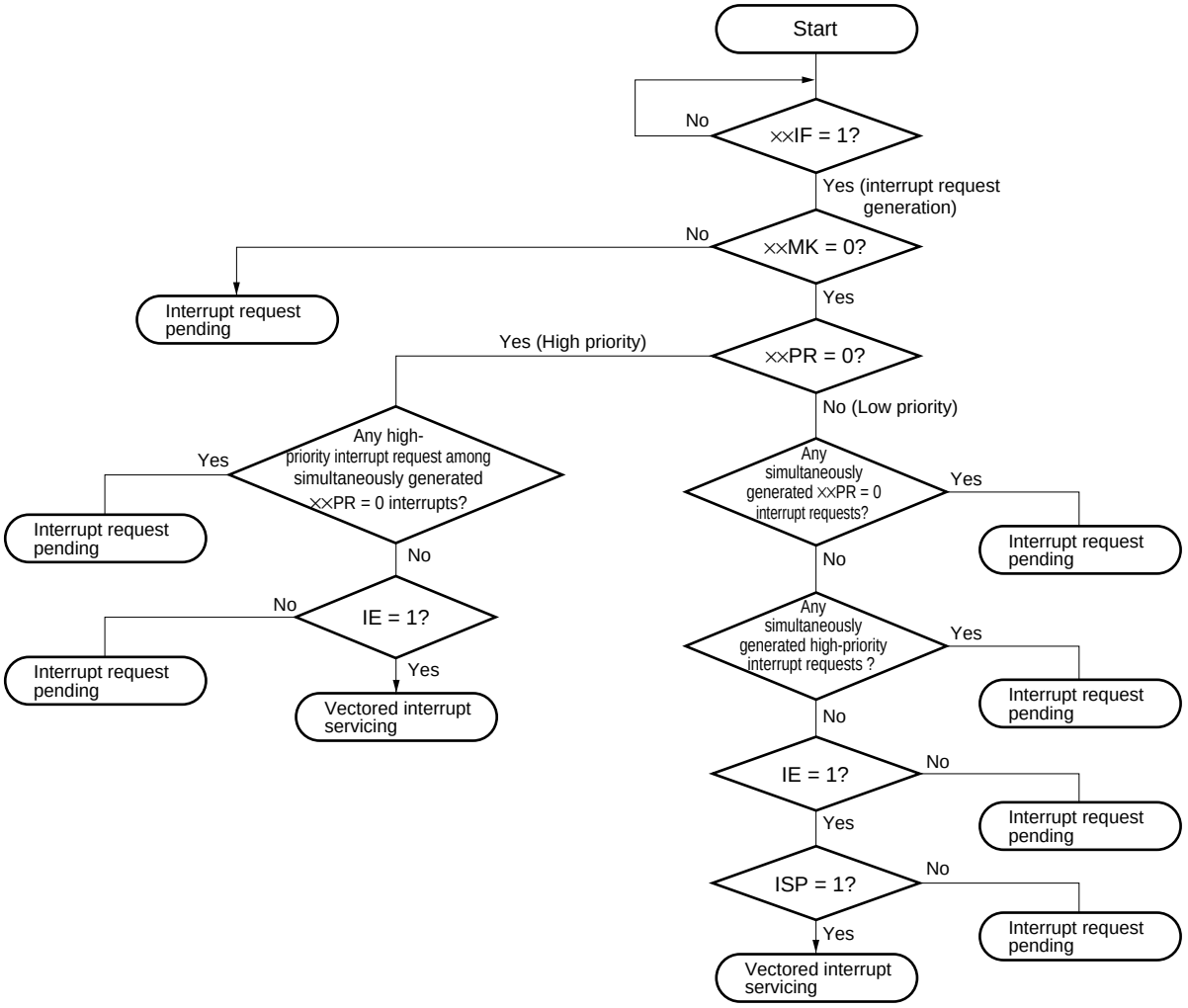
Any pending interrupt requests are acknowledged when they become acknowledgeable.

Figure 12-10 shows interrupt request acknowledgement algorithms.

If a maskable interrupt request is acknowledged, the acknowledged interrupt request is saved to the stack, the program status word (PSW) and the program counter (PC), in that order, the IE flag is reset to 0, and the acknowledged interrupt priority specification flag contents are transferred to the ISP flag. Further, the vector table data determined for each interrupt request is loaded into the PC and branched.

Return from the interrupt is possible with the RETI instruction.

Figure 12-10. Interrupt Request Acknowledgement Processing Algorithm



××IF: Interrupt request flag

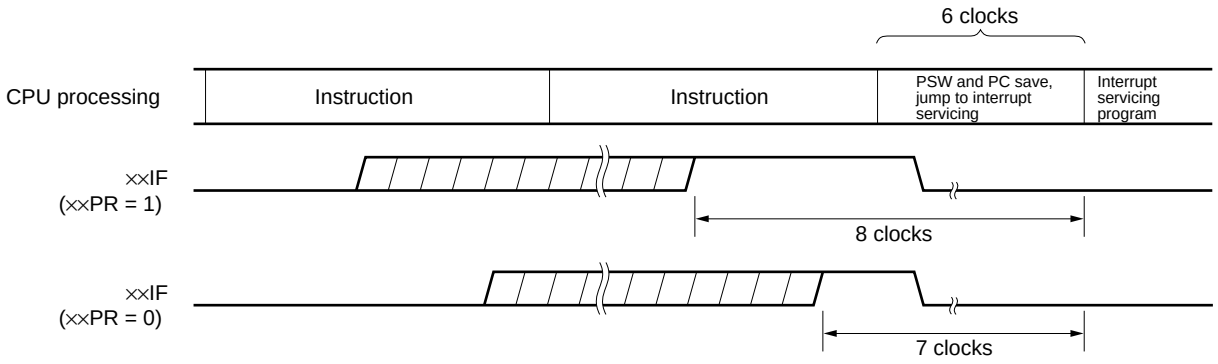
××MK: Interrupt mask flag

××PR: Priority specification flag

IE: Flag controlling acknowledging maskable interrupt request (1 = enable, 0 = disable)

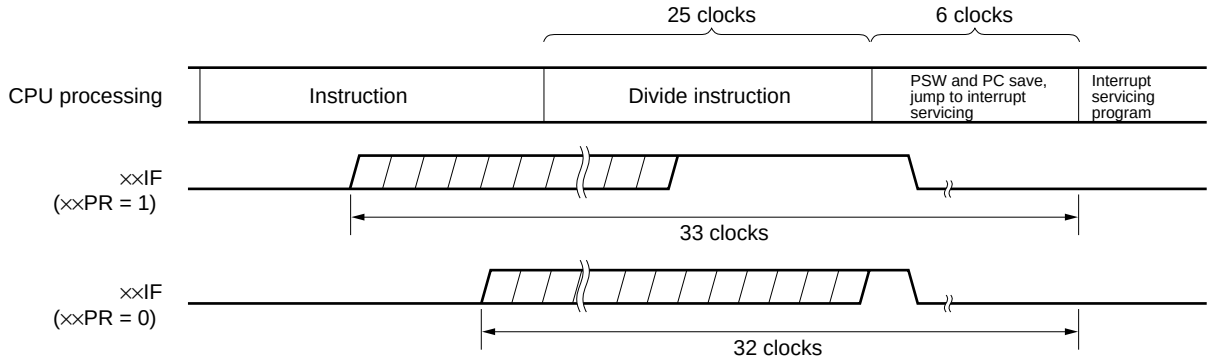
ISP: Flag indicating priority of interrupt currently being serviced (0 = interrupt with high priority serviced, 1 = interrupt request is not acknowledged, or interrupt with low priority serviced)

Figure 12-11. Interrupt Request Acknowledgement Timing (Minimum Time)



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

Figure 12-12. Interrupt Request Acknowledgement Timing (Maximum Time)



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

12.4.3 Software interrupt request acknowledgement operation

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

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

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

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

12.4.4 Multiple interrupt servicing

The acknowledgement of another interrupt request while an interrupt is being serviced is called multiple interrupt servicing.

Multiple interrupt servicing does not take place unless the interrupts (except the non-maskable interrupt) are abled to be acknowledged (IE = 1). Acknowledging another interrupt request is disabled (IE = 0) when one interrupt has been acknowledged. Therefore, to enable multiple interrupt servicing, the EI flag must be set to 1 during interrupt servicing, to enable other interrupts.

Multiple interrupt servicing may not occur even when interrupts are enabled. This is controlled by the priorities of the interrupts. Although two types of priorities, default priority and programmable priority, may be assigned to an interrupt, multiple interrupt servicing is controlled by using the programmable priority.

If an interrupt with the same priority as or a higher priority than the interrupt currently being serviced occurs, that interrupt can be acknowledged and serviced. If an interrupt with a priority lower than that of the interrupt currently being serviced occurs, that interrupt cannot be acknowledged and serviced.

An interrupt that is not acknowledged and serviced because it is disabled or it has a low priority is held pending. This interrupt is acknowledged after servicing of the current interrupt has been completed and one instruction of the main routine has been executed.

Multiple interrupt servicing is not enabled while a non-maskable interrupt is being serviced.

Table 12-4 shows the interrupts that can enter multiple interrupt servicing, and Figure 12-13 shows an example of multiple interrupt servicing.

Table 12-4. Interrupt Request Enabled for Multiple Interrupt Servicing 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		D	D	D	D	D
Maskable interrupt	ISP = 0	E	E	D	D	D
	ISP = 1	E	E	D	E	D
Software interrupt servicing		E	E	D	E	D

Remarks 1. E: Multiple interrupt servicing enabled

2. D: Multiple interrupt servicing disabled

3. ISP and IE are the flags contained in PSW

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

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

IE = 0: Interrupt request acknowledgement is disabled

IE = 1: Interrupt request acknowledgement is enabled

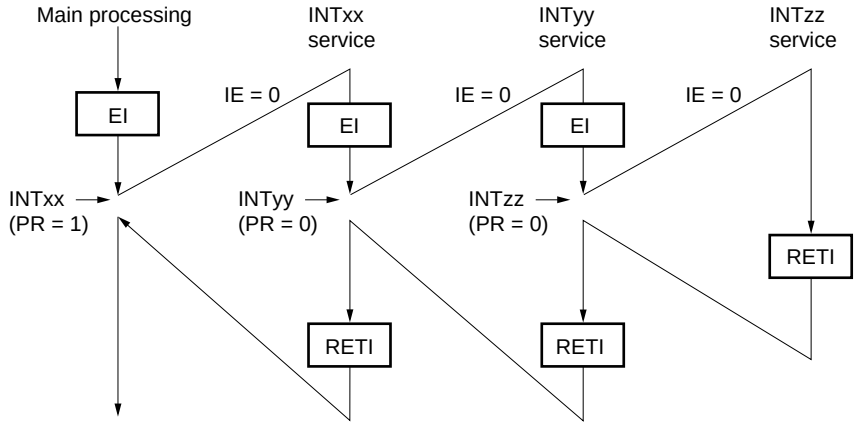
4. PR is a flag contained in PR0L and PR0R.

PR = 0: Higher priority level

PR = 1: Lower priority level

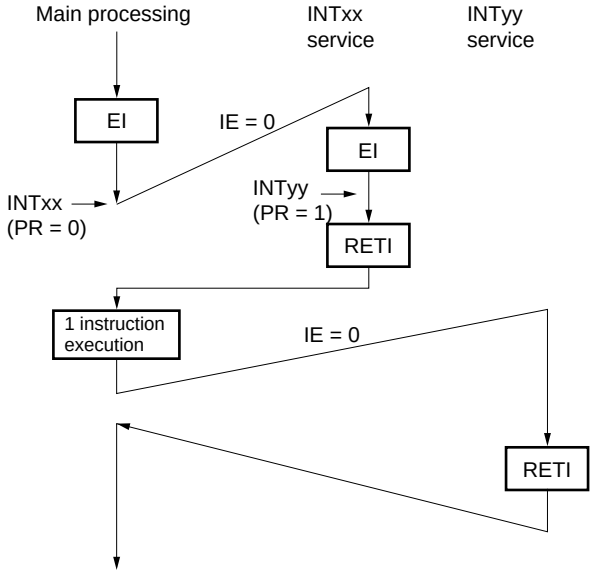
Figure 12-13. Multiple Interrupt Servicing Example (1/2)

Example 1. Example where multiple interrupt occurs two times



Two interrupt requests, INTyy and INTzz, are acknowledged while interrupt INTxx is serviced, and multiple interrupt occurs. Before each interrupt request is acknowledged, the EI instruction is always executed, and the interrupt is enabled.

Example 2. Example where multiple interrupt does not occur because of priority control



Interrupt request INTyy that is generated while interrupt INTxx is being serviced is not acknowledged because its priority is lower than that of INTxx, and therefore, multiple interrupt does not occur. INTyy request is held pending, and is acknowledged after one instruction of the main routine has been executed.

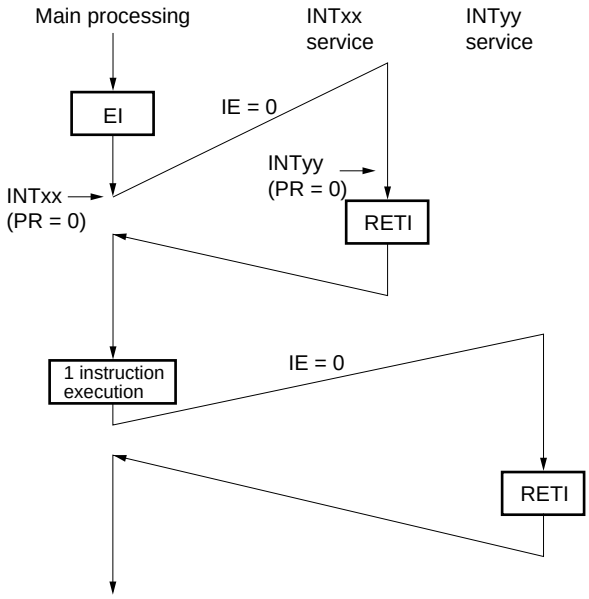
PR = 0: High-priority level

PR = 1: Low-priority level

IE = 0: Acknowledging interrupt request is disabled.

Figure 12-13. Multiple Interrupt Servicing Example (2/2)

Example 3. Example where multiple interrupt does not occur because interrupts are not enabled



Because interrupts are not enabled (EI instruction is not issued) in interrupt servicing INTxx, interrupt request INTyy is not acknowledged, and multiple interrupt does not occur. The INTyy request is held pending, and is acknowledged after one instruction of the main routine has been executed.

PR = 0: High priority level
IE = 0: Acknowledging interrupts is disabled.

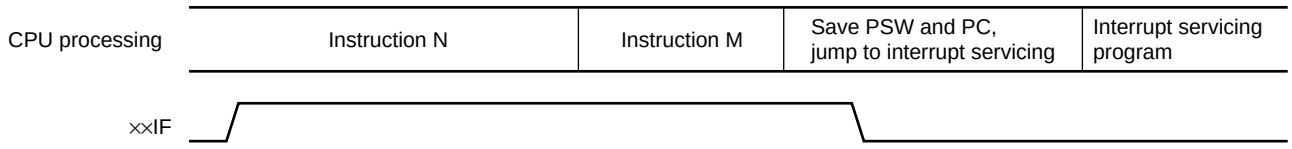
12.4.5 Pending interrupt requests

Even if an interrupt request is generated, the following instructions hold it pending.

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW.bit, CY
- MOV1/AND1/OR1/XOR1 CY, PSW.bit
- SET1/CLR1 PSW.bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT/BF/BTCLR PSW.bit, $addr16
- EI
- DI
- Instructions manipulating IF0L, IF0H, MK0L, MK0H, PR0L, and PR0H registers

Caution Because the IE flag is cleared to 0 by the software interrupt (caused by execution of the BRK instruction), a maskable interrupt request is not acknowledged even if it occurs while the BRK instruction is executed. However, a non-maskable interrupt is acknowledged.

Figure 12-14. Pending Interrupt Request



Remarks 1. Instruction N: Instruction that holds interrupt request pending

2. Instruction M: Instruction that does not hold interrupt request pending

3. Operation of ××IF is not affected by value of ××PR.

CHAPTER 13 PLL FREQUENCY SYNTHESIZER

13.1 Function of PLL Frequency Synthesizer

The PLL (Phase Locked Loop) frequency synthesizer is used to lock the frequency in the MF (Middle Frequency), HF (High Frequency), and VHF (Very High Frequency) ranges to a specific frequency by means of phase difference comparison.

The PLL frequency synthesizer divides the frequency of the signal input from the VCOL or VCOH pin by using a programmable divider, and outputs the phase difference between the frequency of this signal and reference frequency from the EO0 and EO1 pin.

The following input pin states and frequency division modes are used.

(1) Direct division (MF) mode

The VCOL pin is used.

The VCOH pin is set in the status specified by bit 3 (VCOHDMD) of the PLL mode select register (PLLMD).

(2) Pulse swallow (HF) mode

The VCOL pin is used.

The VCOH pin is set in the status specified by bit 3 (VCOHDMD) of PLLMD.

(3) Pulse swallow (VHF) mode

The VCOH pin is used.

The VCOL pin is set in the status specified by bit 2 (VCOLDMD) of PLLMD.

(4) VCOL and VCOH pin disable

The VCOL and VCOH pins are set in the status specified by bits 2 (VCOLDMD) and 3 (VCOHDMD) of PLLMD.

At this time, the phase comparator, reference frequency generator, and charge pump operate.

(5) PLL disable

The PLL disabled status is set by the PLL reference mode register (PLLRF).

The VCOH and VCOL pins are set in the status specified by bits 2 (VCOLDMD) and 3 (VCOHDMD) of PLLMD.

The EO0 and EO1 pins go into a high-impedance state.

At this time, all the internal PLL operations are stopped.

These division modes are selected by using the PLL mode select register (PLLMD).

The division value (N value) is set to the programmable divider by using the PLL data register. Frequency division in each of the above modes is carried out according to the value (N value) set to the programmable divider.

Table 13-1 shows the division modes, input pins used (VCOL pin or VCOH pin), and the value that can be set to the programmable divider.

Table 13-1. Division Mode, Input Pin, and Division Value

Division Mode	Pin Used	Value That Can Be Set
Direct division (MF)	VCOL	32 to 2^{12}–1
Pulse swallow (HF)	VCOL	1024 to 2^{17}–1
Pulse swallow (VHF)	VCOH	1024 to 2^{17}–1

Caution For the frequencies that can be actually input, and input amplitude, refer to CHAPTER 19 ELECTRICAL SPECIFICATIONS.

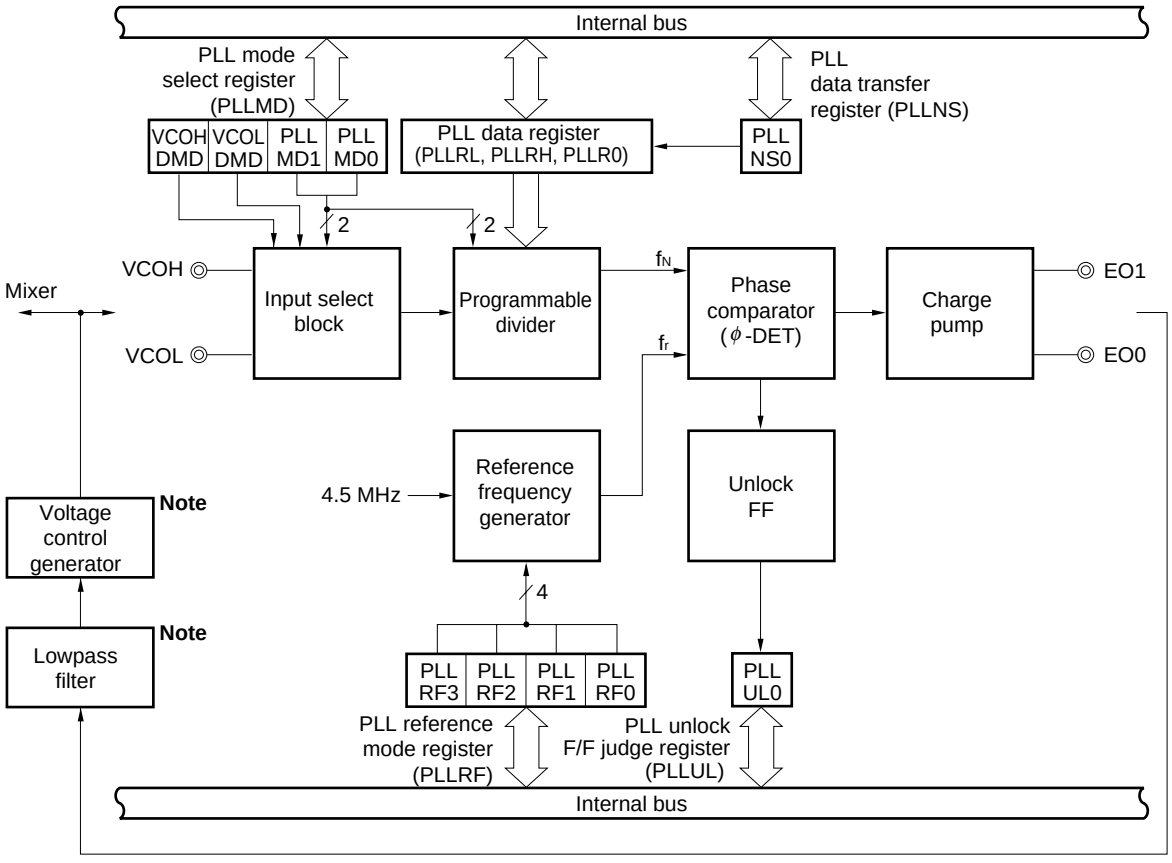
13.2 Configuration of PLL Frequency Synthesizer

The PLL frequency synthesizer consists of the following hardware.

