NEC

User's Manual

μPD789188 Subseries

8-Bit Single-Chip Microcontrollers

μPD789188
μPD78F9189

1st	30th/May/2003
2nd	2nd/Jul./2003

[MEMO]

SUMMARY OF CONTENTS

CHAPTER 1 GENERAL 12

CHAPTER 2 PIN FUNCTIONS 22

CHAPTER 3 CPU ARCHITECTURE 30

CHAPTER 4 PORT FUNCTIONS 52

CHAPTER 5 CLOCK GENERATION CIRCUIT 71

CHAPTER 6 16-BIT TIMER 79

CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 94

CHAPTER 8 WATCH TIMER 112

CHAPTER 9 WATCHDOG TIMER 118

CHAPTER 10 8-BIT A/D CONVERTER 124

CHAPTER 11 SERIAL INTERFACE 20 137

CHAPTER 12 MULTIPLIER 173

CHAPTER 13 INTERRUPT FUNCTIONS 177

CHAPTER 14 STANDBY FUNCTION 193

CHAPTER 15 RESET FUNCTION 201

CHAPTER 16 μPD78F9189 205

CHAPTER 17 MASK OPTION 211

CHAPTER 18 INSTRUCTION SET 213

CHAPTER 19 ELECTRICAL SPECIFICATION 223

CHAPTER 20 PACKAGE DRAWING 236

CHAPTER 21 RECOMMENED SOLDERING CONDITIONS 237

APPENDIX A DEVELOPMENT TOOLS 238

APPENDIX B RELATED DOCUMENTS 240

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.

- **The information in this document is subject to change without notice. Before using this document, please confirm that this is the latest version.**
- **Not all devices/types available in every country. Please check with local NEC representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.
- NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation or others.
- 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 the customer's equipment shall be done under the full responsibility of the customer. NEC Corporation assumes no responsibility for any losses incurred by the customer or third parties arising from the use of these circuits, software, and information.
- While NEC Corporation has been making continuous effort to enhance the reliability of its semiconductor devices, the possibility of defects cannot be eliminated entirely. To minimize risks of damage or injury to persons or property arising from a defect in an NEC semiconductor device, customers must incorporate sufficient safety measures in its design, such as redundancy, fire-containment, and anti-failure features.
- NEC devices are classified into the following three quality grades:
  "Standard", "Special", and "Specific". The Specific quality grade applies only to devices developed based on a customer designated "quality assurance program" for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device before using it in a particular application.

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

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

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

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

M7D 98.12

Regional Information

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

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

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

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

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

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

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

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

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

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

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

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

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

NEC Electronics Singapore Pte. Ltd.
United Square, Singapore 1130
Tel: 65-253-8311
Fax: 65-250-3583

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

NEC do Brasil S.A.
Electron Devices Division
Rodovia Presidente Dutra, Km 214
07210-902-Guarulhos-SP Brasil
Tel: 55-11-6465-6810
Fax: 55-11-6465-6829

J99.1

CONTENTS

CHAPTER 1 GENERAL 12

1.1 Features 12
1.2 Applications 12
1.3 Ordering Information 13
1.4 Pin Configuration (Top View) 14
1.5 78K/0S Series Development 16
1.6 Block Diagram 18
1.7 Outline of Function 19

CHAPTER 2 PIN FUNCTIONS 22

2.1 Pin Function List 22
2.2 Description of Pin Functions 24
2.2.1 P00 to P04 (Port 0) 24
2.2.2 P10, P11 (Port 1) 24
2.2.3 P20 to P26 (Port 2) 24
2.2.4 P30 to P33 (Port 3) 25
2.2.5 P50 to P53 (Port 5) 25
2.2.6 P60 to P67 (Port 6) 26
2.2.7 $\overline{\text{RESET}}$ 26
2.2.8 X1, X2 26
2.2.9 V_{DD} 26
2.2.10 V_{SS} 26
2.2.11 V_{PP} (μPD78F9189 only) 26
2.2.12 IC0 (μPD789188 only) 27
2.3 Pin Input/Output Circuits and Recommended Connection of Unused Pins 28

CHAPTER 3 CPU ARCHITECTURE 30

3.1 Memory Space 30
3.1.1 Internal program memory space 33
3.1.2 Internal data memory (internal high-speed RAM) space 33
3.1.3 Special function register (SFR) area 33
3.1.4 Data memory addressing 33
3.2 Processor Registers 35
3.2.1 Control registers 35
3.2.2 General-purpose registers 38
3.2.3 Special function registers (SFR) 39
3.3 Instruction Address Addressing 42
3.3.1 Relative addressing 42
3.3.2 Immediate addressing 43

3.3.3 Table indirect addressing 44
3.3.4 Register addressing 44
3.4 Operand Address Addressing 45
3.4.1 Direct addressing 45
3.4.2 Short direct addressing 46
3.4.3 Special function register (SFR) addressing 47
3.4.4 Register addressing 48
3.4.5 Register indirect addressing 49
3.4.6 Based addressing 50
3.4.7 Stack addressing 50

CHAPTER 4 PORT FUNCTIONS 52

4.1 Port Functions 52
4.2 Port Configuration 53
4.2.1 Port 0 53
4.2.2 Port 1 54
4.2.3 Port 2 55
4.2.4 Port 3 60
4.2.5 Port 5 63
4.2.6 Port 6 64
4.3 Port Function Control Registers 65
4.4 Operation of Port Functions 69
4.4.1 Writing to I/O port 69
4.4.2 Reading from I/O port 69
4.4.3 Arithmetic operation of I/O port 69

CHAPTER 5 CLOCK GENERATION CIRCUIT 71

5.1 Clock Generation Circuit Functions 71
5.2 Clock Generation Circuit Configuration 71
5.3 Registers Controlling Clock Generation Circuit 73
5.4 System Clock Oscillators 74
5.4.1 Main system clock oscillator 74
5.5 Clock Generation Circuit Operation 77
5.6 Changing Setting of System Clock and CPU Clock 78
5.6.1 Time required for switching between system clock and CPU clock 78

CHAPTER 6 16-BIT TIMER 79

6.1 16-Bit Timer Functions 79
6.2 16-Bit Timer Configuration 80
6.3 Registers Controlling 16-Bit Timer 83
6.4 16-Bit Timer Operation 87
6.4.1 Operation as timer interrupt 87
6.4.2 Operation as timer output 89

6.4.3 Capture operation 90
6.4.4 16-bit timer counter 90 readout 91
6.4.5 Buzzer output operation 92
6.5 Notes on Using 16-Bit Timer 93

CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 94

7.1 Functions of 8-Bit Timer/Event Counters 94
7.2 8-Bit Timer/Event Counter Configuration 96
7.3 8-Bit Timer/Event Counter Control Registers 99
7.4 Operation of 8-Bit Timer/Event Counter 103
7.4.1 Operation as interval timer 103
7.4.2 Operation as external event counter 105
7.4.3 Operation as square wave output 106
7.4.4 PWM output operation 108
7.5 Notes on Using 8-Bit Timer/Event Counters 110

CHAPTER 8 WATCH TIMER 112

8.1 Watch Timer Functions 112
8.2 Watch Timer Configuration 113
8.3 Watch Timer Control Register 114
8.4 Watch Timer Operation 115
8.4.1 Operation as watch timer 115
8.4.2 Operation as interval timer 115

CHAPTER 9 WATCHDOG TIMER 118

9.1 Watchdog Timer Functions 118
9.2 Watchdog Timer Configuration 119
9.3 Watchdog Timer Control Registers 120
9.4 Watchdog Timer Operation 122
9.4.1 Operation as watchdog timer 122
9.4.2 Operation as interval timer 123

CHAPTER 10 8-BIT A/D CONVERTER 124

10.1 8-Bit A/D Converter Functions 124
10.2 8-Bit A/D Converter Configuration 124
10.3 8-Bit A/D Converter Control Registers 127
10.4 8-Bit A/D Converter Operation 129
10.4.1 Basic operation of 8-bit A/D converter 129
10.4.2 Input voltage and conversion result 130
10.4.3 Operation mode of 8-bit A/D converter 131
10.5 Cautions Related to 8-Bit A/D Converter 132

CHAPTER 11 SERIAL INTERFACE 20 137

11.1 Serial Interface 20 Functions 137
11.2 Serial Interface 20 Configuration 137
11.3 Serial Interface 20 Control Registers 141
11.4 Serial Interface 20 Operation 148
11.4.1 Operation stop mode 148
11.4.2 Asynchronous serial interface (UART) mode 150
11.4.3 3-wire serial I/O mode 163

CHAPTER 12 MULTIPLIER 173

12.1 Multiplier Function 173
12.2 Multiplier Configuration 173
12.3 Multiplier Control Register 175
12.4 Multiplier Operation 176

CHAPTER 13 INTERRUPT FUNCTIONS 177

13.1 Interrupt Function Types 177
13.2 Interrupt Sources and Configuration 177
13.3 Interrupt Function Control Registers 180
13.4 Interrupt Processing Operation 186
13.4.1 Non-maskable interrupt request acceptance operation 186
13.4.2 Maskable interrupt request acceptance operation 188
13.4.3 Multiple interrupt processing 190
13.4.4 Interrupt request reserve 192

CHAPTER 14 STANDBY FUNCTION 193

14.1 Standby Function and Configuration 193
14.1.1 Standby function 193
14.1.2 Standby function control register 194
14.2 Operation of Standby Function 195
14.2.1 HALT mode 195
14.2.2 STOP mode 198

CHAPTER 15 RESET FUNCTION 201

CHAPTER 16 μPD78F9189 205

16.1 Flash Memory Programming 206
16.1.1 Selecting communication mode 206
16.1.2 Function of flash memory programming 207
16.1.3 Flashpro III connection 207
16.1.4 Setting with Flashpro III (PG-FP3) 209

CHAPTER 17 MASK OPTION 211

CHAPTER 18 INSTRUCTION SET 213

18.1 Operation 213

18.1.1 Operand identifiers and description methods 213

18.1.2 Description of "Operation" column 214

18.1.3 Description of "Flag" column 214

18.2 Operation List 215

18.3 Instructions Listed by Addressing Type 220

CHAPTER 19 ELECTRICAL SPECIFICATIONS 223

CHAPTER 20 PACKAGE DRAWING 236

CHAPTER 21 RECOMMENDED SOLDERING CONDITIONS 237

APPENDIX A DEVELOPMENT TOOLS 238

APPENDIX B RELATED DOCUMENTS 240

CHAPTER 1 GENERAL

1.1 Features

• ROM and RAM capacity

Item / Product Name	Program Memory (ROM)		Data Memory (Internal High-Speed RAM)
μPD789188	Mask ROM	16 Kbytes	512 bytes
μPD78F9189	Flash memory	24 Kbytes	

- Minimum instruction execution time (0.4 μs: Main system clock 5.0-MHz operation)
- I/O port: 26
- Serial interface
  - 3-wire serial I/O mode/UART mode: 1 channel
- 8-bit resolution A/D converter: 4 channels
- Timer: 6 channels
  - 16-bit timer: 1 channel
  - 8-bit timer/event counter: 2 channels
  - 8-bit timer: 1 channel
  - Watch timer: 1 channel
  - Watchdog timer: 1 channel
- Vectored interrupt source: 15
- Supply voltage: V_{DD} = 4.0 to 5.5 V
- Operating ambient temperature: T_A = –40°C to +85°C

1.2 Applications

Air-Conditioner, Home Appliance, etc.

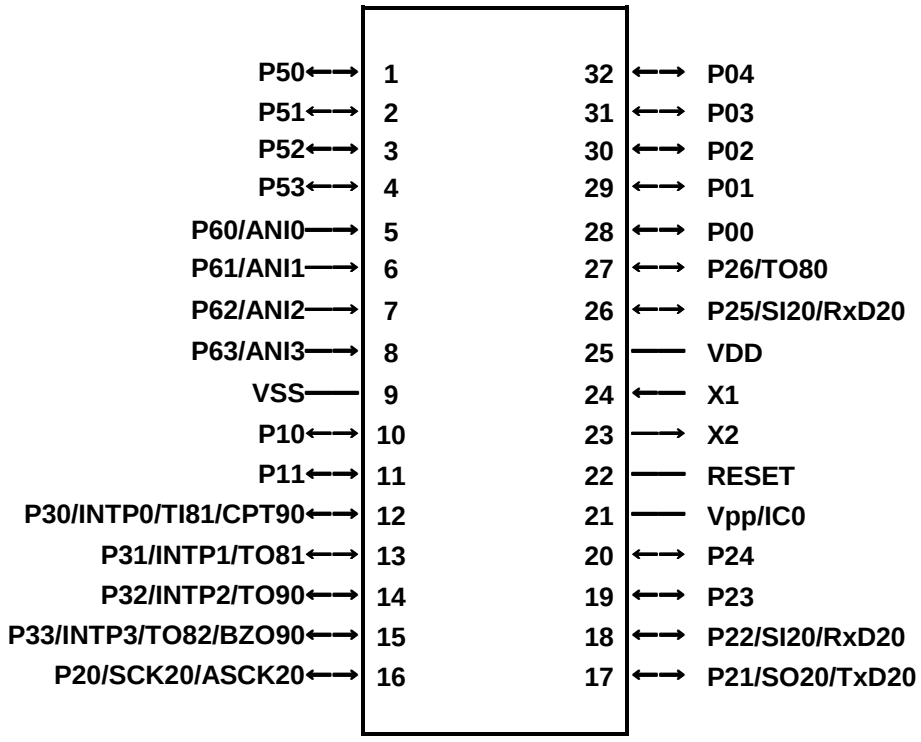
1.3 Ordering Information

Part Number	Package	Internal ROM
μPD789188CT-×××	32-pin plastic Shrink DIP (400mil)	Mask ROM
μPD78F9189CT	32-pin plastic Shrink DIP (400mil)	Flash memory

Remark ××× indicates ROM code suffix.