Table 13-2. Configuration of PLL Frequency Synthesizer

Item	Configuration
Data registers	PLL data register L (PLLRL) PLL data register H (PLLRH) PLL data register 0 (PLLR0)
Control registers	PLL mode select register (PLLMD) PLL reference mode register (PLLRF) PLL unlock F/F judge register (PLLUL) PLL data transfer register (PLLNS)

Figure 13-1. Block Diagram of PLL Frequency Synthesizer



Note External circuit

(1) PLL data register L (PLLRL), PLL data register H (PLLRH), and PLL data register 0 (PLLR0)

These registers set the division value of the PLL frequency synthesizer. The division value of the PLL frequency synthesizer is made up of 17 bits. The higher 16 bits of this value are set by PLL data register L (PLLRL) and PLL data register H (PLLRH). The higher 16 bits can also be set by the PLL data register (PLLR). The least significant bit is set by bit 7 (PLLSCN) of PLL data register 0 (PLLR0).

Reset input makes the contents of these registers undefined. These registers hold the current values in the STOP and HALT modes.

(2) Input select block

The input select block consists of the VCOL and VCOH pins, and input amplifiers of the respective pins.

(3) Programmable divider

The programmable divider consists of two modulus prescalers, a programmable counter (12 bits), a swallow counter (5 bits), and a division mode select switch.

(4) Reference frequency generator

The reference frequency generator consists of a divider that generates the reference frequency f_r of the PLL frequency synthesizer, and a multiplexer.

(5) Phase comparator

The phase comparator (ϕ-DET) compares the phase of the divided frequency output f_N of the programmable divider with that of the reference frequency output f_r of the reference frequency generator, and outputs an up request signal ($\overline{UP}$) and down request signal ($\overline{DW}$).

(6) Unlock F/F

The unlock F/F detects the unlock status of the PLL frequency synthesizer from the up request signal ($\overline{UP}$) and down request signal ($\overline{DW}$) of the phase comparator (ϕ-DET).

(7) Charge pump

The charge pump outputs the result of the output of the phase comparator from the error out pins (EO0 and EO1 pins).

13.3 Registers Controlling PLL Frequency Synthesizer

The PLL frequency synthesizer is controlled by the following four registers.

- PLL mode select register (PLLMD)
- PLL reference mode register (PLLRF)
- PLL unlock F/F judge register (PLLUL)
- PLL data transfer register (PLLNS)

(1) PLL mode select register (PLLMD)

This register selects the input pin and division mode of the PLL frequency synthesizer.

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

Reset input clears this register to 00H.

In the STOP mode, only bits 3 and 2 (VCOHDMD and VCOLDMD) retain the previous value. Bits 1 and 0 (PLLMD1 and PLLMD0) are reset to 0.

In the HALT mode, it holds the value immediately before the HALT mode was set.

Figure 13-2. Format of PLL Mode Select Register (PLLMD)

Symbol	7	6	5	4	<3>	<2>	<1>	<0>	Address	After reset	R/W
PLLMD	0	0	0	0	VCOHDMD	VCOLDMD	PLLMD1	PLLMD0	FFA0H	00H	R/W

VCOH DMD	Selection of disable status of VCOH pin
0	Connected to pull-down resistor.
1	High-impedance state

VCOL DMD	Selection of disable status of VCOL pin
0	Connected to pull-down resistor.
1	High-impedance state

PLLMD1	PLLMD0	Selection of division mode of PLL frequency synthesizer and VCO input pin
0	0	Disables VCOL and VCOH pins[Note]
0	1	Direct division (VCOL pin and MF mode)
1	0	Pulse swallow (VCOH pin and VHF mode)
1	1	Pulse swallow (VCOL pin and HF mode)

Note This does not mean that the PLL is disabled. The VCOH and VCOL pins become the status specified by bit 3 (VCOHDMD) and bit 2 (VCOLDMD). The EO0 and EO1 pins go low.

Remark Bits 4 to 7 are fixed to 0 by hardware.

(2) PLL reference mode register (PLLRF)

This register selects the reference frequency f_r of the PLL frequency synthesizer and sets the disabled status of the PLL frequency synthesizer.

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

The value of this register is set to 0FH after reset and in the STOP mode.

In the HALT mode, it holds the value immediately before the HALT mode was set.

Figure 13-3. Format of PLL Reference Mode Register (PLLRF)

Symbol	7	6	5	4	<3>	<2>	<1>	<0>	Address	After reset	R/W
PLLRF	0	0	0	0	PLLRF3	PLLRF2	PLLRF1	PLLRF0	FFA1H	0FH	R/W

PLLRF3	PLLRF2	PLLRF1	PLLRF0	Setting of reference frequency f_r of PLL frequency synthesizer
0	0	0	0	50 kHz
0	0	0	1	25 kHz
0	0	1	0	12.5 kHz
0	0	1	1	9 kHz
0	1	0	0	1 kHz
0	1	0	1	3 kHz
0	1	1	0	10 kHz
0	1	1	1	Setting prohibited
1	×	×	×	PLL disable**Note**

Note When PLL disable is selected, the status of the VCOL, VCOH, EO0, and EO1 pins are as follows:

VCOH, VCOL pins: Status specified by bit 3 (VCOHDMD) and bit 2 (VCOLDMD) of the PLL mode select register (PLLMD).

EO0, EO1 pins: High-impedance state

Remark Bits 4 to 7 are fixed to 0 by hardware.

×: Don't care

(3) PLL unlock F/F judge register (PLLUL)

This register detects whether the PLL frequency synthesizer is in the unlock status.

Because this register is an R&RESET register, it is reset to 0 after it has been read.

Reset input sets this register to 0×H[Note 1].

In the STOP and HALT modes, this register holds the value immediately before the STOP or HALT mode was set.

Figure 13-4. Format of PLL Unlock F/F Judge Register (PLLUL)

Symbol	7	6	5	4	3	2	1	<0>	Address	After reset	R/W
PLLUL	0	0	0	0	0	0	0	PLLUL0	FFA2H	0×H[Note 1]	R[Note 2]

PLLUL0	Detection of status of unlock F/F
0	Unlock F/F = 0: PLL lock status
1	Unlock F/F = 1: PLL unlock status

Notes 1. The value of bit 0 (PLLUL0) at reset differs depending on the type of reset that has been executed (refer to the table below).

2. Bit 0 (PLLUL0) is R&Reset.

		7	6	5	4	3	2	1	0
After reset	Power-on clear	0	0	0	0	0	0	0	Undefined
	Watchdog timer	↓	↓	↓	↓	↓	↓	↓	Retained
	$\overline{\text{RESET}}$ input	↓	↓	↓	↓	↓	↓	↓	Retained
STOP mode		↓	↓	↓	↓	↓	↓	↓	Retained
HALT mode		↓	↓	↓	↓	↓	↓	↓	Retained

Remark Bits 1 to 7 are fixed to 0 by hardware.

(4) PLL data transfer register (PLLNS)

This register transfers the values of the PLL data registers (PLLRL, PLLRH, and PLLR0) to the programmable counter and swallow counter.

The value of this register is 00H after reset and in the STOP mode.

In the HALT mode, this register holds the previous value immediately before the HALT mode is set.

Figure 13-5. Format of PLL Data Transfer Register (PLLNS)

Symbol	7	6	5	4	3	2	1	<0>	Address	After reset	R/W
PLLNS	0	0	0	0	0	0	0	PLLNS0	FFA3H	00H	W

PLLNS0	Transfers value of PLL data register to programmable counter and swallow counter
0	Does not transfer
1	Transfers

Remark Bits 1 to 7 are fixed to 0 by hardware.

13.4 Operation of PLL Frequency Synthesizer

13.4.1 Operation of each block of PLL frequency synthesizer

(1) Operation of input select block and programmable divider

The input select block and programmable divider select the input pin and division mode of the PLL frequency synthesizer and divide the frequency in the selected division mode, according to the setting of the PLL mode select register (PLLMD).

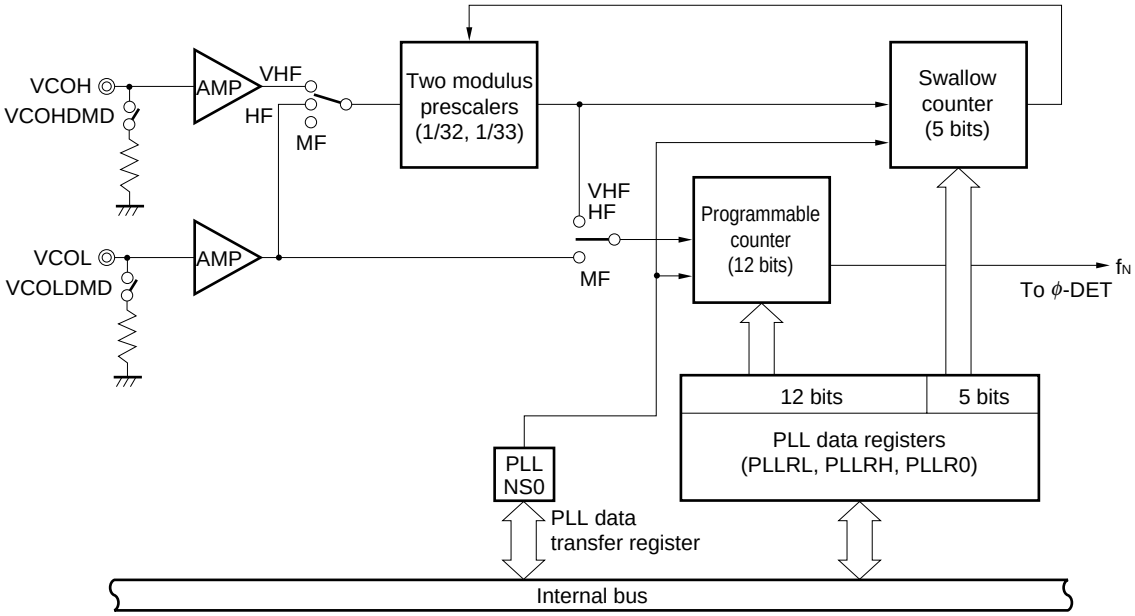
The programmable counter (12 bits) and pulse swallow counter (5 bits) are binary counters.

The division value (N value) is set to the programmable counter (12 bits) and swallow counter (5 bits) by the PLL data registers (PLLRL, PLLRH, and PLLR0).

When the N value has been transferred to the programmable counter and swallow counter, frequency division is performed in the selected division mode according to the status of bit 0 (PLLNS0) of the PLL data transfer register.

Figure 13-6 shows the configuration of the input select block and programmable divider.

Figure 13-6. Configuration of Input Select Block and Programmable Divider



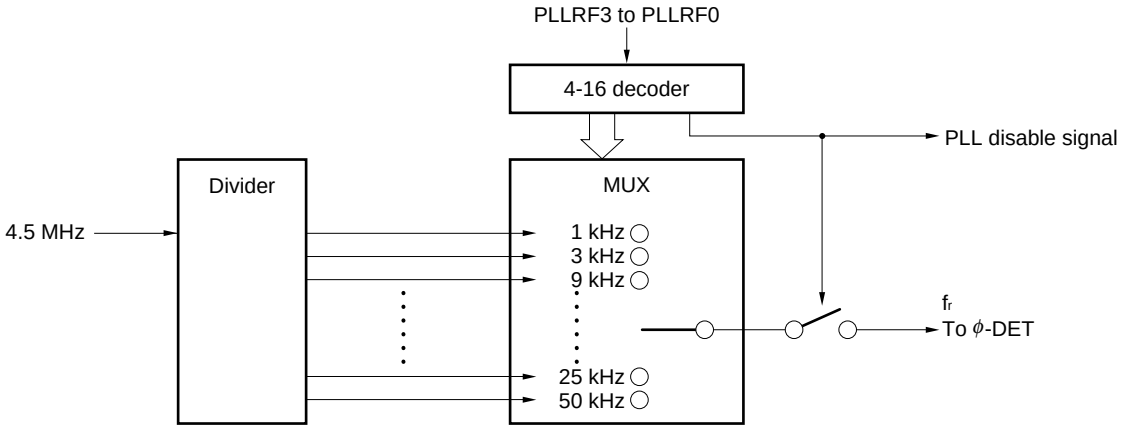
(2) Operation of reference frequency generator

The reference frequency generator divides the 4.5 MHz output of the crystal oscillator and generates seven types of reference frequency f_r for the PLL frequency synthesizer.

Reference frequency f_r is selected by the PLL reference mode register (PLLRF).

Figure 13-7 shows the configuration of the reference frequency generator.

Figure 13-7. Configuration of Reference Frequency Generator



(3) Operation of phase comparator (ϕ-DET)

Figure 13-8 shows the configuration of the phase comparator (ϕ-DET), charge pump, and unlock F/F.

The phase comparator (ϕ-DET) compares the phase of the divided frequency f_N of the programmable divider with that of the reference frequency f_r of the reference frequency generator, and outputs an up request signal, $\overline{UP}$, or a down request signal, $\overline{DW}$.

If the divided frequency f_N is lower than the reference frequency f_r, the up request signal is output. If f_N is higher than f_r, the down request signal is output.

Figure 13-9 shows the relation among reference frequency f_r, divided frequency f_N, up request signal $\overline{UP}$, and down request signal $\overline{DW}$.

When the PLL is disabled, neither the up nor the down request signal is output.

The up and down request signals are input to the charge pump and unlock F/F.

Figure 13-8. Configuration of Phase Comparator, Charge Pump, and Unlock F/F

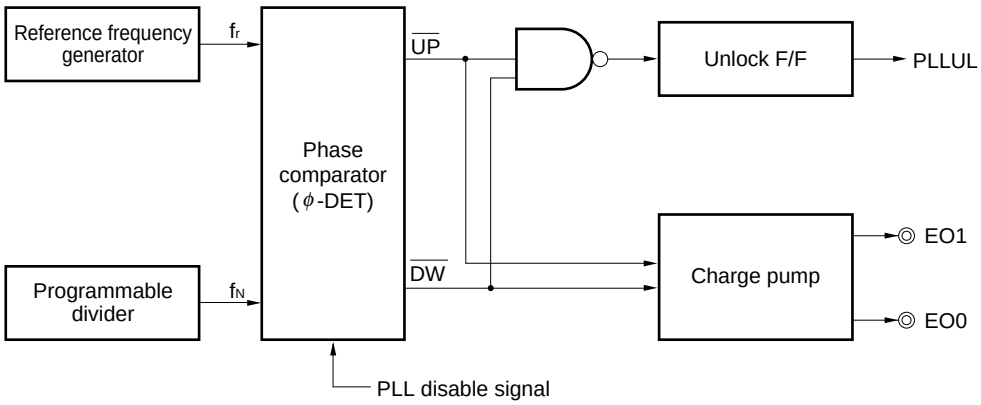
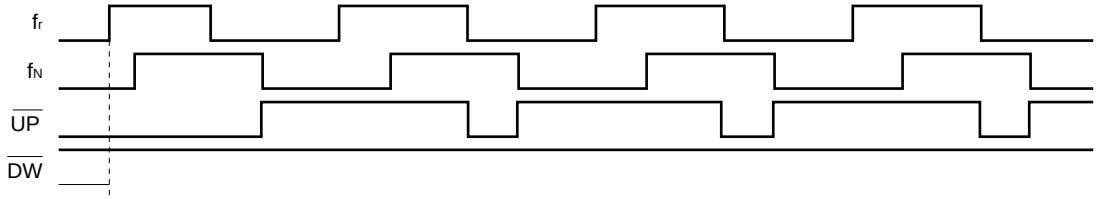
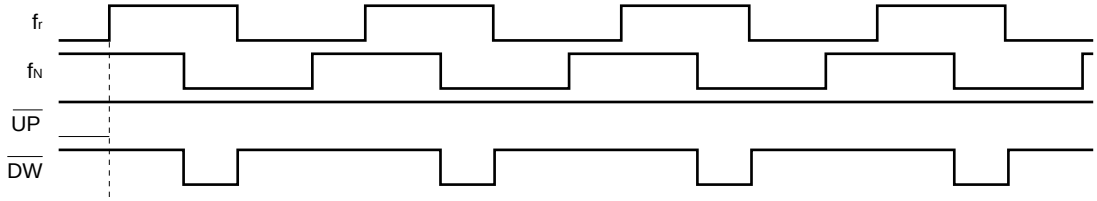


Figure 13-9. Relationship Between f_r, f_N, $\overline{UP}$, and $\overline{DW}$

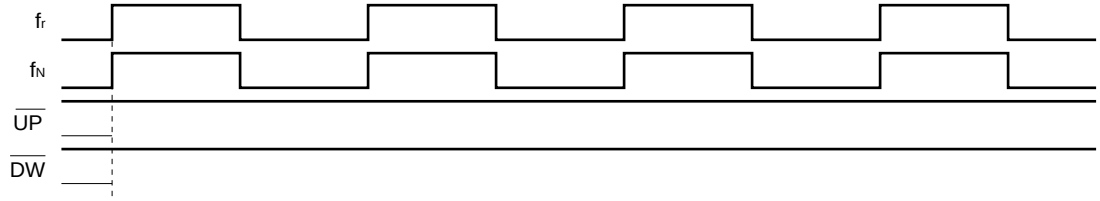
(a) If f_r advances f_N in phase



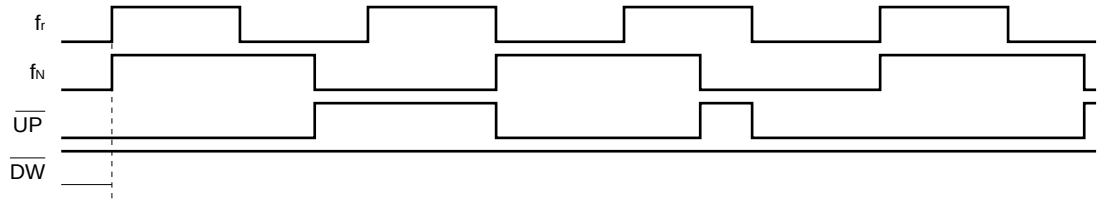
(b) If f_N advances f_r in phase



(c) If f_N and f_r are in phase



(d) If f_N is lower than f_r



(4) Operation of charge pump

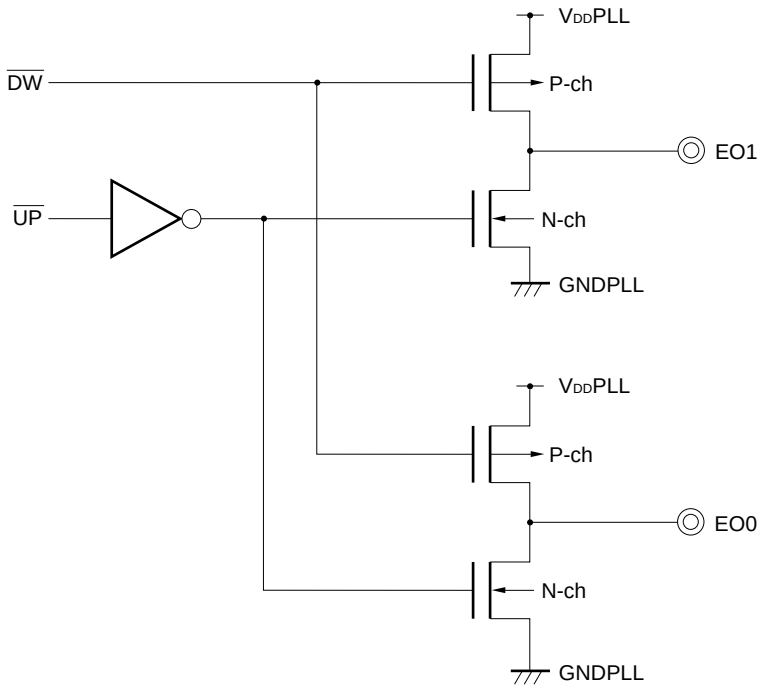
The charge pump outputs the result of the up request ($\overline{UP}$) or down request ($\overline{DW}$) signal from the phase comparator (ϕ-DET) from the error out pins (EO0 and EO1 pins). Table 13-3 shows the output signals. The EO0 and EO1 pins are of voltage-driven type pins.

Figure 13-10 shows the configuration of the error out pins.

Table 13-3. Error Out Output Signal

Relationship Between Divided Frequency f_N and Reference Frequency f_r	Error Out Output Signal
When $f_r > f_N$	Low level
When $f_r < f_N$	High level
When $f_r = f_N$	Floating (high impedance)

Figure 13-10. Configuration of Error Out Output



(5) Operation of unlock F/F

The unlock F/F detects the unlock status of the PLL frequency synthesizer.

The unlock status of the PLL frequency synthesizer is detected from the up request signal $\overline{UP}$ and down request signal $\overline{DW}$ of the phase comparator (ϕ-DET).

Because either of the up request or down request signal outputs a low level in the unlock status, the unlock status can be detected by using this low-level signal.

The status of the unlock F/F is detected by bit 0 (PLLUL0) of the PLL unlock F/F judge register (PLLUL).

The unlock F/F is set at the cycle of reference frequency f_r selected at that time.

The PLL unlock F/F judge register is reset when its contents have been read.

To read the PLLUL, therefore, it must be read at a cycle longer than the cycle ($1/f_r$) of the reference frequency.

13.4.2 Operation to set N value of PLL frequency synthesizer

The division value (N value) is set to the programmable counter (12 bits) and swallow counter (5 bits) by the PLL data registers (PLLRL, PLLRH, and PLLR0).

When the N value has been transferred to the programmable counter and swallow counter by bit 0 (PLLNS0) of the PLL data transfer register (PLLNS), frequency division is carried out in the selected division mode.

Examples of setting the N value in the respective division modes (MF, HF, and VHF) are shown below.

(1) Direct division mode (MF)

(a) Calculating division value N (value set to PLL data register)

$$N = \frac{f_{VCOL}}{f_r}$$

where, f_{VCOL}: Input frequency of V_{COL} pin

f_r: Reference frequency

(b) Example of setting PLL data register

An example of setting the PLL data register to receive broadcasting stations in the following MW band is shown below.

Receive frequency: 1422 kHz (MW band)

Reference frequency: 9 kHz

Intermediate frequency: 450 kHz

Division value N is calculated as follows:

$$N = \frac{f_{VCOL}}{f_r} = \frac{1422 + 450}{9} = 208 \text{ (decimal)}$$

$$= 0D0H \text{ (hexadecimal)}$$

Data is set to the PLL data registers (PLLR and PLLR0) as follows.

PLLR																PLLR0							
PLLRH								PLLRL								←PLLSCN							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0							
Programmable counter value												Don't care				Fixed to 0							
0	0	0	0	1	1	0	1	0	0	0	0												
0				D				0															

After setting the above PLL data registers (PLLR and PLLR0), data must be transferred to the programmable counter by setting bit 0 (PLLNS0) of the PLL data transfer register (PLLNS).

(2) Pulse swallow mode (HF)

(a) Calculating division value N (value set to PLL data register)

$$N = \frac{f_{VCOL}}{f_r}$$

where, f_{VCOL}: Input frequency of V_{COL} pin

f_r: Reference frequency

(b) Example of setting PLL data register

An example of setting the PLL data register to receive broadcasting stations in the following SW band is shown below.

Receive frequency: 25.50 MHz (SW band)

Reference frequency: 10 kHz

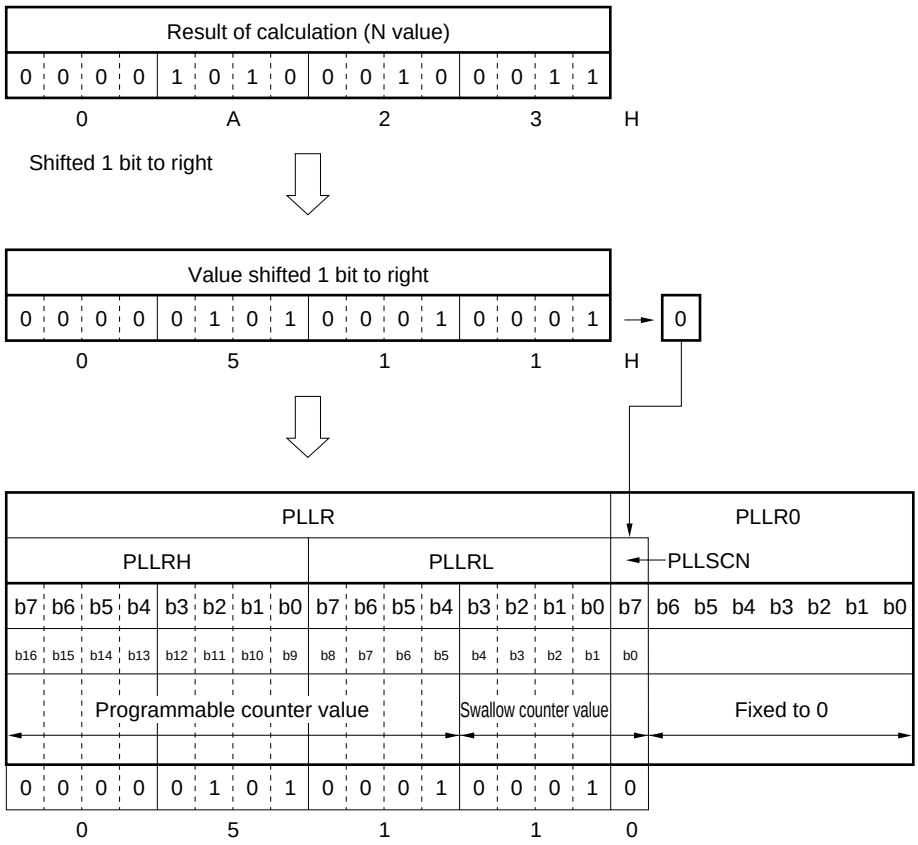
Intermediate frequency: 450 kHz

Division value N is calculated as follows:

$$N = \frac{f_{VCOL}}{f_r} = \frac{25500 + 450}{10} = 2595 \text{ (decimal)}$$

$$= 0A23H \text{ (hexadecimal)}$$

Because the least significant bit of the division value N must be set to bit 7 (PLLSCN) of PLL data register 0 (PLLR0), data must be set by shifting the result of the above calculation 1 bit to the right.
Data is set to the PLL data registers (PLLR and PLLR0) as follows.



After setting the above PLL data registers (PLLR and PLLR0), data must be transferred to the programmable counter and swallow counter by setting bit 0 (PLLNS0) of the PLL data transfer register (PLLNS).
In this example, a value of half the N value is set to the high-order 16 bits of the PLL data register (PLLR) by shifting the N value resulting from calculation 1 bit to the right.
If the N value is calculated as follows with the least significant bit of the N value in PLLSCN fixed to 0, the result of the calculation (N_{PLLR}) can be set to the PLL data register (PLLR) as is.
If the calculation result is set in this way, however, the input frequency (f_{VCOL}) is $2 \times f_r$ (reference frequency) of the set value N_{PLLR}.

$$N_{PLLR} = \frac{f_{VCOL}}{2f_r}$$

(3) Pulse swallow mode (VHF)

(a) Calculating division value N (value set to PLL data register)

$$N = \frac{f_{VCOH}}{f_r}$$

where, f_{VCOH}: Input frequency of VCOH pin

f_r: Reference frequency

(b) Example of setting PLL data register

An example of setting the PLL data register to receive broadcasting stations in the following FM band is shown below.

Receive frequency: 100.0 MHz (FM band)

Reference frequency: 50 kHz

Intermediate frequency: +10.7 MHz

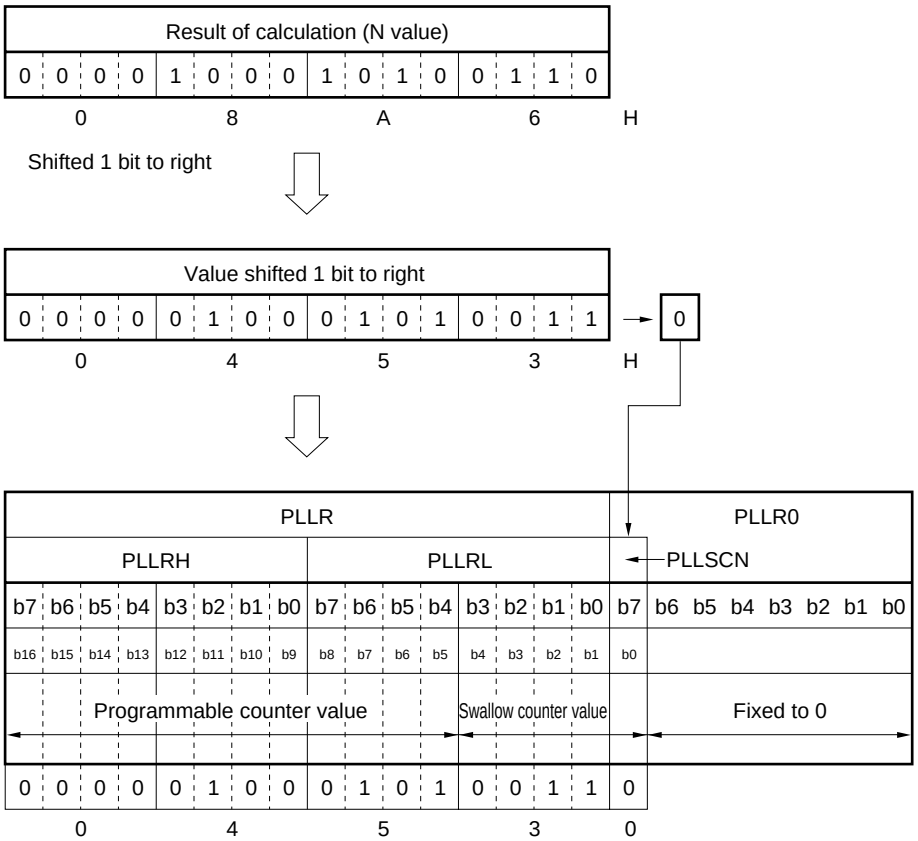
Division value N is calculated as follows:

$$N = \frac{f_{VCOH}}{f_r} = \frac{100.0 + 10.7}{0.05} = 2214 \text{ (decimal)}$$

$$= 08A6H \text{ (hexadecimal)}$$

Because the least significant bit of the division value N must be set to the PLL data register 0 (PLLR0), data must be set by shifting the value calculated by the above expression 1 bit to the right.

Data is set to the PLL data registers (PLLR and PLLR0) as follows.



After setting the above PLL data registers (PLLR and PLLR0), data must be transferred to the programmable counter and swallow counter by setting bit 0 (PLLNS0) of the PLL data transfer register (PLLNS).

In this example, a value of half the N value is set to the higher 16 bits of the PLL data register (PLLR) by shifting the N value resulting from calculation 1 bit to the right.

If the N value is calculated as follows with the least significant bit of the N value in PLLSCN fixed to 0, the result of the calculation (N_{PLLR}) can be set to the PLL data register (PLLR) as is.

If the calculation result is set in this way, however, the input frequency (f_{VCOH}) is $2 \times f_r$ (reference frequency) of the set value N_{PLLR}.

$$N_{PLLR} = \frac{f_{VCOH}}{2f_r}$$

13.5 PLL Disable Status

The PLL frequency synthesizer can be stopped (PLL disabled status) by performing any of the following settings while the PLL frequency synthesizer is operating.

- Setting value of bit 3 (PLLRF3) of the PLL reference mode register (PLLRF) to 1 to set PLL disabled status
- Setting STOP mode with the STOP instruction
- Setting reset status with the reset function

The following table shows the operation of each block and the status of each register in the PLL disabled status.

Table 13-4. Operation of Each Block and Register Status in PLL Disabled Status

Block/Register	Status in PLL Disabled Status
VCOL and VCOH pins	Status set in bit 3 (VCOHDMD) and bit 2 (VCOLDMD) of PLLMD
Programmable divider	Division stops
Reference frequency generator	Output stops
Phase comparator	Output stops
EO0 and EO1 pin	High impedance
PLL mode select register	Retains value on execution of write instruction
PLL data register	
PLL unlock F/F judge register	

13.6 Notes on PLL Frequency Synthesizer

- **Notes on using PLL frequency synthesizer**
  Because the input pins (VCOL and VCOH pins) of the PLL frequency synthesizer are provided with an AC amplifier, cut the DC component of the input signal by connecting a capacitor to the input pins in series.
  The potential of the selected input pin is intermediate (about $1/2V_{DD}$). The input pin not selected becomes the status set in bit 3 (VCOHDMD) and bit 2 (VCOLDMD) of the PLL mode select register (PLLMD).
  For the frequencies that can be actually input and input amplitude, refer to **CHAPTER 19 ELECTRICAL SPECIFICATIONS**.

CHAPTER 14 FREQUENCY COUNTER

14.1 Function of Frequency Counter

The frequency counter counts the intermediate frequency (IF) of a tuner.

The intermediate frequency input to the FMIFC or AMIFC pin is counted for a specific time (1 ms, 4 ms, 8 ms, or open) by a 16-bit counter. The count value of the frequency counter is stored in the IF counter register.

For the range of the frequency that can be input to the FMIFC and AMIFC pins, refer to **CHAPTER 19 ELECTRICAL SPECIFICATIONS**.

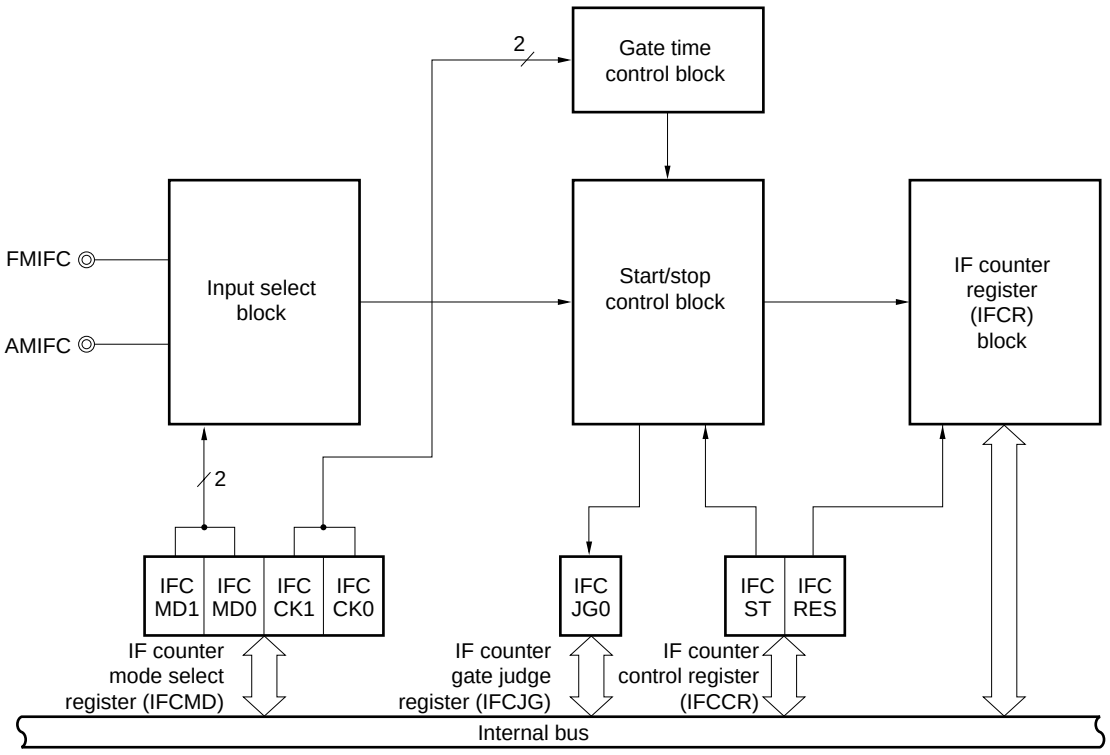
14.2 Configuration of Frequency Counter

The frequency counter consists of the following hardware.

Table 14-1. Configuration of Frequency Counter

Item	Configuration
Counter register	IF counter register (IFCR)
Control registers	IF counter mode select register (IFCMD) IF counter control register (IFCR) IF counter gate judge register (IFCJG)

Figure 14-1. Block Diagram of Frequency Counter



(1) IF counter input select block

The IF counter input select block selects the pin to be used from the FMIFC and AMIFC pins, and a count mode.

(2) Gate time control block

The gate time control block sets a gate time (count time).

(3) Start/stop control block

The start/stop control block starts counting by the IF counter register and detects the end of counting.

(4) IF counter register block

The IF counter register block is a 16-bit register that counts up the frequency input in the set gate time. The count value is stored to the IF counter register (IFCR). When the count value reaches FFFFH, the IF counter register holds FFFFH at the next input, and stops counting. The value of this register is reset to 0000H after reset or in the STOP mode. In the HALT mode, it holds the current count value.

14.3 Registers Controlling Frequency Counter

The frequency counter is controlled by the following three registers.

- IF counter mode select register (IFCMD)
- IF counter control register (IFCCR)
- IF counter gate judge register (IFCJG)

(1) IF counter mode select register (IFCMD)

This register selects the input pin of the frequency counter, and selects a mode and gate time (count time).

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

The value of this register is reset to 00H after reset or in the STOP mode.

In the HALT mode, this register holds the value immediately before the HALT mode is set.

Figure 14-2. Format of IF Counter Mode Select Register (IFCMD)

Symbol	7	6	5	4	<3>	<2>	<1>	<0>	Address	After reset	R/W
IFCMD	0	0	0	0	IFCMD1	IFCMD0	IFCCK1	IFCCK0	FFA9H	00H	R/W

IFCMD1	IFCMD0	Selection of frequency counter pin and mode
0	0	Disables FMIFC, AMIFC pins[Note]
0	1	AMIFC pin, AMIF count mode
1	0	FMIFC pin, FMIF count mode
1	1	FMIFC pin, AMIF count mode

IFCCK1	IFCCK0	Selection of gate time
0	0	1 ms
0	1	4 ms
1	0	8 ms
1	1	Open

Note The FMIFC and AMFIC pins are in a high-impedance state.

Remark Bits 4 to 7 are fixed to 0 by hardware.

(2) IF counter control register (IFCCR)

This register starts counting by the IF counter register and clears the IF counter register.

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

The value of this register is reset to 00H after reset and in the STOP mode.

In the HALT mode, this register holds the value immediately before the HALT mode is set.

Figure 14-3. Format of IF Counter Control Register (IFCCR)

Symbol	7	6	5	4	3	2	<1>	<0>	Address	After reset	R/W
IFCCR	0	0	0	0	0	0	IFCST	IFCRES	FFACH	00H	W

IFCST	Setting of IF counter register start
0	Nothing is affected
1	Starts counting

IFCRES	Setting of data clear of IF counter register
0	Nothing is affected
1	Clears data of IF counter register

Remark Bits 2 to 7 are fixed to 0 by hardware.

(3) IF counter gate judge register (IFCJG)

This register detects opening/closing of the gate of the frequency counter.

The value of this register is reset to 00H after reset and in the STOP mode.

In the HALT mode, this register holds the value immediately before the HALT mode is set.

Figure 14-4. Format of IF Counter Gate Judge Register (IFCJG)

Symbol	7	6	5	4	3	2	1	<0>	Address	After reset	R/W
IFCJG	0	0	0	0	0	0	0	IFCJG0	FFABH	00H	R

IFCJG0	Detection of opening/closing of frequency counter gate
0	Gate is closed
1	• If gate time is set to other than open Status until gate is closed after IFCST has been set to 1 • If gate time is set to open Status where gate is open as soon as it has been set to be opened

Remark Bits 1 to 7 are fixed to 0 by hardware.

Caution IFCJG0 remains set even if the IF counter register overflows and stops counting, until the set gate time expires.

14.4 Operation of Frequency Counter

<1> Select an input pin, mode, and gate time using the IF counter mode select register (IFCMD).
Figure 14-5 shows a block diagram of input pin and mode selection.

<2> Set bit 0 (IFCRES) of the IF counter control register (IFCCR) to 1, and clear the data of the IF counter register.

<3> Set bit 1 (IFCST) of the IF counter control register (IFCCR) to 1.

<4> The gate is opened only for the set gate time since a 1 kHz internal signal has risen after IFCST was set. If the gate time is set to be opened, the gate is opened as soon as it has been specified to be opened.
Bit 0 (IFCJG0) of the IF counter gate judge register (IFCJG) is automatically set to 1 as soon as IFCST has been set to 1.
When the gate time has elapsed, bit 0 (IFCJG0) of the IF counter gate judge register (IFCJG) is automatically cleared to 0. If it is specified that the gate be open, however, IFCJG0 is not automatically cleared. In this case, set a gate time. Figure 14-6 shows the gate timing of the frequency counter.