1.4 Pin Configuration (Top View)

32-pin plastic Shrink DIP (400mil)
μPD789188CT-××× μPD78F9189CT



Cautions 1. Connect the IC0 (internally connected) pin directly to the V_{SS} pin.

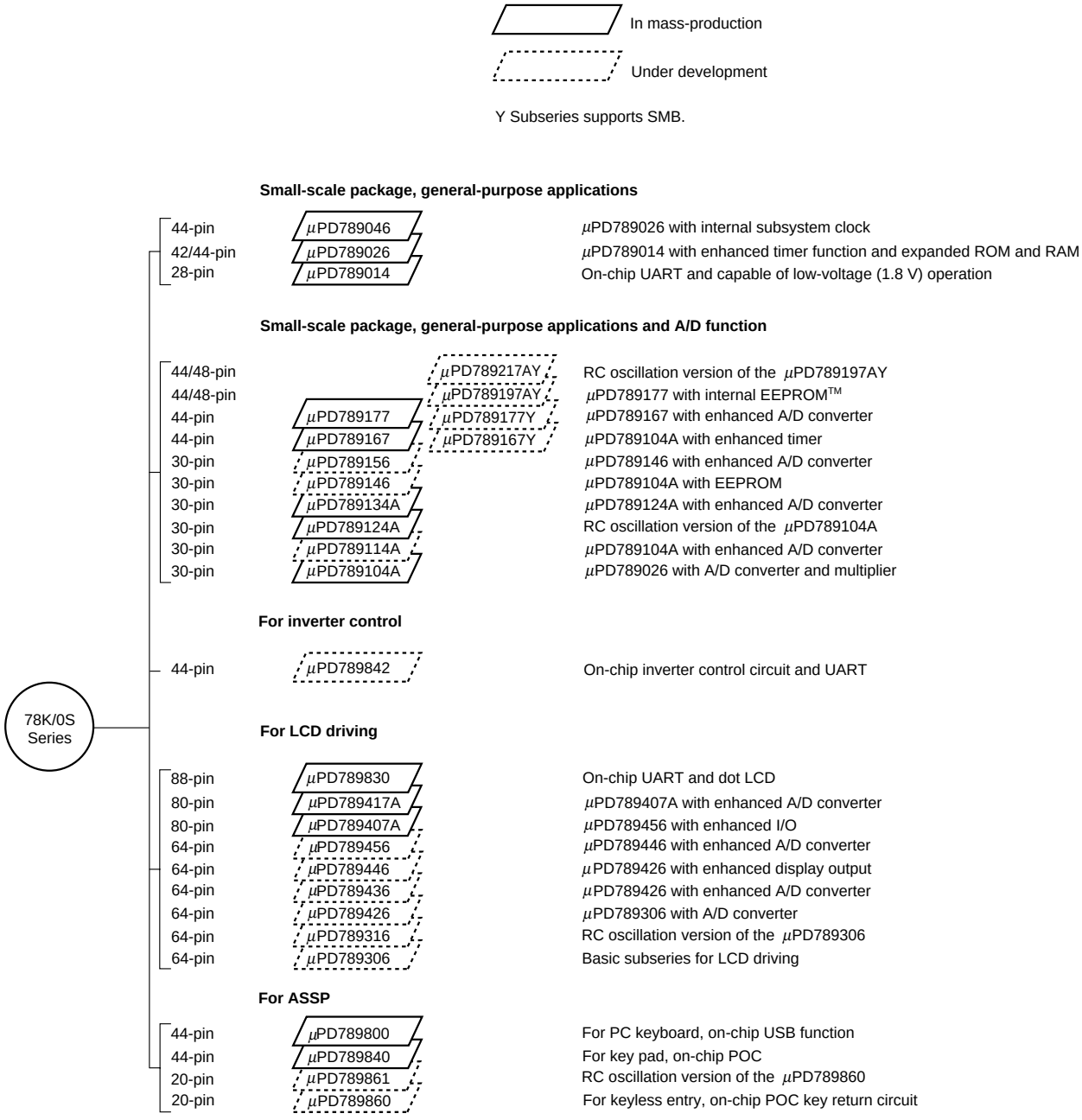
Remark Pin connections in parentheses are intended for the μPD78F9189.

ANI0 to ANI3:	Analog Input
ASCK20:	Asynchronous Serial Input
BZO90:	Buzzer Output
CPT90:	Capture Trigger Input
IC0:	Internally Connected
INTP0 to INTP3:	Interrupt from Peripherals
P00 to P04:	Port 0
P10, P11:	Port 1
P20 to P26:	Port 2
P30 to P33:	Port 3
P50 to P53:	Port 5
P60 to P63:	Port 6

$\overline{\text{RESET}}$:	Reset
RxD20:	Receive Data
$\overline{\text{SCK20}}$:	Serial Clock
SI20:	Serial Input
SO20:	Serial Output
$\overline{\text{SS20}}$:	Chip Select Input
TI80, TI81:	Timer Input
TO80 to TO82, TO90:	Timer Output
TxD20:	Transmit Data
V_{DD}:	Power Supply
V_{PP}:	Programming Power Supply
V_{SS}:	Ground
X1, X2:	Crystal (Main System Clock)