<5> While the gate opens the frequency input to the selected FMIFC or AMIFC pin, the IF counter register counts the frequency.
If the FMIFC pin is used in the FMIF count mode, however, the input frequency is divided by half before it is counted.

The relationship between the count value x (decimal), the input frequencies (f_{FMIFC} and f_{AMIFC}), and the gate time (T_{GATE}) is shown below.

- FMIF count mode (FMIFC pin)

$$f_{FMIFC} = \frac{x}{T_{GATE}} \times 2 \text{ (kHz)}$$

- AMIF count mode (FMIFC or AMIFC pin)

$$f_{AMIFC} = \frac{x}{T_{GATE}} \text{ (kHz)}$$

Figure 14-5. Block Diagram of Input Pin and Mode Selection

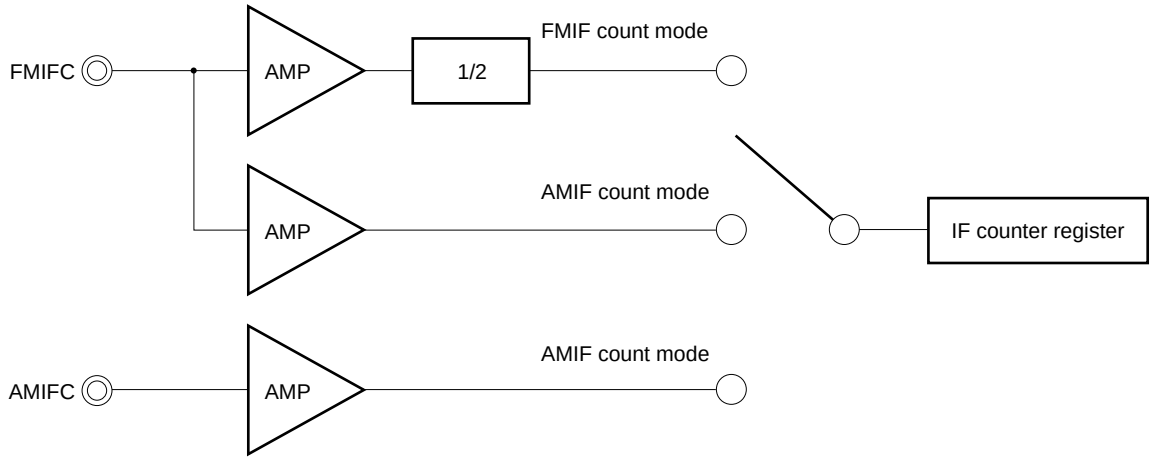
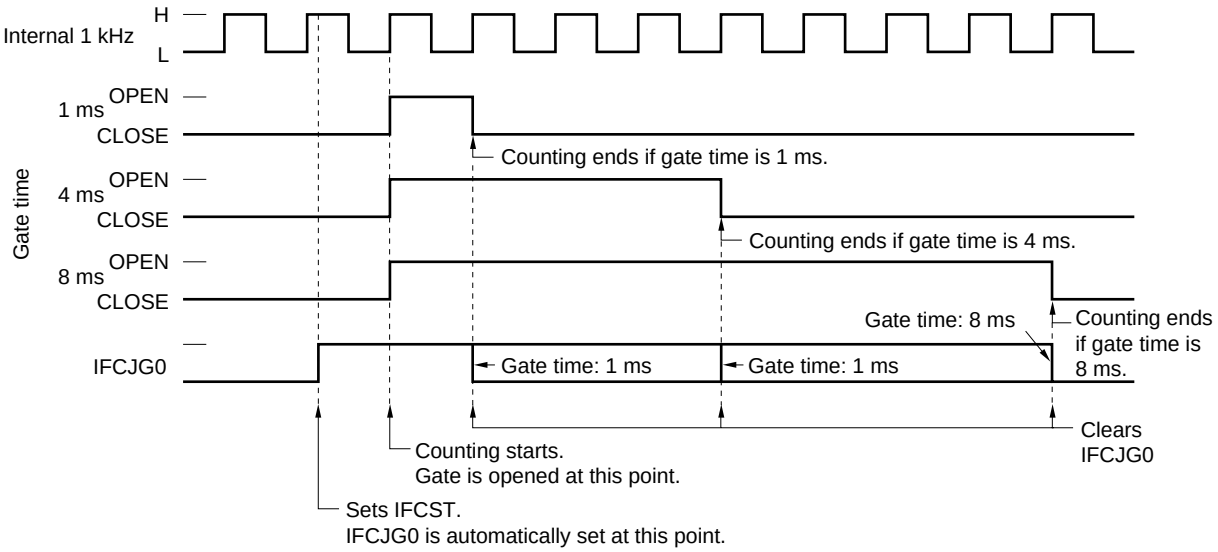
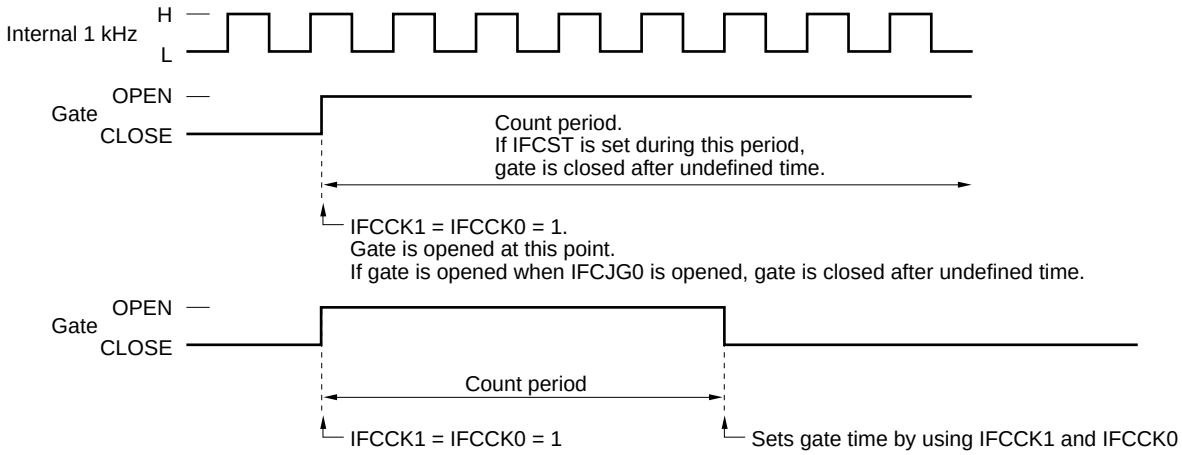


Figure 14-6. Gate Timing of Frequency Counter

(a) If gate time is set to 1, 4, or 8 ms



(b) If gate is set to be open



Caution If counting is started by using IFCST while this gate is open, the gate is closed after an undefined time. To open the gate, therefore, do not set IFCST to 1.

Remark IFCST: Bit 1 of IF counter control register (IFCCR)
IFCJG0: Bit 0 of IF counter gate judge register (IFCJG)
IFCCK1, 0: Bits 1 and 0 of IF counter mode select register (IFCMD)

14.5 Notes on Frequency Counter

(1) Notes on using frequency counter

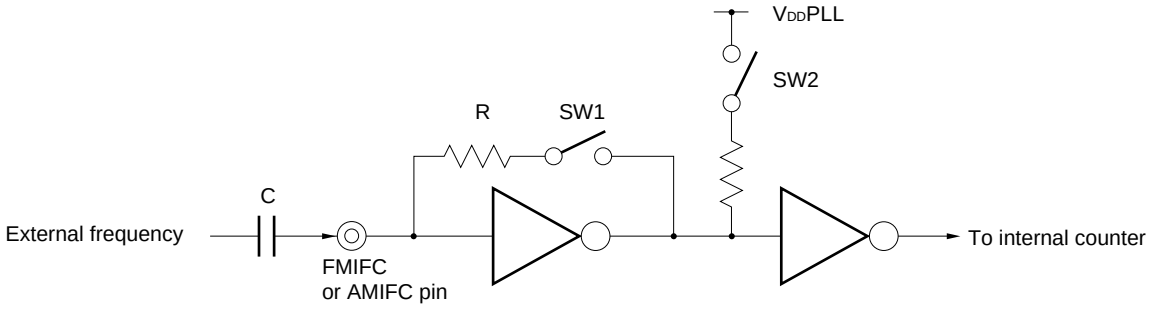
Because signals are input to the frequency counter from an input pin (FMIFC or AMIFC pin) with an AC amplifier as shown in Figure 14-7, cut the DC component of the input signals by using capacitor C.

If the FMIFC or AMIFC pin is selected by the IF counter mode select register, switch SW1 turns ON, and switch SW2 turns OFF. As a result, the voltage on the pin is about $1/2V_{DD}$.

Unless the voltage has risen to a sufficient intermediate level at this time, counting may not be performed normally because the AC amplifier is not in the normal operating range.

Therefore, make sure that sufficient wait time elapses after a pin has been selected and before counting is started (IFCST = 1).

Figure 14-7. Frequency Counter Input Pin Circuit



(2) Notes in HALT mode

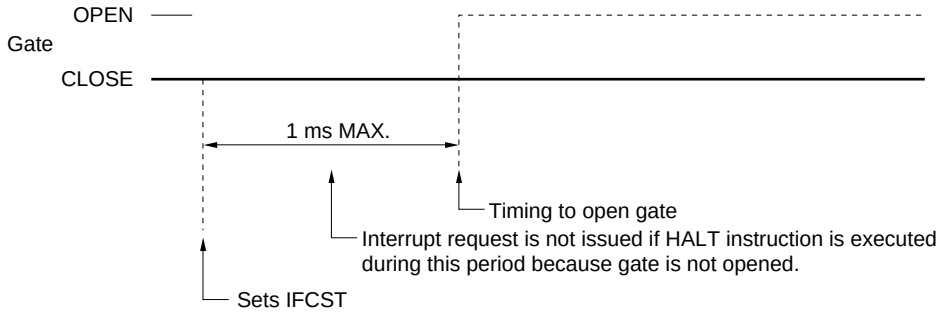
The FMIFC and AMIFC pins hold the status immediately before the HALT status was set.

To release the HALT mode by using the interrupt of the frequency counter at this time, the following point must be noted.

The gate will not be opened if the HALT instruction is executed after counting has been started by IFCST before the gate is actually opened.

Therefore, wait for at least 1 ms before executing the HALT instruction.

Figure 14-8. Gate Status When HALT Instruction Is Executed



(3) Error of frequency counter

The error of the frequency counter includes an error of gate time and a count error.

(1) Error of gate time

The gate time of the frequency counter is created by dividing 4.5 MHz.

Therefore, if 4.5 MHz is shifted “+x” ppm, the gate time is also shifted “–x” ppm.

(2) Count error

The frequency counter counts the frequency at the rising edge of the input signal.

If a high level is input to the pin when the gate is opened, therefore, one excess pulse is counted. When the gate is closed, however, counting is not affected by the status of the pin.

Therefore, the count error is “maximum + 1”.

CHAPTER 15 STANDBY FUNCTION

15.1 Standby Function and Configuration

15.1.1 Standby function

The standby function is designed to decrease power consumption of the system. The following two modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. The HALT mode is intended to stop the CPU operation clock. System clock oscillator continues oscillation. In this mode, current consumption cannot be decreased as in the STOP mode. The HALT mode is valid to restart immediately upon interrupt request and to carry out intermittent operations such as in watch applications.

Although the CPU stops operating, the peripheral functions can operate. To lower the current consumption, therefore, stop all unnecessary circuits before executing the HALT instruction.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the system clock oscillator stops and the whole system stops. CPU current consumption can be considerably decreased.

Data memory low-voltage hold (down to V_{DD} = 2.2 V) is possible. Thus, the STOP mode is effective to hold data memory contents with ultra-low current consumption.

If the supply voltage drops below 2.2 V, the system is reset by means of power-on clear reset. For reset, refer to **CHAPTER 16 RESET FUNCTION**.

Because this mode can be cleared upon interrupt request, it enables intermittent operations to be carried out. However, because a wait time is necessary to secure an oscillation stabilization time after the STOP mode is cleared, select the HALT mode if it is necessary to start processing immediately upon interrupt request.

All the functions stop operating.

Some registers of the PLL frequency synthesizer and frequency counter are reset, but the other functions are stopped with their current status retained.

Cautions 1. When shifting to the STOP mode, be sure to stop the peripheral hardware operation before executing the STOP instruction.

2. The following sequence is recommended for power consumption reduction of the A/D converter: first clear bit 7 (ADCS3) of ADM3 to 0 to stop the A/D conversion operation, then execute the HALT or STOP instruction.

15.1.2 Register controlling standby function

A wait time after the STOP mode is released upon interrupt request until the oscillation stabilizes is controlled with the oscillation stabilization time select register (OSTS).

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

Reset input sets OSTS to 04H.

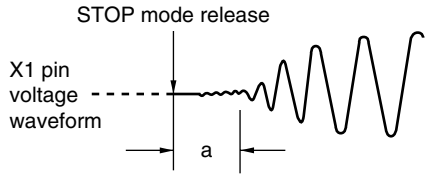
Figure 15-1. Format of Oscillation Stabilization Time Select Register (OSTS)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFAH	04H	R/W

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time when STOP mode is released
0	0	0	2^{12}/fx (910 μs)
0	0	1	2^{14}/fx (3.64 ms)
0	1	0	2^{15}/fx (7.28 ms)
0	1	1	2^{16}/fx (14.6 ms)
1	0	0	2^{17}/fx (29.1 ms)
Other than above			Setting prohibited

Remark fx: System clock oscillation frequency
( ): fx = 4.5 MHz

Caution The wait time when the STOP mode is released does not include the time required for the clock oscillation to start after the STOP mode has been released (see "a" in the figure below), regardless of whether the mode has been released by the $\overline{\text{RESET}}$ signal or an interrupt request.



15.2 Operations of Standby Function

15.2.1 HALT mode

(1) HALT mode set and operating status

The HALT mode is set by executing the HALT instruction.

The operating status in the HALT mode is described below.

Table 15-1. HALT Mode Operating Status

Item	Status
Clock generator	Can oscillate system clock. Stops clock supply to CPU.
CPU	Stops operating.
Port	Holds status before HALT mode is set.
8-bit timer/event counter	Holds operation before HALT mode is set and can operate.
Basic timer	
Watchdog timer	
Buzzer output controller	
A/D converter	Retains operation performed when HALT mode is set. However, comparison cannot be performed correctly in A/D conversion operation mode. In power-fail comparison mode, operation is as follows depending on setting of bit 5 (PFHRM3) of power-fail comparison mode register 3 (PFM3): • PFHRM3 = 0: Comparison cannot be performed normally. • PFHRM3: Power-fail comparison operation can be performed.
Serial interface (SIO30 to SIO32)	Retains operation performed when HALT mode is set and can operate.
External interrupt	Hold operation before HALT mode is set and can operate.
PLL frequency synthesizer	
Frequency counter	Retains operation performed before HALT mode is set. However, operation is not performed correctly though it is continued.
Power-on clear circuit	Reset when voltage of less than 3.5 V is detected.

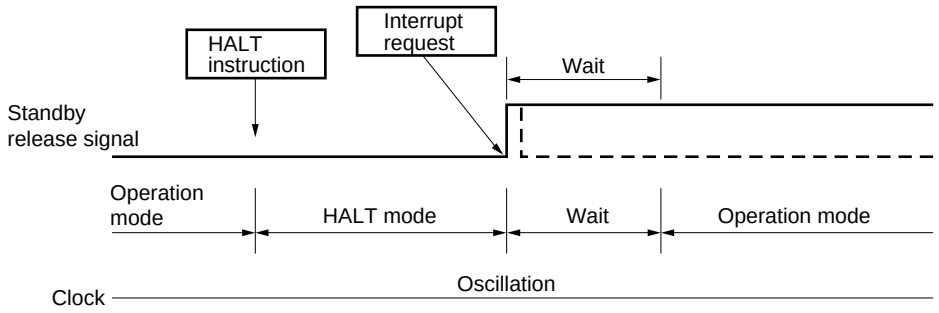
(2) HALT mode release

The HALT mode can be released by the following three types of sources.

(a) Release upon unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledgement is enabled, vectored interrupt servicing is carried out. If disabled, the next address instruction is executed.

Figure 15-2. HALT Mode Release upon Interrupt Generation



Remarks 1. The broken lines indicate the case when the interrupt request that released the standby status is acknowledged.

2. Wait time will be as follows:

- When vectored interrupt servicing is carried out: 8 to 9 clocks
- When vectored interrupt servicing is not carried out: 2 to 3 clocks

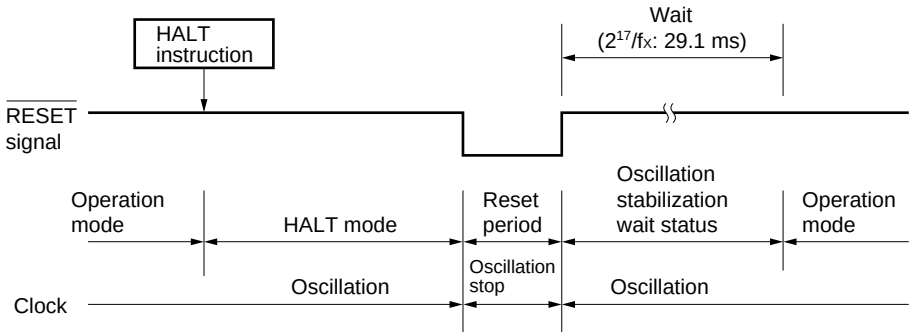
(b) Release upon non-maskable interrupt request

When a non-maskable interrupt is generated, the HALT mode is released and vectored interrupt servicing is carried out whether interrupt acknowledgement is enabled or disabled.

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

If the $\overline{\text{RESET}}$ signal is input, the HALT mode is released. As is the case with normal reset operation, the program is executed after branch to the reset vector address.

Figure 15-3. HALT Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fx: System clock oscillation frequency

2. ( ): fx = 4.5 MHz

Table 15-2. Operation After HALT Mode Release

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

Remark ×: Don't care

15.2.2 STOP mode

(1) STOP mode set and operating status

The STOP mode is set by executing the STOP instruction.

Cautions 1. When the STOP mode is set, the X1 pin is pulled down to GND, and the X2 pin is internally pulled up to V_{DD} to minimize the leakage current at the crystal oscillator block.

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 released 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 15-3. STOP Mode Operating Status

Item	Status
Clock generator	Can oscillate system clock. Stops clock supply to CPU.
CPU	Stops operating.
Port	Holds status before HALT mode is set.
8-bit timer/event counter	Operation stops and cannot operate.
Basic timer	
Watchdog timer	
Buzzer output controller	
A/D converter	
Serial interface (SIO30 to SIO32)	
External interrupt	Can operate.
PLL frequency synthesizer	Operation stops and cannot operate.
Frequency counter	
Power-on clear circuit	$\overline{\text{RESET}}$ generated when detecting 2.2 V or less.

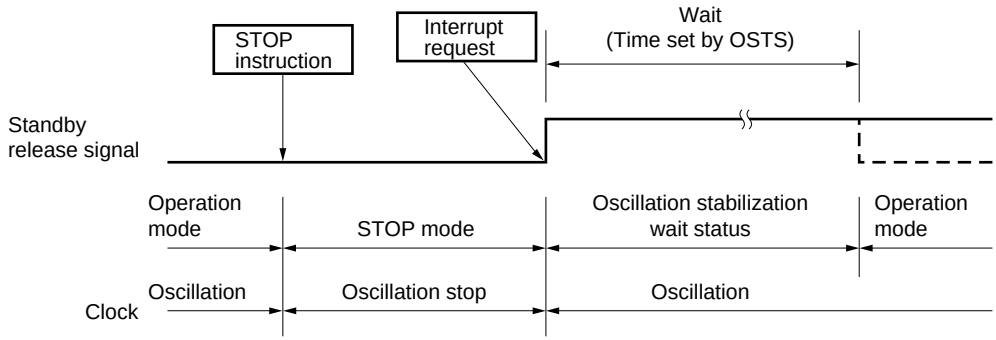
(2) STOP mode release

The STOP mode can be released by the following two types of sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the STOP mode is released. If interrupt request acknowledgement is enabled after the lapse of oscillation stabilization time, vectored interrupt servicing is carried out. If interrupt request acknowledgement is disabled, the next address instruction is executed.

Figure 15-4. STOP Mode Release by Interrupt Request Generation

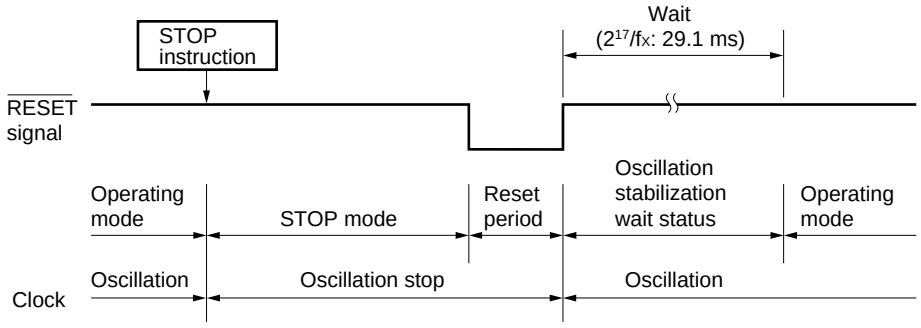


Remark The broken lines indicate the case when the interrupt request that released the standby status is acknowledged.

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

If the $\overline{\text{RESET}}$ signal is input, the STOP mode is released, and after the lapse of oscillation stabilization time, a reset operation is carried out.

Figure 15-5. Release by STOP Mode $\overline{\text{RESET}}$ Input



Remarks 1. f_X: System clock oscillation frequency

2. ( ): f_X = 4.5 MHz

Table 15-4. Operation After STOP Mode Release

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

Remark ×: Don't care

CHAPTER 16 RESET FUNCTION

16.1 Reset Function

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

(1) External reset input via a $\overline{\text{RESET}}$ pin
(2) Internal reset by inadvertent program loop time detection watchdog timer
(3) Internal reset by power-on clear (POC)

(1) External reset input by $\overline{\text{RESET}}$ pin

When a low level is input to the $\overline{\text{RESET}}$ pin, the device is reset, and each hardware unit enters the status shown in Table 16-1. While the reset signal is input and during the oscillation stabilization time immediately after the $\overline{\text{RESET}}$ signal has been deasserted, each pin goes into a high-impedance state (however, the P130 to P132 pins become low level, and the VCOH and VCOL pins are pulled down).

The $\overline{\text{RESET}}$ signal is deasserted when a high level is input to the $\overline{\text{RESET}}$ pin, and the program execution is started after the oscillation stabilization time ($2^{17}/f_X$) has elapsed.

(2) Internal reset by inadvertent program loop time detection of watchdog timer

Reset is effected and each hardware unit enters the status shown in Table 16-1 when the watchdog timer overflow. While reset is in effect and during the oscillation stabilization time immediately after the effect of reset has been cleared, each pin goes into a high-impedance state (however, the P130 to P132 pins become low level, and the VCOH and VCOL pins are pulled down).

Reset by the watchdog timer is cleared immediately after reset has been effected, and the program execution is started after the oscillation stabilization time ($2^{17}/f_X$) has elapsed.

(3) Internal reset by power-on clear (POC)

Reset is effected by means of power-on clear under the following conditions:

- If supply voltage is less than 3.5 V**Note** on power application
- If supply voltage drops to less than 2.2 V**Note** in STOP mode
- If supply voltage drops to less than 3.5 V**Note** (including HALT mode)

When these reset conditions of power-on clear are satisfied, reset is effected, and each hardware unit enters the status shown in Table 16-1. While the reset signal is input and during the oscillation stabilization time immediately after the reset signal has been deasserted, each pin goes into a high-impedance state (the P130 to P132 pins become low level, however).

Reset by power-on clear is cleared if the supply voltage rises beyond a specific level, and the program execution is started after the oscillation stabilization time ($2^{17}/f_X$) has elapsed.

Note These voltage values are maximum values. Actually, reset is effected at a voltage lower than these.

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, system clock oscillation remains stopped.

3. When the STOP mode is released by $\overline{\text{RESET}}$ input, the STOP mode register contents are held during reset input. However, the I/O port pin becomes high-impedance. Output dedicated port pin (P130 to P132) becomes low level regardless of the previous status.

Figure 16-1. Reset Function Block Diagram

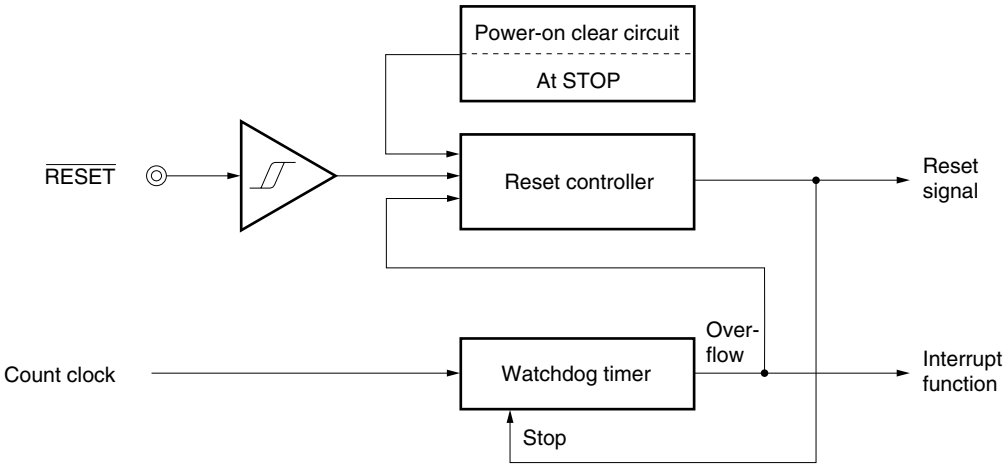
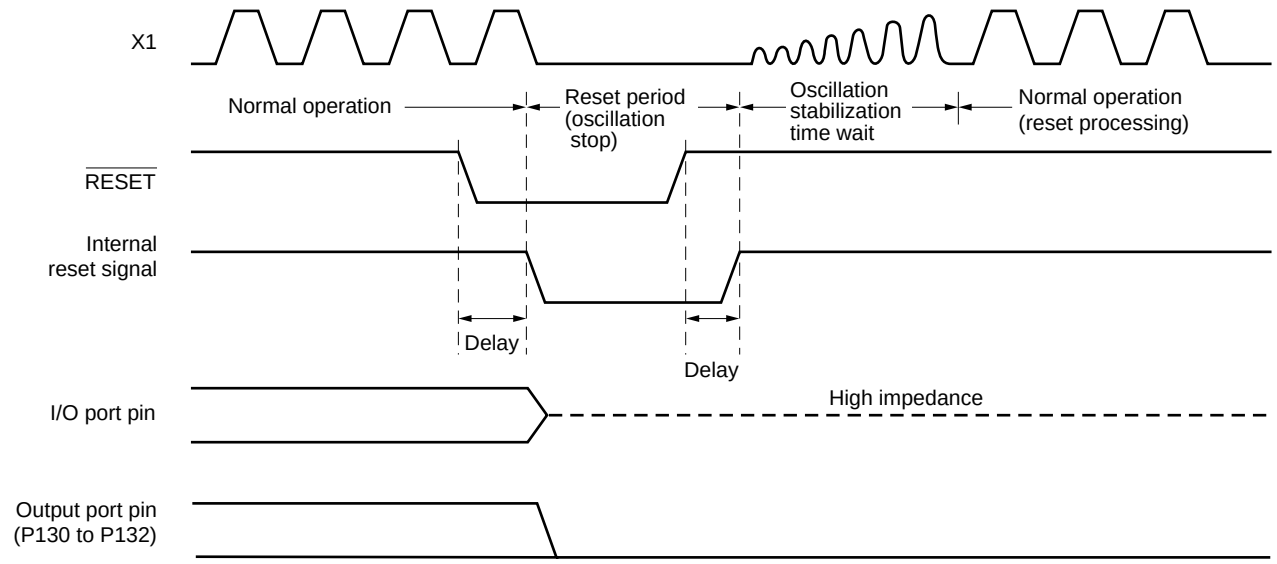


Figure 16-2. Timing of Reset by $\overline{\text{RESET}}$ Input

(a) In normal operation mode



(b) In STOP mode

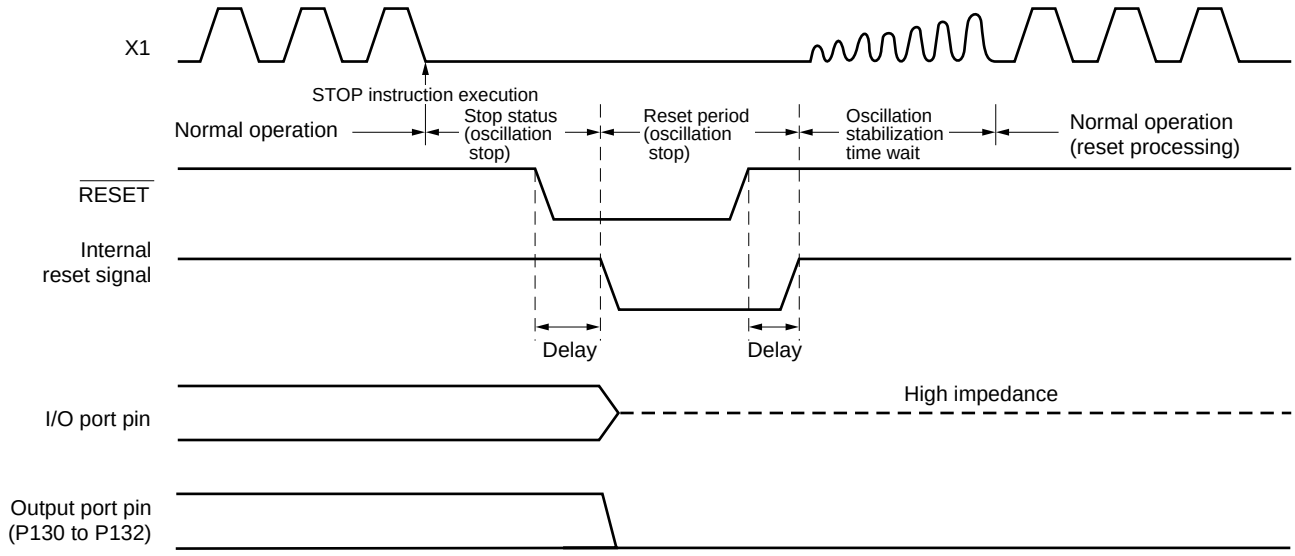


Figure 16-3. Timing of Reset due to Watchdog Timer Overflow

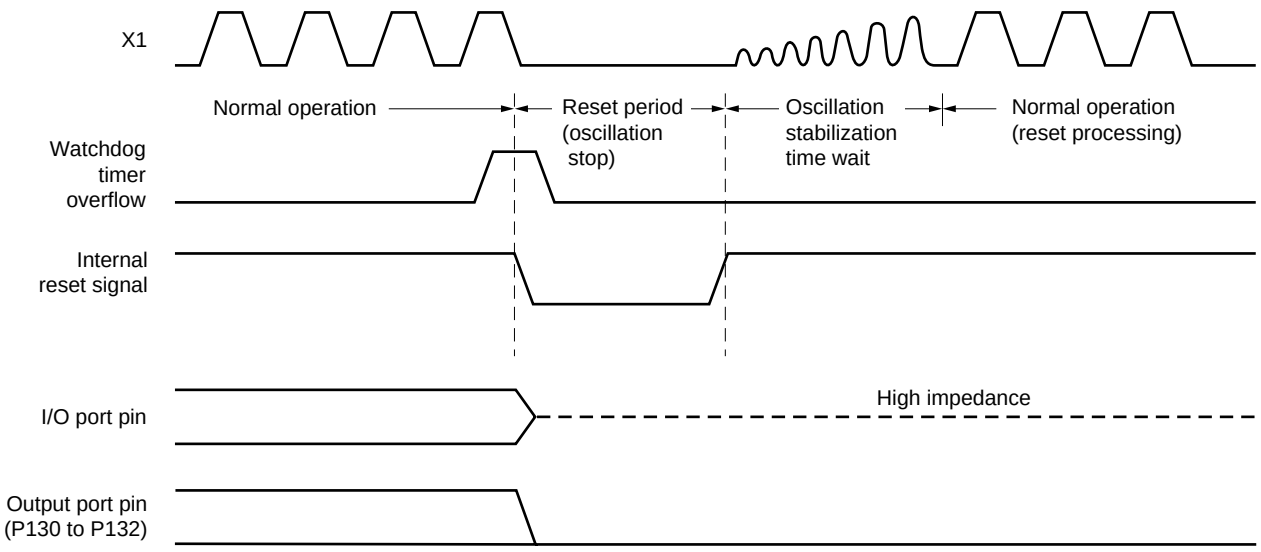
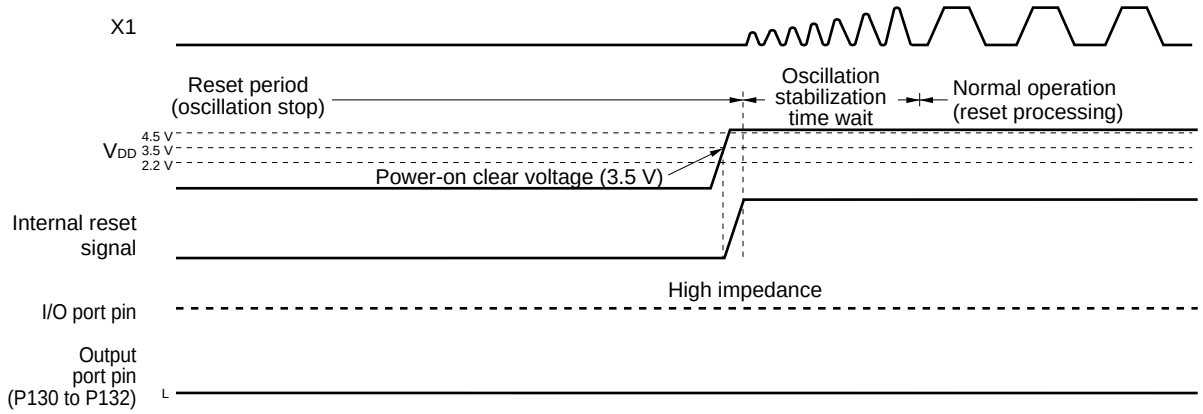
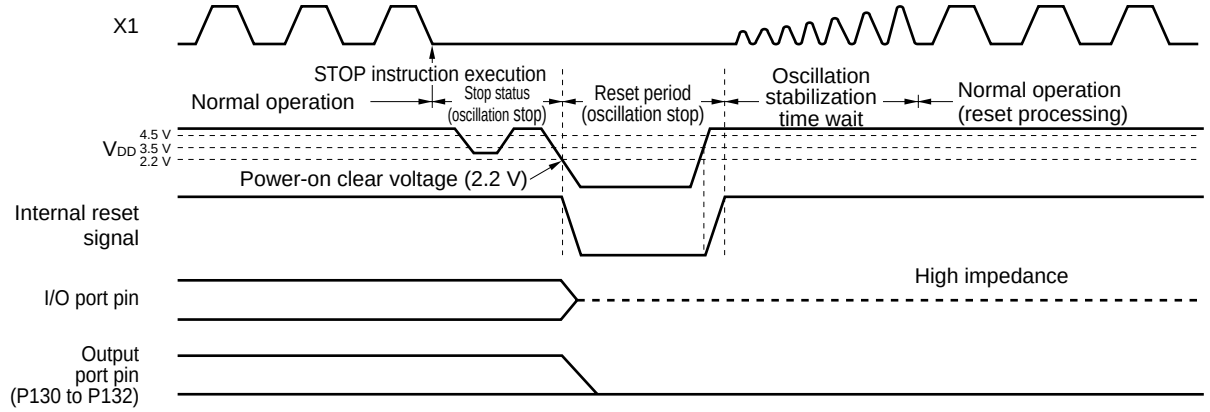


Figure 16-4. Timing of Reset by Power-on Clear

(a) At power application



(b) In STOP mode



(c) In normal operating mode (including HALT mode)

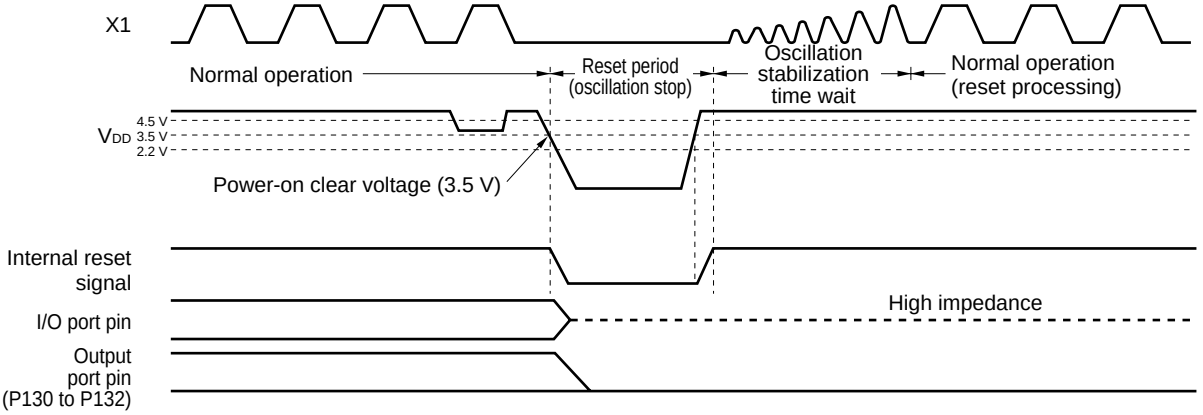


Table 16-1. Hardware Status After Reset (1/2)

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)		Undefined
RAM	Data memory	UndefinedNote 2
	General-purpose register	UndefinedNote 2
Port (output latch)	Ports 0, 1, 3 to 7, 12, 13 (P0, P1, P3 to P7, P12, P13)	00H
Port mode registers (PM0, PM3 to PM7, PM12)		FFH
Pull-up resistor option register 4 (PU4)		00H
Processor clock control register (PCC)		04H
Oscillation stabilization time select register (OSTS)		04H
DTS system clock select register (DTSCK)		00HNote 3
Memory size switching register (IMS)		CFHNote 4
Internal expansion RAM size switching register (IXS)		0CHNote 5
8-bit timer/event counter	Counters 50 to 53 (TM50 to TM53)	00H
	Compare registers 50 to 53 (CR50 to CR53)	Undefined
	Clock select registers 50 to 53 (TCL50 to TCL53)	00H
	Mode control registers 50 to 53 (TMC50 to TMC53)	00H
Watchdog timer	Clock select register (WDCS)	00H
	Mode register (WDTM)	00H
Buzzer output controller	BEEP clock select register 0 (BEEPCL0)	00H
	Clock output select register (CKS)	00H
Serial interface	Shift registers 30 to 32 (SIO30 to SIO32)	Undefined
	Operating mode registers 30 to 32 (CSIM30 to CSIM32)	00H
	Port select register 32 (SIO32SEL)	00H

Notes 1. During reset input or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. The status before reset is retained even after reset in the standby mode.

3. Though the initial value is 00H, be sure to set it to 01H before use.

4. The initial value is CFH. Set the following value to this register depending on the model:
μPD178053: C6H
μPD178054: C8H
μPD178F054: Value corresponding to mask ROM versions

5. Do not assign a value other than 0CH.

Table 16-1. Hardware Status After Reset (2/2)

Hardware		Status After Reset
A/D converter	Mode register 3 (ADM3)	00H
	A/D conversion result register 3 (ADCR3)	Undefined
	Analog input channel specification register 3 (ADS3)	00H
	Power-fail comparison mode register 3 (PFM3)	00H
	Power-fail comparison threshold value register 3 (PFT3)	00H
Interrupt	Request flag registers (IF0L and IF0H)	00H
	Mask flag registers (MK0L and MK0H)	FFH
	Priority specification flag registers (PR0L and PR0H)	FFH
	External interrupt rising edge enable register (EGP)	00H
	External interrupt falling edge enable register (EGN)	00H
PLL frequency synthesizer	PLL mode select register (PLLMD)	00H
	PLL reference mode register (PLLRF)	0FH
	PLL unlock F/F judge register (PLLUL)	Retained**Note 1**
	PLL data registers (PLLRH, PLLRL, and PLLR0)	Undefined
	PLL data transfer register (PLLNS)	00H
Frequency counter	IF counter mode select register (IFCMD)	00H
	IF counter gate judge register (IFCJG)	00H
	IF counter control register (IFCCR)	00H
	IF counter register (IFCR)	0000H
Power-on clear	POC status register (POCS)	Retained**Note 2**

Notes 1. Undefined only at power-on clear reset

2. 03H only at power-on clear reset

16.2 Power Failure Detection Function

If reset is effected by means of power-on clear, bit 0 (POCM) of the POC status register (POCS) is set to 1. If reset is effected by the $\overline{\text{RESET}}$ pin or the watchdog timer, however, POCM holds the previous status.

A power failure status can be detected by detecting this POCM after reset by power-on clear has been cleared (after program execution has been started from address 0000H).

Figure 16-5. Format of POC Status Register (POCS)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POCS	0	0	0	0	0	0	VM45	POCM	FF1BH	Retained[Note]	R&Reset

POCM	Detection of power-on clear occurrence status
0	Power-on clear does not occur
1[Note]	Reset is effected by power-on clear

Note The value of this register is set to 03H only when reset is effected through power-on clearing. It is not reset by the $\overline{\text{RESET}}$ pin or watchdog timer.