1.5 78K/0S Series Development

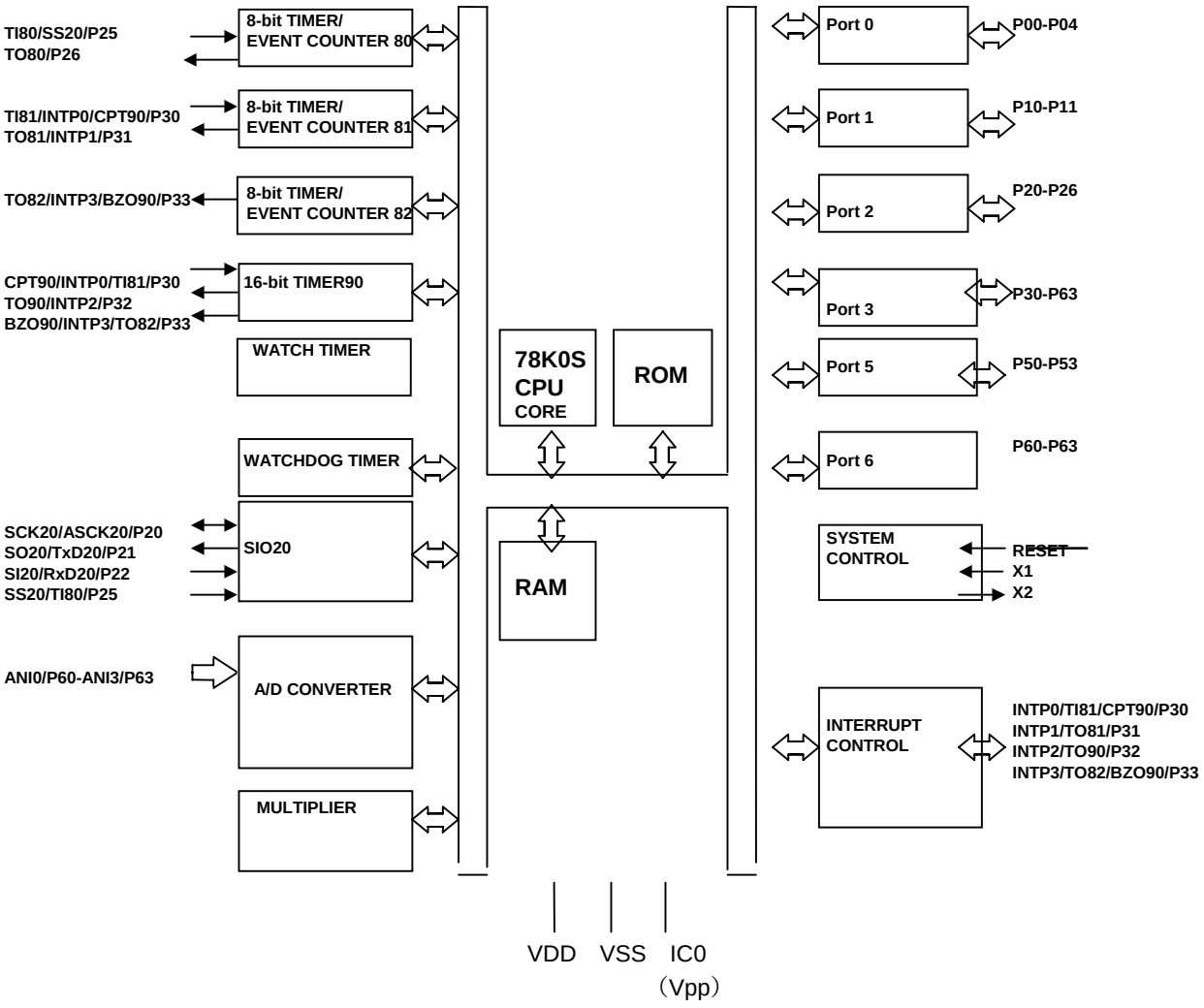
The 78K/0S Series products are shown below. The subseries names are indicated in frames.



The major functional differences among the subseries are listed below.

Function / Subseries Name	ROM Capacity	Timer				8-Bit A/D	10-Bit A/D	6685(16I/O)6(e)-7.1ac			
		8-Bit	16-Bit	Watch	WDT						

1.6 Block Diagram



Remarks 1. Pin connections in parentheses are intended for the μPD78F9189.

1.7 Outline of Function

Item \ Part Number		μPD789188	μPD78F9189
Internal memory	ROM	Mask ROM	Flash Memory
		16 Kbytes	24 Kbytes
	High-speed RAM	512 bytes	
Minimum instruction execution time		0.4/1.6 μs (operation with main system clock running at 5.0 MHz)	
General-purpose registers		8 bits × 8 registers	
Instruction set		• 16-bit operations • Bit manipulations (such as set, reset, and test)	
Multiplier		8 bits × 8 bits = 16 bits	
I/O ports		Total: 26 • CMOS input: 4(Shared with A/D Converter) • CMOS I/O: 16 • N-ch open-drain: 6(12V tolerance 4)	
A/D converter		• 8-bit resolution × 8 channels	
Serial interface		• Switchable between 3-wire serial I/O and UART modes: 1 channel	
Timers		• 16-bit timer: 1 channel • 8-bit timer/event counter: 2 channels • 8-bit timer: 1 channel • Watch timer: 1 channel • Watchdog timer: 1 channel	
Timer output		Four outputs	
Buzzer output		One output	
Vectored interrupt sources	Maskable	Internal: 10, external: 4	
	Nonmaskable	Internal: 1	
Power supply voltage		V_{DD} = 4.0 to 5.5 V	
Operating ambient temperature		T_A = –40°C to +85°C	
Package		32-pin plastic Shrink DIP(400mil)	

The timers are outlined below.

		16-Bit Timer 90	8-Bit Timer/Event Counter 80	8-Bit Timer/Event Counter 81	8-Bit Timer 82	Watch Timer	Watchdog Timer
Operating mode	Interval timer	–	1 channel	1 channel	1 channel	1 channel Note 1	1 channel Note 2
	External event counter	–	1 channel	1 channel	–	–	–
Function	Timer output	1 output	1 output	1 output	1 output	–	–
	PWM output	–	1 output	1 output	1 output	–	–
	Square-wave output	–	1 output	1 output	1 output	–	–
	Buzzer output	1 output	–	–	–	–	–
	Capture	1 input	–	–	–	–	–
	Interrupt source	1	1	1	1	1	1

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

2. The watchdog timer provides the watchdog timer function and interval timer function. Use either of the functions.

[MEMO]

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

(1) Port pins

Pin Name	I/O	Function	After Reset	Alternate Function
P00 to P04	I/O	Port 0 5-bit input/output port Input/output mode can be specified in 1-bit units When used as an input port, an on-chip pull-up resistor can be specified by means of pull-up resistor option register 0 (PU0).	Input	–
P10, P11	I/O	Port 1 2-bit input/output port Input/output mode can be specified in 1-bit units When used as an input port, an on-chip pull-up resistor can be specified by means of pull-up resistor option register 0 (PU0).	Input	–
P20	I/O	Port 2 7-bit input/output port Input/output mode can be specified in 1-bit units For P20 to P22, P25, and P26, an on-chip pull-up resistor can be specified by means of pull-up resistor option register B2 (PUB2). Only P23 and P24 can be used as N-ch open-drain input/output port pins.	Input	$\overline{\text{SCK20}}$/ASCK20
P21				SO20/TxD20
P22				SI20/RxD20
P23				–
P24				–
P25				TI80/$\overline{\text{SS20}}$
P26				TO80
P30	I/O	Port 3 4-bit input/output port Input/output mode can be specified in 1-bit units An on-chip pull-up resistor can be specified by means of pull-up resistor option register B3 (PUB3).	Input	INTP0/TI81/CPT90
P31				INTP1/TO81
P32				INTP2/TO90
P33				INTP3/TO82/BZO90
P50 to P53	I/O	Port 5 4-bit N-ch open-drain input/output port Input/output mode can be specified in 1-bit units For a mask ROM version, an on-chip pull-up resistor can be specified by the mask option.	Input	–
P60 to P63	Input	Port 6 4-bit input-only port	Input	ANI0 to ANI3