Remark The values of the special function registers, other than POCS and PLLUL, at power-on clear are the same as the values following a reset by the $\overline{\text{RESET}}$ pin or watchdog timer (see **Table 16-1**).

16.3 4.5 V Voltage Detection Function

This function is used to detect a voltage drop on the V_{DD} pin below 4.5 V (4.5 V ±0.3 V). If the voltage on the V_{DD} pin drops below 4.5 V (4.5 V ±0.3 V), bit 1 (VM45) of the POC status register (POCS) is set.

Note, however, that this 4.5 V voltage detection function does not cause internal reset.

Figure 16-6. Format of POC Status Register (POCS)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POCS	0	0	0	0	0	0	VM45	POCM	FF1BH	Retained[Note]	R&Reset

VM45	Detection of voltage level of V_{DD} pin
0	Does not detect if V_{DD} pin is less than 4.5 V (4.3 V ±0.3 V)
1	Detects if V_{DD} pin is less than 4.5 V (4.3 V ±0.3 V)

Note The value of this register is set to 03H only at power-on clear reset, and is not reset by the $\overline{\text{RESET}}$ pin and watchdog timer.

Remark The values of the special function registers, other than POCS and PLLUL, at power-on clear are the same as the values following a reset by the $\overline{\text{RESET}}$ pin or watchdog timer (see **Table 16-1**).

CHAPTER 17 μPD178F054

The μPD178F054 is provided with a flash memory to/from which data can be rewritten/erased with the device mounted on the printed circuit board. The differences between the flash memory (μPD178F054) and mask ROM versions (μPD178053 and 178054) are shown in Table 17-1.

Table 17-1. Differences Between μPD178F054 and Mask ROM Versions

Item		μPD178F054	μPD178053, 178054
Internal memory	ROM structure	Flash memory	Mask ROM
	ROM capacity	32 KB	μPD178053: 24 KB μPD178054: 32 KB
Internal ROM capacity selected by memory size switching register (IMS)		Equivalent to mask ROM version	μPD178053: C6H μPD178054: C8H
IC pin		Not provided	Provided
V_{PP} pin		Provided	Not provided
Electrical specifications		Refer to **CHAPTER 19 ELECTRICAL SPECIFICATIONS**.	

17.1 Memory Size Switching Register (IMS)

The internal memory capacity of the μPD178F054 can be changed using the memory size switching register (IMS). By using this register, the memory of the μPD178F054 can be mapped in the same manner as a mask ROM version with a different internal memory capacity.

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

Reset input sets this register to CFH.

Be sure to set IMS to C6H or C8H.

Figure 17-1. Format of Memory Size Switching Register (IMS)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0	FFF0H	CFH	R/W

RAM2	RAM1	RAM0	Selection of internal high-speed RAM capacity
0	1	0	512 bytes
1	1	0	1024 bytes
Other than above			Setting prohibited

ROM3	ROM2	ROM1	ROM0	Selection of internal ROM capacity
0	1	1	0	24 KB
1	0	0	0	32 KB
Other than above				Setting prohibited

Table 17-2 shows the setting of IMS to perform the same memory mapping as that of a mask ROM version.

Table 17-2. Set Value of Memory Size Switching Register

Targeted Model	Set Value of IMS
μPD178053	C6H
μPD178054	C8H

17.2 Internal Expansion RAM Size Switching Register (IXS)

The internal expansion RAM capacity of the μPD178F054 can be changed using the internal expansion RAM size switching register (IXS). By using this register, the memory of the μPD178F054 can be mapped in the same manner as a mask ROM version with a different internal expansion RAM capacity.

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

Reset input sets this register to 0CH.

Caution Do not set a value other than the initial value.

Figure 17-2. Format of Internal Expansion RAM Size Switching Register (IXS)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
IXS	0	0	0	IXRAM4	IXRAM3	IXRAM2	IXRAM1	IXRAM0	FFF4H	0CH	R/W

IXRAM4	IXRAM3	IXRAM2	IXRAM1	IXRAM0	Selection of internal expansion RAM capacity
0	1	1	0	0	0 bytes
Other than above					Setting prohibited

Table 17-3 shows the setting of IXS to perform the same memory mapping as that of a mask ROM version.

Table 17-3. Set Value of Internal Expansion RAM Size Switching Register

Targeted Model	Set Value of IXS
μPD178053, 178054	0CH

17.3 Flash Memory Programming

The program memory provided in the μPD178F054 is flash memory.

The flash memory can be written on-board, i.e., with the μPD178F054 mounted on the target system. To do so, connect a dedicated flash programmer (Flashpro III (Part number: FL-PR3, PG-FP3)) to the host machine and target system.

Remark FL-PR3 and PG-FP3 are products of Naito Densei Machida Mfg. Co., Ltd.

17.3.1 Selecting communication mode

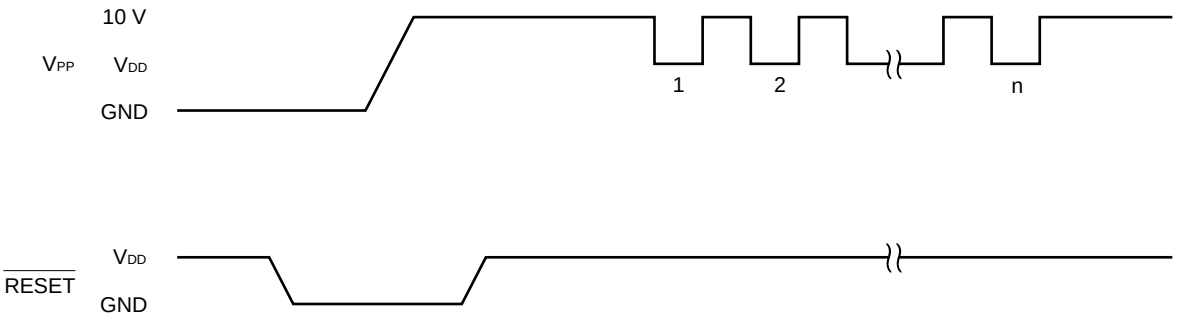
The flash memory is written by using Flashpro III and by means of serial communication. Select a communication mode from those listed in Table 17-4. To select a communication mode, the format shown in Figure 17-3 is used. Each communication mode is selected depending on the number of V_{PP} pulses shown in Table 17-4.

Table 17-4. Communication Modes

Communication Mode	Pins Used	Number of V_{PP} Pulses
3-wire serial I/O (SIO3)	SI30/P70 SO30/P71 $\overline{SCK30}$/P72	0
	SI31/P74 $\overline{SO31}$/P75 SCK31/P76	1
	SI32/P120 $\overline{SO32}$/P121 $\overline{SCK32}$/P122	2

Caution Be sure to select a communication mode by the number of V_{PP} pulses shown in Table 17-4.

Figure 17-3. Format of Communication Mode Selection



17.3.2 Flash memory programming function

An operation such as writing the flash memory is performed when a command or data is transmitted/received in the selected communication mode. The major flash memory programming functions are listed in Table 17-5.

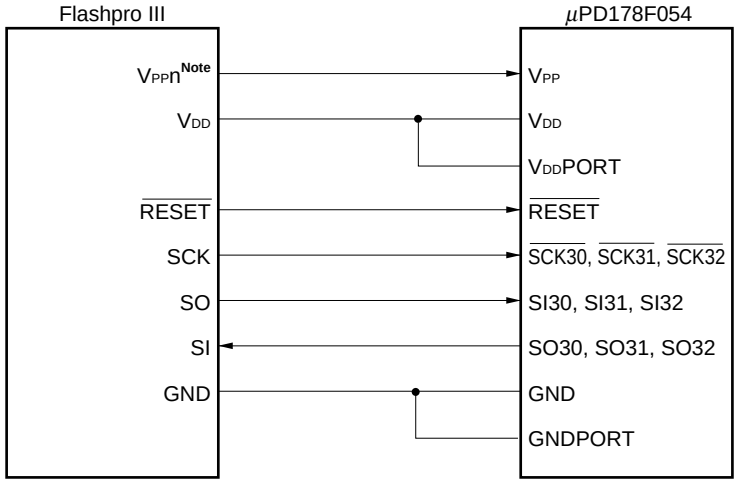
Table 17-5. Major Functions of Flash Memory Programming

Function	Description
Batch erase	Erases all memory contents.
Batch blank check	Checks erased status of entire memory.
Data write	Writes data to flash memory starting from write start address and based on number of data (bytes) to be written).
Batch verify	Compares all contents of memory with input data.

17.3.3 Connecting Flashpro III

The connection between Flashpro III and the μPD178F054 is shown in Figure 17-4.

Figure 17-4. Connection of Flashpro III in 3-Wire Serial I/O Mode



Note n = 1, 2

17.3.4 Setting example for Flashpro III (PG-FP3)

When writing data to flash memory using Flashpro III (PG-FP3), use the following settings.

<1> Load parameter file.
<2> Select the serial mode and serial cock using the type command.
<3> An example of the settings for PG-FP3 is shown in Table 17-6.

Table 17-6. Setting Example for Flashpro III (PG-FP3)

Communication Mode	Setting of Flashpro III		Number of V_{PP} Pulses[Note]
3-wire serial I/O (SIO3)	COMM PORT	SIO ch-0	0
	CPU CLK	On Target Board	
		In Flashpro	
	On Target Board	4.1943 MHz	
	SIO CLK	1.0 MHz	
	In Flashpro	4.0 MHz	
	SIO CLK	1.0 MHz	
3-wire serial I/O (SIO31)	COMM PORT	SIO-ch1	1
	CPU CLK	On Target Board	
		In Flashpro	
	On Target Board	4.1943 MHz	
	SIO CLK	1.0 MHz	
	In Flashpro	4.0 MHz	
	SIO CLK	1.0 MHz	
3-wire serial I/O (SIO32)	COMM PORT	SIO-ch2	2
	CPU CLK	On Target Board	
		In Flashpro	
	On Target Board	4.1943 MHz	
	SIO CLK	1.0 MHz	
	In Flashpro	4.0 MHz	
	SIO CLK	1.0 MHz	

Note Number of V_{PP} pulse supplied by Flashpro III (PG-FP3) when serial mode is initialized. This determines the pin used for the communication.

Remark COMM PORT: Selection of serial port
SIO CLK: Selection of serial clock frequency
CPU CLK: Selection of CPU clock source to be input

CHAPTER 18 INSTRUCTION SET

This chapter describes each instruction set of the μPD178054 Subseries as list table. For details of its operation and operation code, refer to the **78K/0 Series User's Manual Instruction (U12326E)**.

18.1 Conventions

18.1.1 Operand symbols and description

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

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

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

For operand register symbols, 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.

Table 18-1. Operand Symbols and Descriptions

Symbol	Description
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7),
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special-function register symbol[Note]
sfrp	Special-function register symbol (16-bit manipulatable register even addresses only)[Note]
saddr	FE20H to FF1FH Immediate data or labels
saddrp	FE20H to FF1FH Immediate data or labels (even address only)
addr16	0000H to FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions)
addr11	0800H to 0FFFH Immediate data or labels
addr5	0040H to 007FH Immediate data or labels (even address only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special function register symbols, refer to **Table 3-4 Special Function Registers**.

18.1.2 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)
——:	Inverted data
addr16:	16-bit immediate data or label
jdisp8:	Signed 8-bit data (displacement value)

18.1.3 Description of “flag operation” column

(Blank):	Nt affected
0:	Cleared to 0
1:	Set to 1
$\times$:	Set/cleared according to the result
R:	Previously saved value is restored

18.2 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”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

2. This clock cycle applies to internal ROM program.

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”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

2. This clock cycle applies to internal ROM program.

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”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

2. This clock cycle applies to internal ROM program.

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”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

2. This clock cycle applies to internal ROM program.

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 manipulate	MOV1	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7	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

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

2. This clock cycle applies to internal ROM program.

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

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

2. This clock cycle applies to internal ROM program.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Call/return	CALL	!addr16	3	7	–	$(SP-1) \leftarrow (PC+3)_H$, $(SP-2) \leftarrow (PC+3)_L$, $PC \leftarrow$ addr16, $SP \leftarrow SP-2$	
	CALLF	!addr11	2	5	–	$(SP-1) \leftarrow (PC+2)_H$, $(SP-2) \leftarrow (PC+2)_L$, $PC_{15-11} \leftarrow 00001$, $PC_{10-0} \leftarrow$ addr11, $SP \leftarrow SP-2$	
	CALLT	[addr5]	1	6	–	$(SP-1) \leftarrow (PC+1)_H$, $(SP-2) \leftarrow (PC+1)_L$, $PC_H \leftarrow$ (00000000, addr5 + 1), $PC_L \leftarrow$ (00000000, addr5), $SP \leftarrow SP-2$	
	BRK		1	6	–	$(SP-1) \leftarrow PSW$, $(SP-2) \leftarrow (PC+1)_H$, $(SP-3) \leftarrow (PC+1)_L$, $PC_H \leftarrow$ (003FH), $PC_L \leftarrow$ (003EH), $SP \leftarrow SP-3$, $IE \leftarrow 0$	
	RET		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $SP \leftarrow SP+2$	
	RETI		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP+2)$, $SP \leftarrow SP+3$, $NMIS \leftarrow 0$	R R R
	RETB		1	6	–	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP+2)$, $SP \leftarrow SP+3$	R R R
Stack manipulate	PUSH	PSW	1	2	–	$(SP-1) \leftarrow PSW$, $SP \leftarrow SP-1$	
		rp	1	4	–	$(SP-1) \leftarrow rp_H$, $(SP-2) \leftarrow rp_L$, $SP \leftarrow SP-2$	
	POP	PSW	1	2	–	$PSW \leftarrow (SP)$, $SP \leftarrow SP+1$	R R R
		rp	1	4	–	$rp_H \leftarrow (SP+1)$, $rp_L \leftarrow (SP)$, $SP \leftarrow SP+2$	
	MOVW	SP, #word	4	–	10	$SP \leftarrow$ word	
		SP, AX	2	–	8	$SP \leftarrow AX$	
		AX, SP	2	–	8	$AX \leftarrow SP$	
Unconditional branch	BR	!addr16	3	6	–	$PC \leftarrow$ addr16	
		$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8	
		AX	2	8	–	$PC_H \leftarrow A$, $PC_L \leftarrow X$	
Conditional branch	BC	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if CY = 1	
	BNC	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if CY = 0	
	BZ	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if Z = 1	
	BNZ	$addr16	2	6	–	$PC \leftarrow PC+2+$ jdisp8 if Z = 0	

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

2. This clock cycle applies to internal ROM program.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Conditional branch	BT	saddr.bit, $addr16	3	8	9	PC ← PC + 3 + jdisp8 if(saddr.bit) = 1	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 1	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1	
		PSW.bit, $addr16	3	–	9	PC ← PC + 3 + jdisp8 if PSW.bit = 1	
		[HL].bit, $addr16	3	10	11	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

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

2. This clock cycle applies to internal ROM program.

18.3 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

1st Operand \ 2nd 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

2nd Operand / 1st Operand	#word	AX	rp[Note]	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW[Note]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, HL

(3) Bit manipulation instructions

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

2nd Operand / 1st Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call/instructions/branch instructions

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

2nd Operand / 1st 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

Absolute Maximum Ratings (T_A = 25°C)

Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}			–0.3 to +6.0	V
	V_{DD}PORT			–0.3 to V_{DD} + 0.3 Note 1	V
	V_{DD}PLL			–0.3 to V_{DD} + 0.3 Note 1	V
	V_{PP}	μPD178F054 only		–0.3 to +10.5	V
Input voltage	V_I			–0.3 to V_{DD} + 0.3	V
Output voltage	V_O	Excluding P130 to P132		–0.3 to V_{DD} + 0.3	V
Output withstand voltage	V_{BDS}	P130 to P132	N-ch open drain	16	V
Analog input voltage	V_{AN}	P10 to P15	Analog input pin	–0.3 to V_{DD} + 0.3	V
Output current, high	I_{OH}	Per pin		–8	mA
		Total of P00 to P06, P30 to P37, P54 to P57, P60 to P67, and P120 to P125		–15	mA
		Total of P40 to P47, P50 to P53, and P70 to P77		–15	mA
Output current, low	I_{OL} Note 2	Per pin	Peak value	16	mA
			rms value	8	mA
		Total of P00 to P06, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P120 to P125, and P130 to P132	Peak value	30	mA
			rms value	15	mA
Operating temperature	T_A	In normal operation mode		–40 to +85	°C
		During flash memory programming (μPD178F054 only)		10 to 40	
Storage temperature	T_{stg}			–55 to +125	°C

Notes 1. Keep the voltage at V_{DD}PORT and V_{DD}PLL same as that at the V_{DD} pin.

2. The rms value should be calculated as follows: [rms value] = [Peak value] x $\sqrt{\text{Duty}}$

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Recommended Supply Voltage Ranges (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Supply voltage	V_{DD1}	When CPU and PLL are operating	4.5	5.0	5.5	V
	V_{DD2}	When CPU is operating and PLL is stopped	3.5	5.0	5.5	V
Data retention voltage	V_{DDR}	When crystal oscillation stops	2.3		5.5	V
Output withstand voltage	V_{BDS}	P130 to P132 (N-ch open drain)			15	V

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 3.5 to 5.5 V) (1/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P10 to P15, P30 to P32, P35 to P37, P40 to P47, P50 to P57, P60 to P67, P71, P73, P75, P121, P124		0.7 V_{DD}		V_{DD}	V
	V_{IH2}	P00 to P06, P33, P34, P70, P72, P74, P76, P77, P120, P122, P123, P125, $\overline{\text{RESET}}$		0.8 V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P10 to P15, P30 to P32, P35 to P37, P40 to P47, P50 to P57, P60 to P67, P71, P73, P75, P121, P124		0		0.3 V_{DD}	V
	V_{IL2}	P00 to P06, P33, P34, P70, P72, P74, P76, P77, P120, P122, P123, P125, $\overline{\text{RESET}}$		0		0.2 V_{DD}	V
Output voltage, high	V_{OH1}	P00 to P06, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P120 to P125	4.5 V ≤ V_{DD} ≤ 5.5 V, I_{OH} = –1 mA	V_{DD} – 1.0			V
			3.5 V ≤ V_{DD} < 4.5 V, I_{OH} = –100 μA	V_{DD} – 0.5			V
	V_{OH2}	EO0, EO1	4.5 V ≤ V_{DD} ≤ 5.5 V, I_{OH} = –3 mA	V_{DD} – 1.0			V
Output voltage, low	V_{OL1}	P00 to P06, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P120 to P125, P130 to P132	4.5 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 1 mA			1.0	V
			3.5 V ≤ V_{DD} < 4.5 V, I_{OL} = 100 μA			0.5	V
	V_{OL2}	EO0, EO1	V_{DD} = 4.5 to 5.5 V, I_{OL} = 3 mA			1.0	V
Input leakage current, high	I_{LIH1}	P00 to P06, P10 to P15, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P120 to P125, $\overline{\text{RESET}}$	V_{IN} = V_{DD}			3	μA

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 3.5 to 5.5 V) (2/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input leakage current, low	I_{LIL1}	P00 to P06, P10 to P15, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P120 to P125, $\overline{\text{RESET}}$	V_{IN} = 0 V			–3	μA
Output off leakage current	I_{LOH1}	P130 to P132	V_{OUT} = 15 V			–3	μA
	I_{LOL1}	P130 to P132	V_{OUT} = 0 V			3	μA
	I_{LOH2}	EO0, EO1	V_{OUT} = V_{DD}			–3	μA
	I_{LOL2}	EO0, EO1	V_{OUT} = 0 V			3	μA
Supply current[Note]	I_{DD1}	When CPU is operating and PLL is stopped. Sine wave input to X1 pin At f_X = 4.5 MHz V_{IN} = V_{DD}	μPD178053, μPD178054		2.5	15	mA
			μPD178F054		5.0	18	mA
	I_{DD2}	In HALT mode with PLL stopped. Sine wave input to X1 pin At f_X = 4.5 MHz V_{IN} = V_{DD}	μPD178053, μPD178054		0.2	0.8	mA
			μPD178F054		0.3	0.8	mA
Data retention voltage	V_{DDR1}	When crystal resonator is oscillating		3.5		5.5	V
	V_{DDR2}	When crystal oscillation is stopped	Power-failure detection function	2.2			V
	V_{DDR3}		Data memory retained	2.0			V
Data retention current	I_{DDR1}	When crystal oscillation is stopped	T_A = 25°C, V_{DD} = 5 V		2.0	4.0	μA
	I_{DDR2}		T_A = –40 to +85°C, V_{DD} = 3.5 to 5.5 V		2.0	20	μA

Note Excluding AV_{DD} current and V_{DD}PLL current.