(2) Non-port pins

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	Input	P30/TI81/CPT90
INTP1				P31/TO81
INTP2				P32/TO90
INTP3				P33/TO82/BZO90
SI20	Input	Serial data input to serial interface	Input	P22/RxD20
SO20	Output	Serial data output from serial interface	Input	P21/TxD20
SCK20	I/O	Serial clock input/output for serial interface	Input	P20/ASCK20
SS20	Input	Chip select input to serial interface	Input	P25/TI80
ASCK20	Input	Serial clock input for asynchronous serial interface	Input	P20/$\overline{\text{SCK20}}$
RxD20	Input	Serial data input for asynchronous serial interface	Input	P22/SI20
TxD20	Output	Serial data output for asynchronous serial interface	Input	P21/SO20
TI80	Input	External count clock input to 8-bit timer/event counter (TM80)	Input	P25/$\overline{\text{SS20}}$
TI81	Input	External count clock input to 8-bit timer/event counter (TM81)	Input	P30/INTP0/CPT90
TO80	Output	8-bit timer/event counter (TM80) output	Input	P26
TO81	Output	8-bit timer/event counter (TM81) output	Input	P31/INTP1
TO82	Output	8-bit timer (TM82) output	Input	P33/INTP3/BZO90
TO90	Output	16-bit timer (TM90) output	Input	P32/INTP2
CPT90	Input	Capture edge input	Input	P30/INTP0/TI81
BZO90	Output	Buzzer output	Input	P33/INTP3/TO82
ANI0 to ANI3	Input	A/D converter analog input	Input	P60 to P63
X1	Input	Connecting crystal resonator for main system clock oscillation	–	–
X2	–		–	–
$\overline{\text{RESET}}$	Input	System reset input	Input	–
V_{DD}	–	Positive power supply	–	–
V_{SS}	–	Ground potential	–	–
IC0	–	Internally connected. Connect this pin directly to the V_{SS} pin.	–	–
V_{PP}	–	This pin is used to set flash memory programming mode and applies a high voltage when a program is written or verified. In normal operation mode, connect this pin directly to the V_{SS}.	–	–

2.2 Description of Pin Functions

2.2.1 P00 to P04 (Port 0)

These pins constitute a 5bit I/O port and can be set to input or output port mode in 1-bit units by using port mode register 0 (PM0). When these pins are used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 0 (PU0).

2.2.2 P10, P11 (Port 1)

These pins constitute a 2-bit I/O port and can be set to input or output port mode in 1-bit units by using port mode register 1 (PM1). When these pins are used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 0 (PU0).

2.2.3 P20 to P26 (Port 2)

These pins constitute a 7-bit I/O port. In addition, these pins provide a function to perform input/output to/from the timer and to input/output the data and clock of the serial interface.

Port 2 can be set to the following operation modes in 1-bit units.

(1) Port mode

In port mode, P20 to P26 function as a 7-bit I/O port. Port 2 can be set to input or output mode in 1-bit units by using port mode register 2 (PM2). For P20 to P22, P25, and P26, whether to use on-chip pull-up resistors can be specified in 1-bit units by using pull-up resistor option register B2 (PUB2), regardless of the setting of port mode register 2 (PM2). P23 and P24 are N-ch open-drain I/O ports.

(2) Control mode

In this mode, P20 to P26 function as the timer input/output, the data input/output and the clock input/output of the serial interface.

(a) TI80

This is the external clock input pin for 8-bit timer/event counter 80.

(b) TO80

This is the timer output pin of 8-bit timer/event counter 80.

(c) SI20, SO20

These are the serial data I/O pins of the serial interface.

(d) $\overline{\text{SCK20}}$

These are the serial clock I/O pins of the serial interface.

(e) $\overline{\text{SS20}}$

This is the chip select input pin of the serial interface.

(f) RxD20, TxD20

These are the serial data I/O pins of the asynchronous serial interface.

(g) ASCK20

This is the serial clock input pin of the asynchronous serial interface.

Caution When using P20 to P26 as serial interface pins, the input/output mode and output latch must be set according to the functions to be used. For details of the setting, see Table 14-2 Serial Interface 20 Operating Mode Settings.

2.2.4 P30 to P33 (Port 3)

These pins constitute a 4-bit I/O port. In addition, these pins function as the timer input/output and the external interrupt input.

Port 3 can be set to the following operation modes in 1-bit units.

(1) Port mode

In port mode, P30 to P33 function as a 4-bit I/O port. Port 3 can be set to input or output mode in 1-bit units by using port mode register 3 (PM3). Whether to use the on-chip pull-up resistor can be specified in 1-bit units by using pull-up resistor option register B3 (PUB3), regardless of the setting of port mode register 3 (PM3).

(2) Control mode

In this mode, P30 to P33 function as the timer input/output and the external interrupt input.

(a) TI81

This is the external clock input pin for 8-bit timer/event counter 81.

(b) TO90, TO81, TO82

These are the output pins of 16-bit timer 90, 8-bit timer/event counter 81, and 8-bit timer 82.

(c) CPT90

This is the capture edge input pin of 16-bit timer 90.

(d) BZO90

This is the buzzer output pin of 16-bit timer 90.

(e) INTP0 to INTP3

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

2.2.5 P50 to P53 (Port 5)

These pins constitute a 4-bit N-ch open-drain I/O port. Port 5 can be set to input or output mode in 1-bit units by using port mode register 5 (PM5). For a mask ROM version, whether a pull-up resistor is to be incorporated can be specified by a mask option.

2.2.6 P60 to P63 (Port 6)

These pins constitute an 4-bit input-only port. They can function as A/D converter input pins as well as a general-purpose input port.

(1) Port mode

In port mode, P60 to P63 function as an 8-bit input-only port.

(2) Control mode

In control mode, P60 to P63 function as A/D converter analog inputs (ANI0 to ANI3).

2.2.7 $\overline{\text{RESET}}$

A low-level active system reset signal is input to this pin.

2.2.8 X1, X2

These pins are used to connect a crystal resonator for main system clock oscillation.

To supply an external clock, input the clock to X1 and input the inverted signal to X2.

2.2.13 V_{DD}

V_{DD} is a positive power supply port pin.

2.2.14 V_{SS}

V_{SS} is a ground potential port pin.

2.2.15 V_{PP} (μPD78F9189 only)

High voltage apply pin for flash memory programming mode setting and program write/verify.