Remarks 1. f_X: System clock oscillation frequency

2. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Reference Characteristics (T_A = –40 to +85°C, V_{DD} = 4.5 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Supply current	I_{DD3}	When CPU and PLL are operating. Sine wave input to VCOH pin At f_{IN} = 160 MHz V_{IN} = 0.15 V_{P-P}		5		mA

AC Characteristics

(1) Basic operation (T_A = –40 to +85°C, V_{DD} = 3.5 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Cycle time (minimum instruction execution time)	T_{CY}	f_X = 4.5 MHz	0.44		7.11	μs
TI50, TI51 input frequency	f_{TI5}				2	MHz
TI50, TI51 input high-/low-level widths	t_{TIH5} t_{TIL5}		200			ns
Interrupt input high-/low-level widths	t_{INTH} t_{INTL}	INTP0 to INTP4	1			μs
$\overline{\text{RESET}}$ pin low-level width	t_{RSL}		10			μs

(2) Serial interface SIO3 (T_A = –40 to +85°C, V_{DD} = 3.5 to 5.5 V)

(a) 3-wire serial I/O mode ($\overline{SCK3}$ ... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK3}$ cycle time	t_{KCY1}		800			ns
$\overline{SCK3}$ high/low-level width	t_{KH1}, t_{KL1}		$t_{KCY1}/2 - 50$			ns
SI3 setup time to $\overline{SCK3}\uparrow$	t_{SIK1}		100			ns
SI3 hold time from $\overline{SCK3}\uparrow$	t_{KSI1}		400			ns
Output delay time from $\overline{SCK3}\downarrow$ to SO3	t_{KSO1}	C = 100 pF **Note**			300	ns

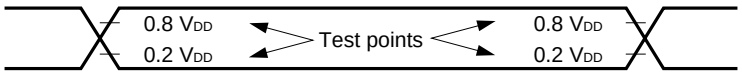
Note C is the load capacitance of $\overline{SCK3}$ and SO3 output line.

(b) 3-wire serial I/O mode ($\overline{SCK3}$ ... external clock input)

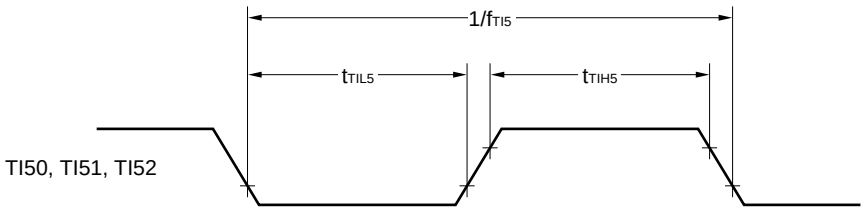
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK3}$ cycle time	t_{KCY2}		800			ns
$\overline{SCK3}$ high/low-level width	t_{KH2}, t_{KL2}		400			ns
SI3 setup time to $\overline{SCK3}\uparrow$	t_{SIK2}		100			ns
SI3 hold time from $\overline{SCK3}\uparrow$	t_{KSI2}		400			ns
Output delay time from $\overline{SCK3}\downarrow$ to SO3	t_{KSO2}	C = 100 pF **Note**			300	ns
$\overline{SCK3}$ at rising or falling edge time	t_{R2}, t_{F2}				1000	ns

Note C is the load capacitance of SO3 output line.

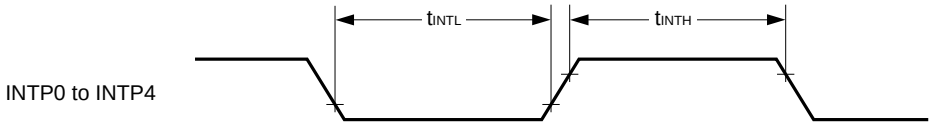
AC Timing Test Point (Excluding X1 Input)



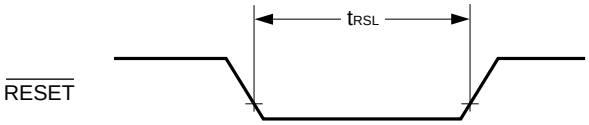
TI Timing



Interrupt Input Timing

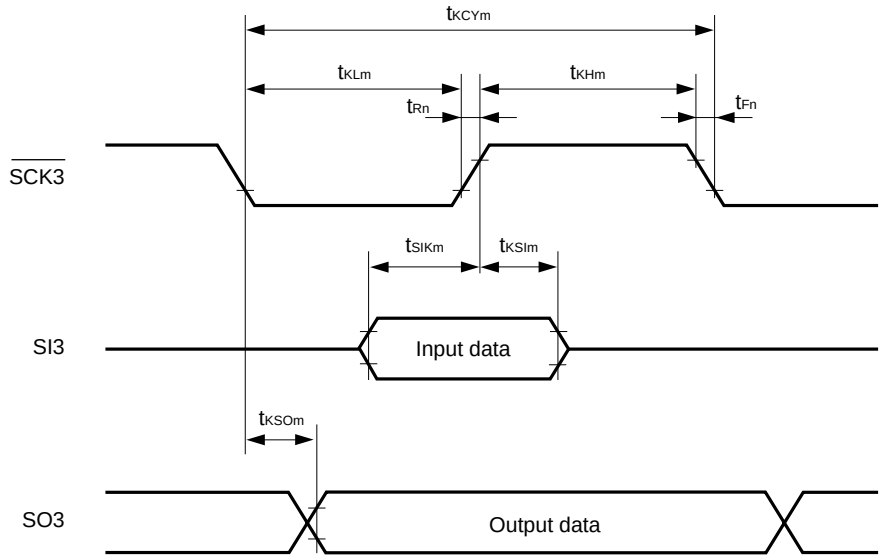


$\overline{\text{RESET}}$ Input Timing



Serial Transfer Timing

3-wire serial I/O mode:



Remark m = 1, 2
n = 2

A/D Converter Characteristics (T_A = –40 to +85°C, V_{DD} = 3.5 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			8	8	8	bit
Total conversion error[Notes 1, 2]		V_{DD} = 4.5 to 5.5 V			±1.0	%FSR
					±1.4	%FSR
Conversion time	t_{CONV}		21.3		64.0	μs
Analog input voltage	V_{IAN}		0		V_{DD}	V

Notes 1. Excluding quantization error (±0.2%FSR)

2. This value is indicated as a ratio to the full-scale value (%FSR).

PLL Characteristics (T_A = –40 to +85°C, V_{DD} = 4.5 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operating frequency	f_{IN1}	VCOL pin, MF mode, sine wave input, V_{IN} = 0.15 V_{P-P}	0.5		3.0	MHz
	f_{IN2}	VCOL pin, HF mode, sine wave input, V_{IN} = 0.15 V_{P-P}	10		40	MHz
	f_{IN3}	VCOH pin, VHF mode, sine wave input, V_{IN} = 0.15 V_{P-P}	60		130	MHz
	f_{IN4}	VCOH pin, VHF mode, sine wave input, V_{IN} = 0.3 V_{P-P}	40		160	MHz

IFC Characteristics (T_A = –40 to +85°C, V_{DD} = 4.5 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operating frequency	f_{IN5}	AMIFC pin, AMIF count mode, sine wave input, V_{IN} = 0.15 V_{P-P}	0.4		0.5	MHz
	f_{IN6}	FMIFC pin, FMIF count mode, sine wave input, V_{IN} = 0.15 V_{P-P}	10		11	MHz
	f_{IN7}	FMIFC pin, AMIF count mode, sine wave input, V_{IN} = 0.15 V_{P-P}	0.4		0.5	MHz

Flash Memory Programming Characteristics (V_{DD} = 3.5 to 5.5 V, T_A = 10 to 40°C) (μPD178F054 only)

(1) Write/delete characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Write current (V_{DD} pin)[Note]	I_{DDW}	When $V_{PP} = V_{PP1}$, f_X = 4.5 MHz			20	mA
Write current (V_{PP} pin)[Note]	I_{PPW}	When $V_{PP} = V_{PP1}$, f_X = 4.5 MHz			20	mA
Delete current (V_{DD} pin)[Note]	I_{DDE}	When $V_{PP} = V_{PP1}$, f_X = 4.5 MHz			20	mA
Delete current (V_{PP} pin)[Note]	I_{PPE}	When $V_{PP} = V_{PP1}$			100	mA
Unit delete time	t_{ER}		0.5	1	1	s
Total delete time	t_{ERA}				20	s
Number of overwrite		Delete and write are counted as one cycle			20	times
V_{PP} power supply voltage	V_{PP0}	In normal mode	0		0.2 V_{DD}	V
	V_{PP1}	During flash memory programming	9.7	10.0	10.3	V

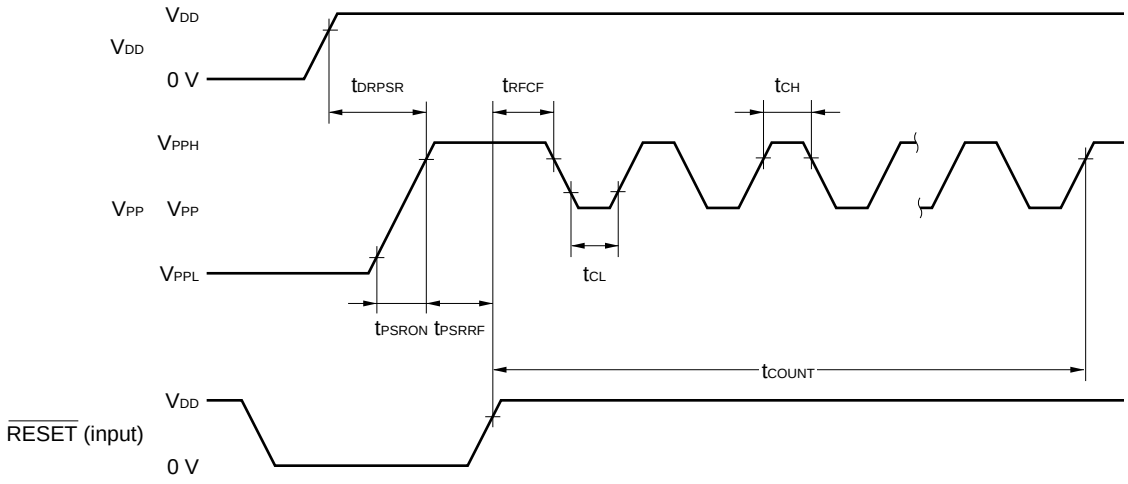
Note Port current (including current flowing to internal pull-up resistors) is not included.

Remark f_X: System clock oscillation frequency

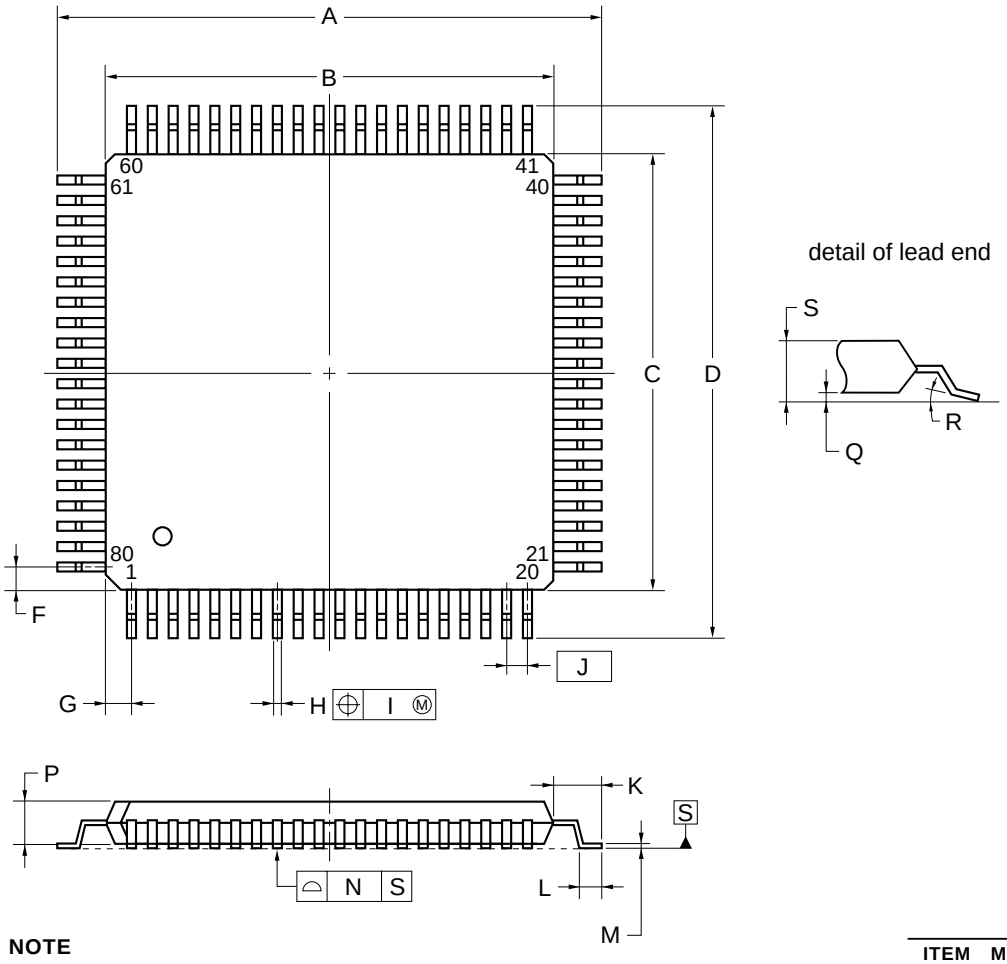
(2) Serial write operation characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{PP} setup time	t_{PSRON}	V_{PP} high voltage	1.0			μs
$V_{PP}\uparrow$ setup time from $V_{DD}\uparrow$	t_{DRPSR}	V_{PP} high voltage	1.0			μs
$\overline{\text{RESET}}\uparrow$ setup time from $V_{PP}\uparrow$	t_{PSRRF}	V_{PP} high voltage	1.0			μs
V_{PP} count start time from $\overline{\text{RESET}}\uparrow$	t_{RFCF}		1.0			μs
Count execution time	t_{COUNT}				2.0	ms
V_{PP} counter high-level width	t_{CH}		8.0			μs
V_{PP} counter low-level width	t_{CL}		8.0			μs
V_{PP} counter noise elimination width	t_{NFW}			40		ns

Flash Write Mode Setting Timing



80-PIN PLASTIC QFP (14x14)



NOTE

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	17.20±0.20
B	14.00±0.20
C	14.00±0.20
D	17.20±0.20
F	0.825
G	0.825
H	0.32±0.06
I	0.13
J	0.65 (T.P.)
K	1.60±0.20
L	0.80±0.20
M	$0.17^{+0.03}_{-0.07}$
N	0.10
P	1.40±0.10
Q	0.125±0.075
R	$3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$
S	1.70 MAX.

P80GC-65-8BT-1

★ CHAPTER 21 RECOMMENDED SOLDERING CONDITIONS

The μPD178053, 178054, and 178F054 should be soldered and mounted under the following recommended conditions.

For details of the recommended soldering conditions, refer to the document **Semiconductor Device Mounting Technology Manual (C10535E)**.

For soldering methods and conditions other than those recommended below, contact an NEC sales representative.

Table 21-1. Surface Mounting Type Soldering Conditions

μPD178053GC-×××-8BT: 80-pin plastic QFP (14 × 14)
μPD178054GC-×××-8BT: 80-pin plastic QFP (14 × 14)
μPD178F054GC-8BT: 80-pin plastic QFP (14 × 14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Twice or less	IR35-00-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Twice or less	VP15-00-2
Wave soldering	Soldering bath temperature: 260°C or less, Time: 10 seconds max., Count: Once, Preheating temperature: 120°C max. (package surface temperature)	WS60-00-1
Partial heating	Pin temperature: 300°C or less, Time: 3 seconds max. (per pin row)	–

Caution Do not use different soldering methods together (except for partial heating).

★

APPENDIX A DEVELOPMENT TOOLS

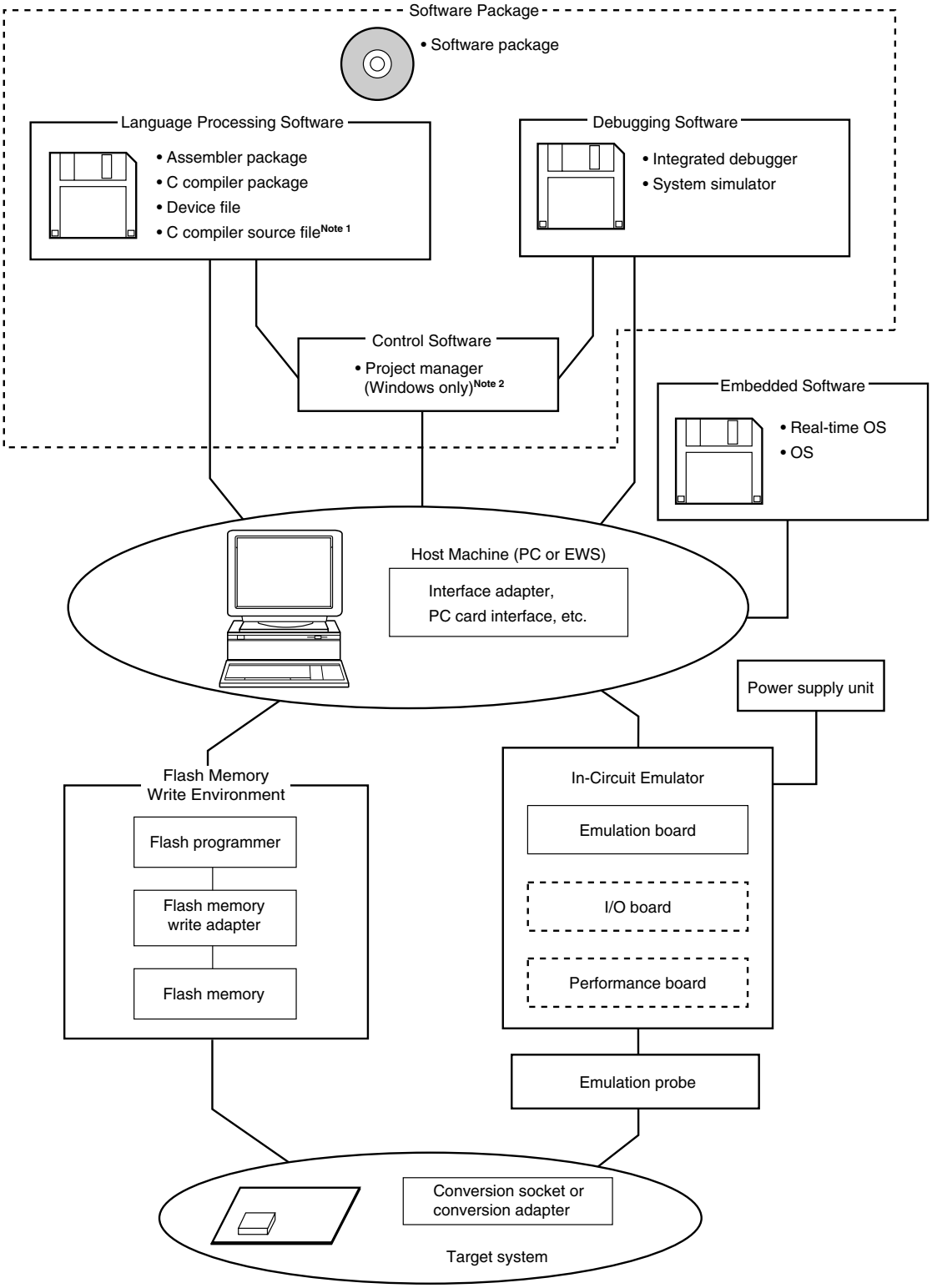
The following development tools are available for the development of systems which employ the μPD178054 Subseries. Figure A-1 shows the configuration example of the tools.

- **Support for PC98-NX series**
  Unless otherwise specified, products supported by IBM PC/AT™ compatibles can be used for PC98-NX series computers. When using PC98-NX series computers, refer to the description for IBM PC/AT compatibles.