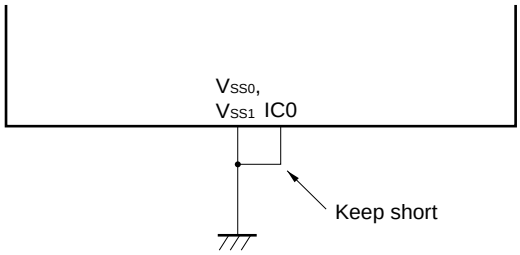
Directly connect this pin to V_{SS} in normal operation mode.

2.2.16 IC0 (mask ROM version only)

The IC0 (Internally Connected) pin is used to set the μPD789188 to test mode before shipment. In normal operation mode, directly connect this pin to the V_{SS} pin with as short a wiring length as possible.

If a potential difference is generated between the IC0 pin and V_{SS} pin due to a long wiring length between the IC0 pin and V_{SS} pin or an external noise superimposed on the IC0 pin, a user program may not run correctly.

- Directly connect the IC0 pin to the V_{SS} pin.



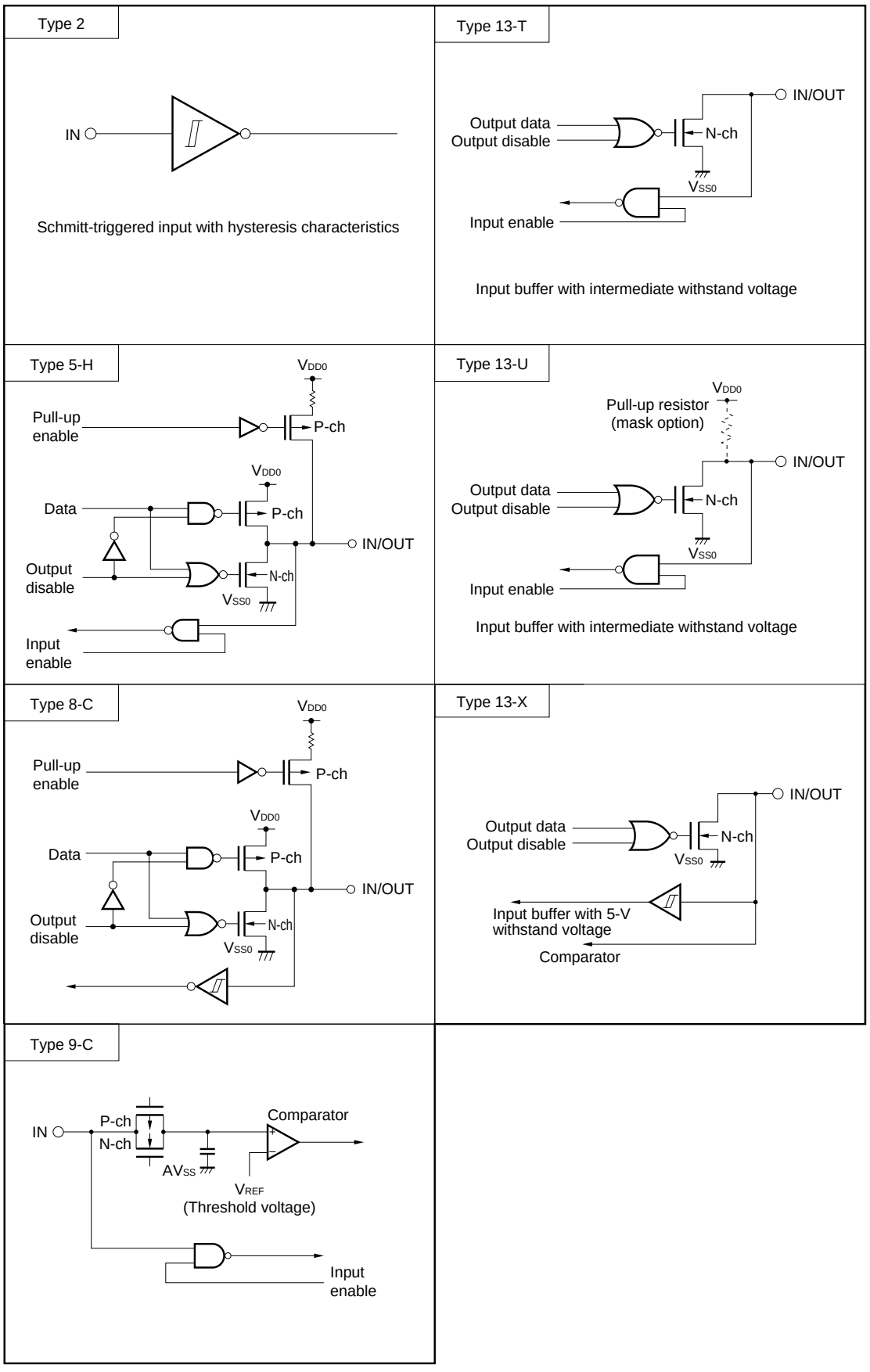
2.3 Pin Input/Output Circuits and Recommended Connection of Unused Pins

The input/output circuit type of each pin and recommended connection of unused pins are shown in Table 2-1. For the input/output circuit configuration of each type, refer to Figure 2-1.

Table 2-1. Types of Input/Output Circuits for Each Pin and Recommended Connection of Unused Pins

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00 to P04	5-H	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P10, P11			
P20/$\overline{SCK20}$/ASCK20	8-C		
P21/SO20/TxD20			
P22/SI20/RxD20			
P23	13-X		Input: Independently connect to V_{DD} via a resistor. Output: Leave open.
P24			
P25/TI80/$\overline{SS20}$	8-C		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P26/TO80			
P30/INTP0/TI81/CPT90			Input: Independently connect to V_{SS} via a resistor. Output: Leave open.
P31/INTP1/TO81			
P32/INTP2/TO90			
P33/INTP3/TO82/BZO90			
P50 to P53 (mask ROM version)	13-U		Input: Independently connect to V_{DD} via a resistor. Output: Leave open.
P50 to P53 (flash memory version)	13-T		
P60/ANI0 to P63/ANI3	9-C	Input	Connect directly to V_{DD} or V_{SS}.
$\overline{RESET}$	2	Input	–
IC0 (mask ROM version)	–	–	Connect directly to V_{SS}.
V_{PP} (flash memory version)			Connect directly to V_{SS}.

Figure 2-1. Pin Input/Output Circuits



CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the μPD789188 and 78F9189 Subseries can each access up to 64 Kbytes of memory space. Figures 3-1 through 3-2 show the memory maps.

Figure 3-1. Memory Map (μPD789188)

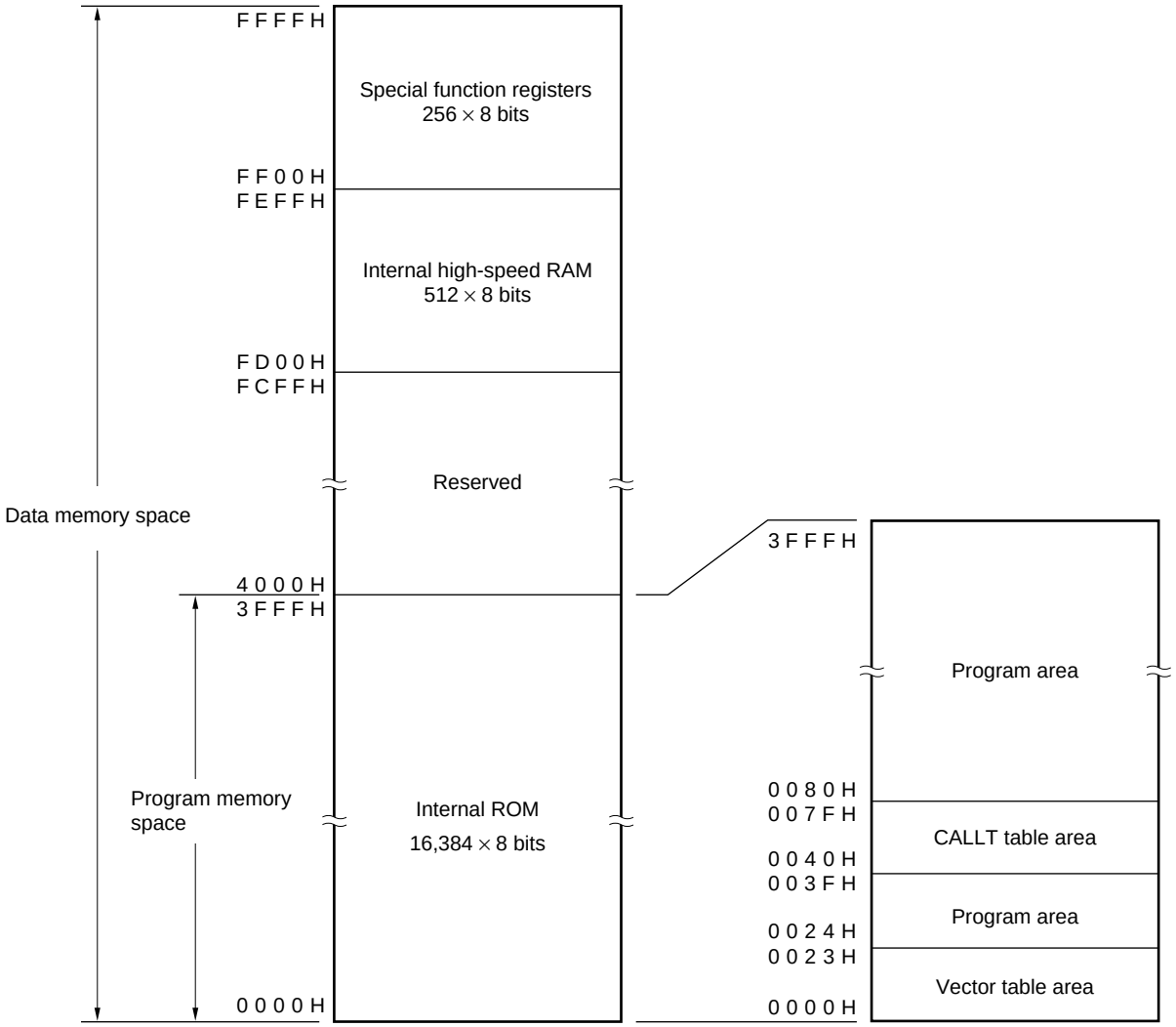
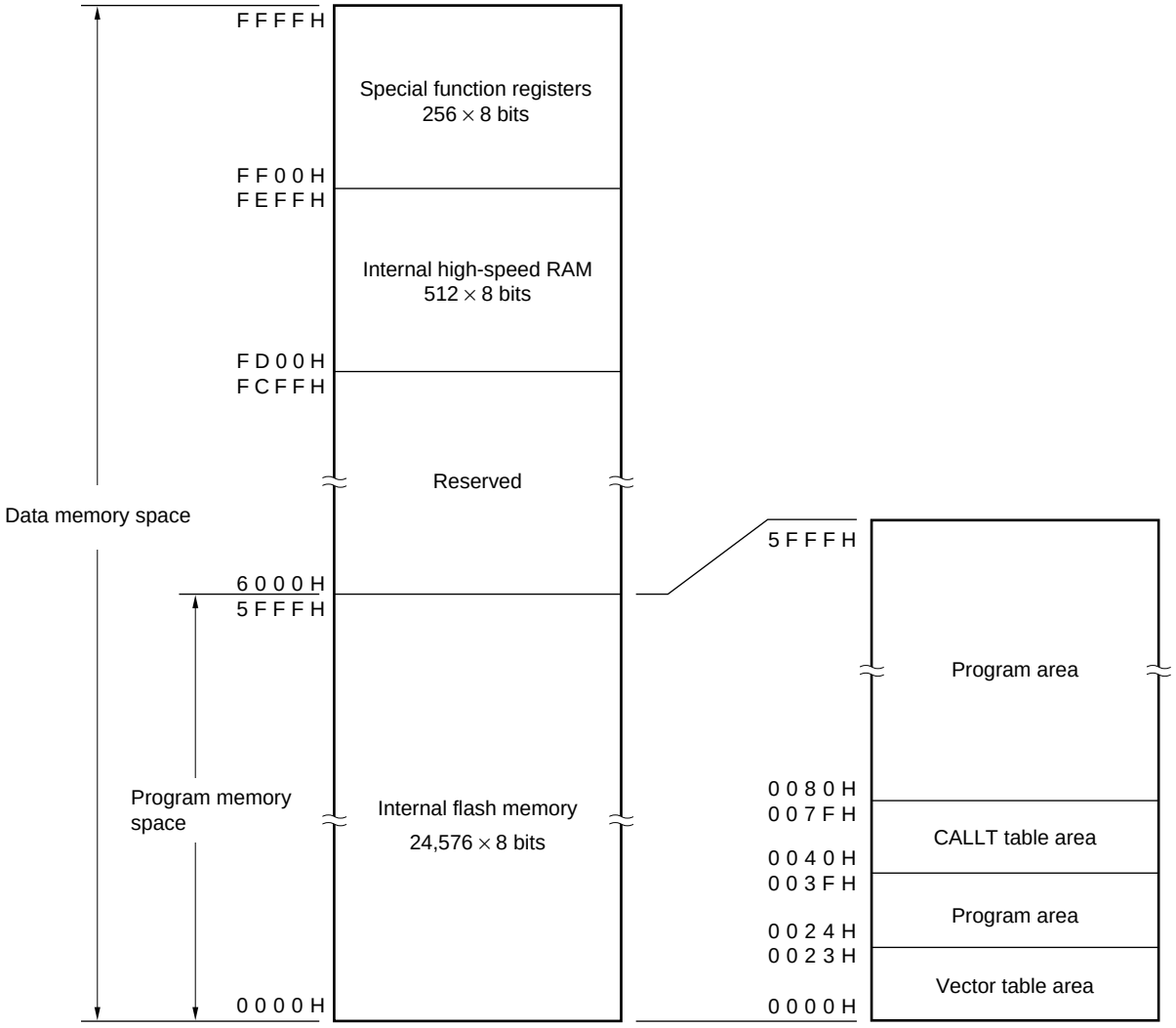


Figure 3-2. Memory Map (μPD78F9189)



3.1.1 Internal program memory space

The internal program memory space stores programs and table data. This space is usually addressed by the program counter (PC).