- **Windows**
  Unless otherwise specified, "Windows" means the following OSs.
  - Windows 3.1
  - Windows 95
  - Windows 98
  - Windows 2000
  - Windows NT™ Ver. 4.0

Figure A-1. Configuration of Development Tools (1/2)

(1) When using the in-circuit emulator IE-78K0-NS

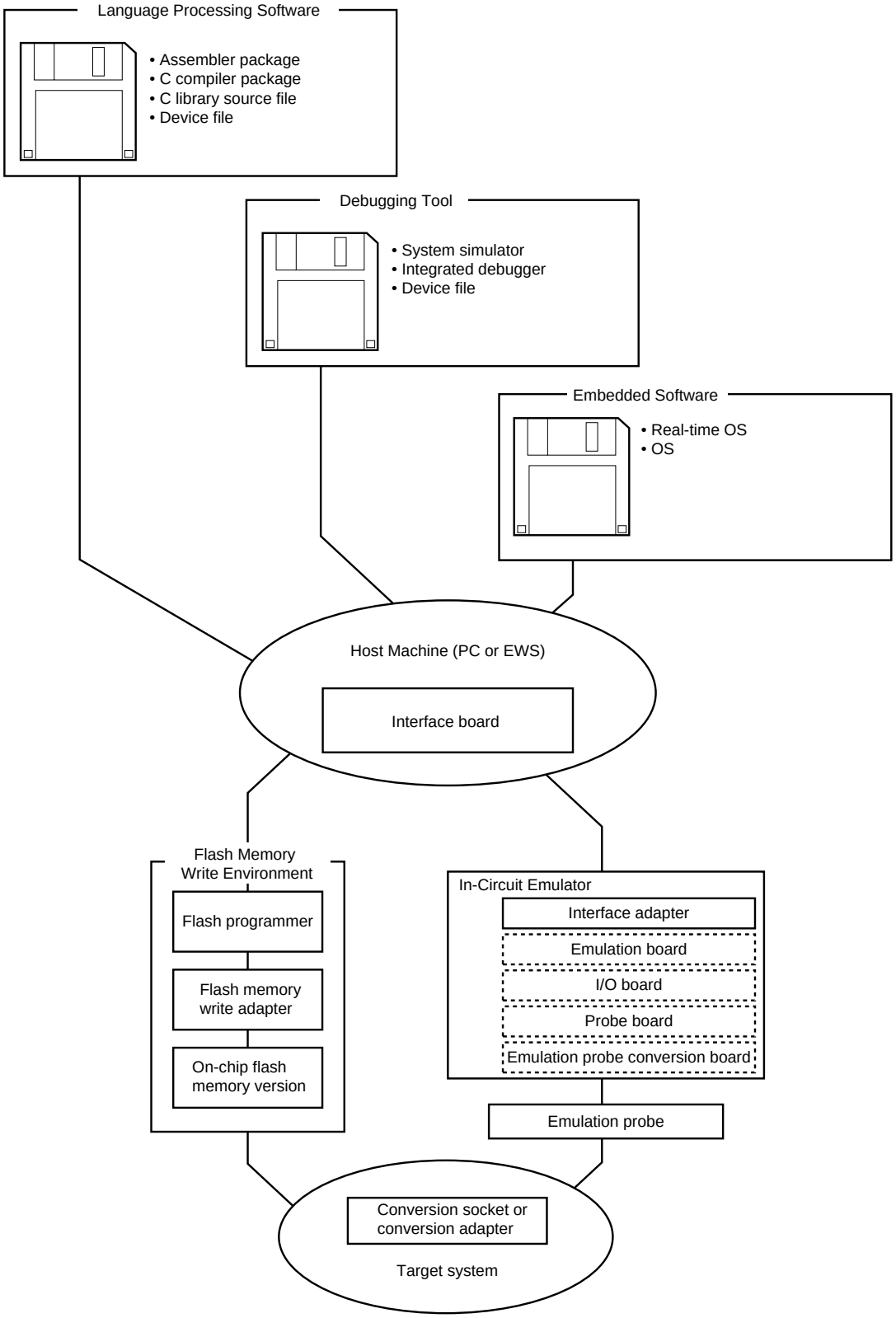


Notes 1. The C compiler source file is not included in the software package.

2. The project manager is included in the assembler package.
The project manager is only used for Windows.

Figure A-1. Configuration of Development Tools (2/2)

(2) When using the in-circuit emulator IE-78001-R-A



Remark Items in broken line boxes differ according to the development environment. Refer to **A.5 Debugging Tools (Hardware)**.

A.1 Software Package

SP78K0 Software package	This package contains various software tools for 78K/0 Series development. The following tools are included. RA78K0, CC78K0, ID78K0-NS, SM78K0, and various device files
	Part Number: μS××××SP78K0

Remark ×××× in the part number differs depending on the OS used.

μS××××SP78K0

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series,	Windows (Japanese version)	CD-ROM
BB17	IBM PC/AT compatibles	Windows (English version)	

A.2 Language Processing Software

RA78K0 Assembler package	This assembler converts programs written in mnemonics into an object codes executable with a microcontroller. Further, this assembler is provided with functions capable of automatically creating symbol tables and branch instruction optimization. This assembler should be used in combination with an optical device file (DF178054). **<Precaution when using RA78K0 in PC environment>** This assembler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part Number: μS××××RA78K0
CC78K0 C compiler package	This compiler converts programs written in C language into object codes executable with a microcontroller. This compiler should be used in combination with an optical assembler package and device file. **<Precaution when using CC78K0 in PC environment>** This C compiler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part Number: μS××××CC78K0
DF178054**Note 1** Device file	This file contains information peculiar to the device. This device file should be used in combination with an optical tool (RA78K0, CC78K0, SM78K0, ID78K0-NS, and ID78K0). Corresponding OS and host machine differ depending on the tool used.
	Part Number: μS××××DF178054
CC78K0-L**Note 2** C library source file	This is a source file of functions configuring the object library included in the C compiler package. This file is required to match the object library included in C compiler package to the user's specifications. It does not depend on the operating environment because it is a source file.
	Part Number: μS××××CC78K0-L

Notes 1. The DF178054 can be used in common with the RA78K0, CC78K0, SM78K0, ID78K0-NS, and ID78K0.

2. CC78K0-L is not included in the software package (SP78K0).

Remark ××××in the part number differs depending on the host machine and OS used.

μS××××RA78K0
μS××××CC78K0

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	
AB17		Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	
3P17	HP9000 series 700™	HP-UX™ (Rel. 10.10)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.1), Solaris™ (Rel. 2.5.1)	

μS××××DF178054
μS××××CC78K0-L

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT
3K13	SPARCstation	SunOS (Rel. 4.1.1),	3.5-inch 2HD FD
3K15		Solaris (Rel. 2.5.1)	1/4-inch CGMT

A.3 Control Software

Project manager	This is control software designed to enable efficient user program development in the Windows environment. All operations used in development of a user program, such as starting the editor, building, and starting the debugger, can be performed from the project manager. **<Caution>** The project manager is included in the assembler package (RA78K0). It can only be used in Windows.

A.4 Flash Memory Writing Tools

Flashpro III (Part number: FL-PR3, PG-FP3) Flash programmer	Flash programmer dedicated to microcontrollers with on-chip flash memory.
FA-80GC Flash memory writing adapter	Flash memory writing adapter used connected to the Flashpro III. • FA-80GC: 80-pin plastic QFP (GC-8BT type)
Flashpro III controller	Control program that runs on a PC. This is supplied with Flashpro III.

Remark Flashpro III and FA-80GC are products of Naito Densei Machida Mfg. Co., Ltd.
Contact: +81-45-475-4191 Naito Densei Machida Mfg. Co., Ltd.

A.5 Debugging Tools (Hardware) (1/2)

(1) When using the in-circuit emulator IE-78K0-NS

IE-78K0-NS In-circuit emulator		The in-circuit emulator serves to debug hardware and software when developing application systems using a 78K/0 Series product. It corresponds to integrated debugger (ID78K0-NS). This emulator should be used in combination with power supply unit, emulation probe, and interface adapter which is required to connect this emulator to the host machine.
IE-78K0-NS-PA Performance board		This board is used for extending the IE-78K0-NS functions, and is used connected to the IE-78K0-NS. With the addition of this board, the addition of a coverage function, enhancement of tracer and timer functions, and other such debugging function enhancements are possible.
IE-78K0-NS-A In-circuit emulator		In-circuit emulator that combines IE-78K0-NS and IE-78K0-NS-PA
IE-70000-MC-PS-B Power supply unit		This adapter is used for supplying power from a receptacle of 100 to 240 V AC.
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 (C bus compatible).
IE-70000-CD-IF-A PC card interface		This is PC card and interface cable required when using the notebook-type computer as the IE-78K0-NS host machine (PCMCIA socket compatible).
IE-70000-PC-IF-C Interface adapter		This adapter is required when using the IBM PC compatible computers as the IE-78K0-NS host machine (ISA bus compatible).
IE-70000-PCI-IF-A Interface adapter		This adapter is required when using a PC with a PCI bus as the IE-78K0-NS host machine.
IE-178054-NS-EM1 Emulation board		This board emulates the operations of the peripheral hardware peculiar to a device. It should be used in combination with an in-circuit emulator.
NP-80GC Emulation probe		This probe is used to connect the in-circuit emulator to the target system and is designed for 80-pin plastic QFP (GC-8BT type).
	EV-9200GC-80 Conversion socket (Refer to **Figures A-2, A-3**)	This conversion socket connects the NP-80GC to the target system board designed to mount an 80-pin plastic QFP (GC-8BT type).

Remarks 1. NP-80GC is a product of Naito Densei Machida Mfg. Co., Ltd.
Contact: +81-45-475-4191 Naito Densei Machida Mfg. Co., Ltd.

2. EV-9200GC-80 is sold in five-unit sets.

A.5 Debugging Tools (Hardware) (2/2)

(2) When using the in-circuit emulator IE-78001-R-A

IE-78001-R-A In-circuit emulator		The in-circuit emulator serves to debug hardware and software when developing application systems using a 78K/0 Series product. It corresponds to integrated debugger (ID78K0). This emulator should be used in combination with emulation probe and interface adapter, which is required to connect this emulator to the host machine.
IE-70000-98-IF-C Interface adapter		This adapter is required when using the PC-9800 series computer (except notebook type) as the IE-78001-R-A host machine (C bus compatible).
IE-70000-PC-IF-C Interface adapter		This adapter is required when using the IBM PC/AT compatible computers as the IE-78001-R-A host machine (ISA bus compatible).
IE-70000-PCI-IF-A Interface adapter		This adapter is required when using a PC with a PCI bus as the IE-78001-R-A host machine.
IE-78000-R-SV3 Interface adapter		This is adapter and cable required when using an EWS computer as the IE-78001-R-A host machine, and is used connected to the board in the IE-78000-R-A. As Ethernet™, 10Base-5 is supported. With the other method, a commercially available conversion adapter is necessary.
IE-178054-NS-EM1 Emulation board		This board emulates the operations of the peripheral hardware peculiar to a device. It should be used in combination with an in-circuit emulator and emulation conversion board.
	IE-78K0-R-EX1 Emulation probe conversion board	This board is required when using the IE-178054-NS-EM1 on the IE-78001-R-A.
EP-78230GC-R Emulation probe		This probe is used to connect the in-circuit emulator to the target system and is designed for 80-pin plastic QFP (GC-8BT type).
	EV-9200GC-80 Conversion socket (Refer to **Figures A-2, A-3**)	This conversion socket connects the EP-78230GC-R to the target system board designed to mount an 80-pin plastic QFP (GC-8BT type).

Remark EV-9200GC-80 is sold in five-unit sets.

A.6 Debugging Tools (Software)

SM78K0 System simulator	This is a system simulator for the 78K/0S Series. The SM78K0 is Windows-based software. It is used to perform debugging at the C source level or assembler level while simulating the operation of the target system on a host machine. Use of the SM78K0 allows the execution of application logical testing and performance testing on an independent basis from hardware development, thereby providing higher development efficiency and software quality. The SM78K0 should be used in combination with the device file (DF178054) (sold separately).
	Part Number: μS××××SM78K0
ID78K0-NS Integrated debugger (supporting in-circuit emulators IE-78K0-NS and IE-78K0-NS-A) ID78K0 Integrated debugger (supporting in-circuit emulator IE-78001-R-A)	This debugger supports the in-circuit emulators for the 78K/0 Series. The ID78K0-NS and ID78K0 are Windows-based software. ID78K0: Supports in-circuit emulator IE-78001-R-A. ID78K0-NS: Supports in-circuit emulators IE-78K0-NS and IE-78K0-NS-A. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. It should be used in combination with the device file (sold separately).
	Part Number: μS××××ID78K0-NS, μS××××ID78K0

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SM78K0
μS××××ID78K0-NS
μS××××ID78K0

××××	Host Machine	OS	Supply Medium
AB13	IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	
AB17		Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

A.7 Embedded Software

RX78K0 Real-time OS	RX78K0 is a real-time OS conforming to the μITRON specifications. Tool (configurator) for generating nucleus of RX78K0 and plural information tables is supplied. Used in combination with an optional assembler package (RA78K0) and device file (DF178054). **<Precaution when using RX78K0 in PC environment>** The real-time OS is a DOS-based application. It should be used in the DOS Prompt when using in Windows.
	Part number: μS××××RX78013-ΔΔΔΔ

Caution When purchasing the RX78K0, fill in the purchase application form in advance and sign the user agreement.

Remark ×××× and ΔΔΔΔ in the part number differ depending on the host machine and OS used.

μS××××RX78013-ΔΔΔΔ

ΔΔΔΔ	Product Outline	Maximum Number for Use in Mass Production
001	Evaluation object	Do not use for mass-produced product.
100K	Mass-production object	0.1 million units
001M		1 million units
010M		10 million units
S01	Source program	Source program for mass-produced object

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	Windows (Japanese version)	3.5-inch 2HD FD
AB13	IBM PC/AT compatibles	Windows (Japanese version)	
BB13		Windows (English version)	

A.8 System Upgrade from Former In-circuit Emulator for 78K/0 Series to IE-78001-R-A

If you already have a former in-circuit emulator for 78K/0 Series microcontrollers (IE-78000-R or IE-78000-R-A), that in-circuit emulator can operate as an equivalent to the IE-78001-R-A by replacing its internal break board with the IE-78001-R-BK.

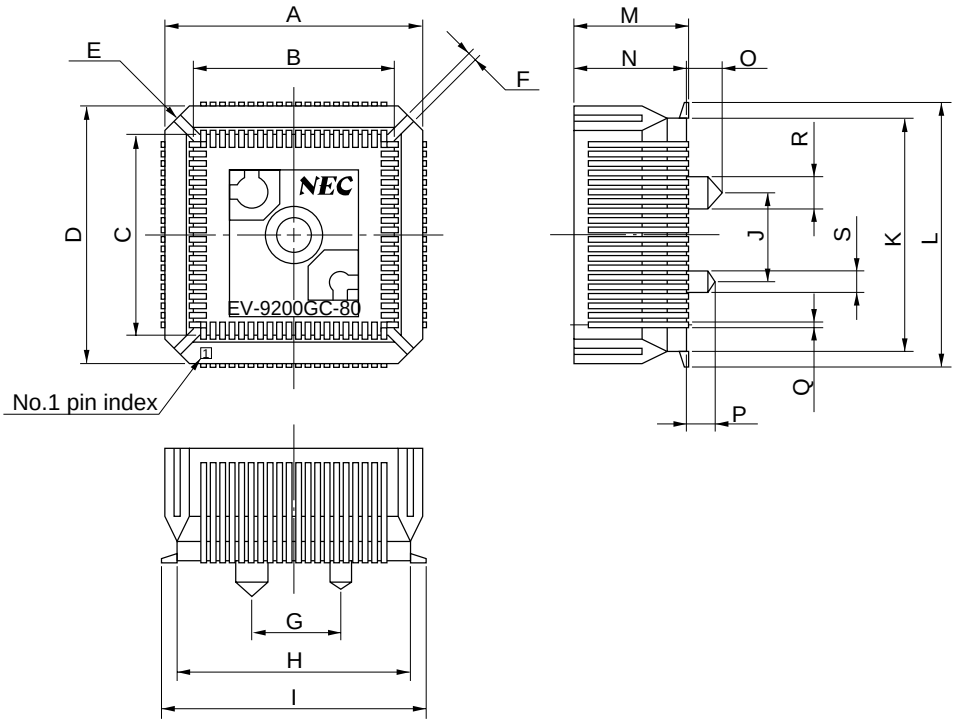
Table A-1. System Upgrade Method from Former In-circuit Emulator for 78K/0 Series to IE-78001-R-A

In-circuit Emulator Owned	In-circuit Emulator Cabinet System Upgrade[Note]	Board to Be Purchased
IE-78000-R	Required	IE-78001-R-BK
IE-78000-R-A	Not required	

Note For upgrading a cabinet, send user's in-circuit emulator to NEC.

Drawing for Conversion Socket (EV-9200GC-80) Package and Recommended Board Mounting Pattern

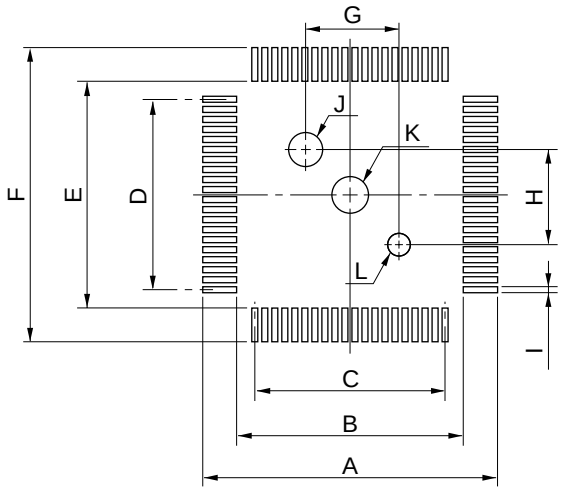
Figure A-2. EV-9200GC-80 Package Drawing (for Reference Only)



EV-9200GC-80-G1E

ITEM	MILLIMETERS	INCHES
A	18.0	0.709
B	14.4	0.567
C	14.4	0.567
D	18.0	0.709
E	4-C 2.0	4-C 0.079
F	0.8	0.031
G	6.0	0.236
H	16.0	0.63
I	18.7	0.736
J	6.0	0.236
K	16.0	0.63
L	18.7	0.736
M	8.2	0.323
N	8.0	0.315
O	2.5	0.098
P	2.0	0.079
Q	0.35	0.014
R	ϕ2.3	ϕ0.091
S	ϕ1.5	ϕ0.059

Figure A-3. EV-9200GC-80 Recommended Board Mounting Pattern (for Reference Only)



EV-9200GC-80-P1E

ITEM	MILLIMETERS	INCHES
A	19.7	0.776
B	15.0	0.591
C	$0.65\pm0.02\times19=12.35\pm0.05$	$0.026^{+0.001}_{-0.002}\times0.748=0.486^{+0.003}_{-0.002}$
D	$0.65\pm0.02\times19=12.35\pm0.05$	$0.026^{+0.001}_{-0.002}\times0.748=0.486^{+0.003}_{-0.002}$
E	15.0	0.591
F	19.7	0.776
G	6.0 ± 0.05	$0.236^{+0.003}_{-0.002}$
H	6.0 ± 0.05	$0.236^{+0.003}_{-0.002}$
I	0.35 ± 0.02	$0.014^{+0.001}_{-0.001}$
J	$\phi2.36\pm0.03$	$\phi0.093^{+0.001}_{-0.002}$
K	$\phi2.3$	$\phi0.091$
L	$\phi1.57\pm0.03$	$\phi0.062^{+0.001}_{-0.002}$

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

APPENDIX B REGISTER INDEX

B.1 Register Index

B.2 Register Index (Symbol)

[W]

APPENDIX C REVISION HISTORY

A history of the revisions up to this edition is shown below. “Applied to:” indicates the chapters to which the revision was applied.

Edition	Description	Applied to:
2nd	Change of μPD178053, 178054, and 178F054 status from under development to development completed	Throughout
	Modification of **Related Documents**	**PREFACE**
	Modification of **1.5 Development of 8-Bit DTS Series**	**CHAPTER 1 OUTLINE**
	Modification of bit units for manipulation for OSTS in **Table 3-4 Special Function Registers**	**CHAPTER 3 CPU ARCHITECTURE**
	Deletion of pins P10 to P15 from **Table 4-3 Port Mode Register and Output Latch Settings When Using Alternate Functions**	**CHAPTER 4 PORT FUNCTIONS**
	Modification of description in **(3) Oscillation stabilization time select register (OSTS)** in **8.3 Registers Controlling Watchdog Timer**	**CHAPTER 8 WATCHDOG TIMER**
	Addition of electrical specifications	**CHAPTER 19 ELECTRICAL SPECIFICATIONS**
	Addition of package drawing	**CHAPTER 20 PACKAGE DRAWING**
	Addition of recommended soldering conditions	**CHAPTER 21 RECOMMENDED SOLDERING CONDITIONS**
	Modification of **Figure A-1 Configuration of Development Tools**	**APPENDIX A DEVELOPMENT TOOLS**
	Addition of **A.1 Software Package** and **A.3 Control Software**	
	Addition of **Note 2** to **A.2 Language Processing Software**	
	Addition of description for IE-78K0-NS-A to **A.5 Debugging Tools (Hardware)**	
	Deletion of MX78K0 from **A.7 Embedded Software**	

NEC

Facsimile Message

From:

Name

Company

Tel. FAX

Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: +1-800-729-9288
+1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Market Communication 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: +82-2-528-4411

Taiwan
NEC Electronics Taiwan Ltd.
Fax: +886-2-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: +81- 44-435-9608

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	❑	❑	❑	❑