The μPD789188 and 78F9189 Subseries provide the following internal ROMs (or flash memory) containing the following capacities.

Table 3-1. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD789188	Mask ROM	16,384 × 8 bits
μPD78F9189	Flash memory	24,576 × 8 bits

The following areas are allocated to the internal program memory space:

(1) Vector table area

A 36-byte area of addresses 0000H to 0023H is reserved as a vector table area. This area stores program start addresses to be used when branching by the $\overline{\text{RESET}}$ input or an interrupt request generation. Of a 16-bit program address, the lower 8 bits are stored in an even address, and the higher 8 bits are stored in an odd address.

Table 3-2. Vector Table

Vector Table Address	Interrupt Request	Vector Table Address	Interrupt Request
0000H	$\overline{\text{RESET}}$ input	0014H	INTWTI
0004H	INTWDT	0016H	INTTM80
0006H	INTP0	0018H	INTTM81
0008H	INTP1	001AH	INTTM82
000AH	INTP2	001CH	INTTM90
000CH	INTP3	001EH	-
000EH	INTSR20/INTCSI20	0020H	-
0010H	INTST20	0022H	INTAD0
0012H	INTWT		

Note

(2) CALLT instruction table area

In a 64-byte area of addresses 0040H to 007FH, the subroutine entry address of a 1-byte call instruction (CALLT) can be stored.

3.1.2 Internal data memory (internal high-speed RAM) space

The μPD789188 and 78F9189 Subseries provide 512-byte internal high-speed RAM.

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

3.1.3 Special function register (SFR) area

Special function registers (SFRs) of on-chip peripheral hardware are allocated to an area of FF00H to FFFFH (see **Table 3-3**).

3.1.4 Data memory addressing

Each of the μPD789188 Subseries is provided with a wide range of addressing modes to make memory manipulation as efficient as possible. A data memory area (FD00H to FFFFH) can be accessed using a unique addressing mode according to its use, such as a special function register (SFR). Figures 5-4 through 5-6 illustrate the data memory addressing modes.

Figure 3-4. Data Memory Addressing Modes (μPD789188)

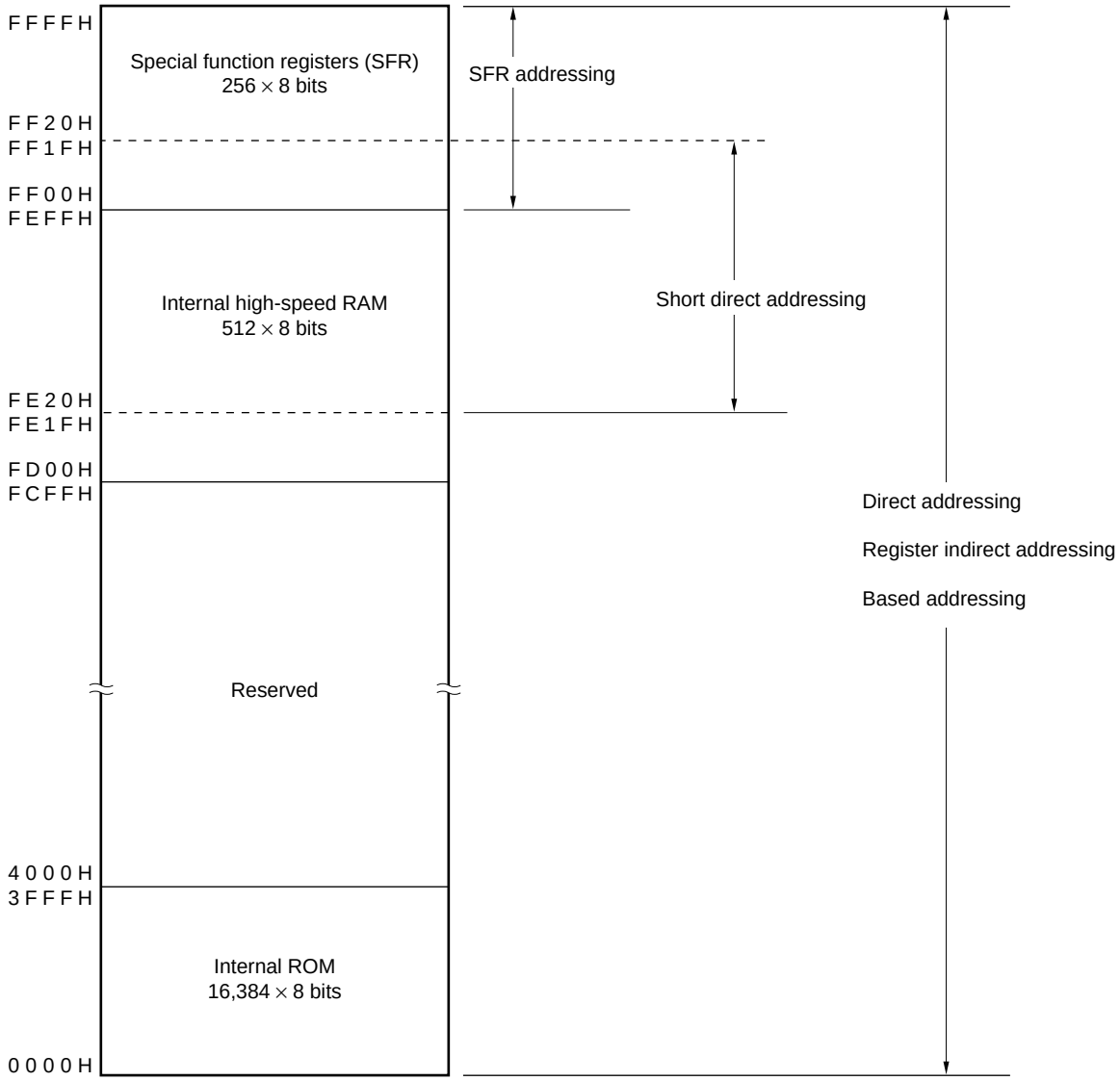
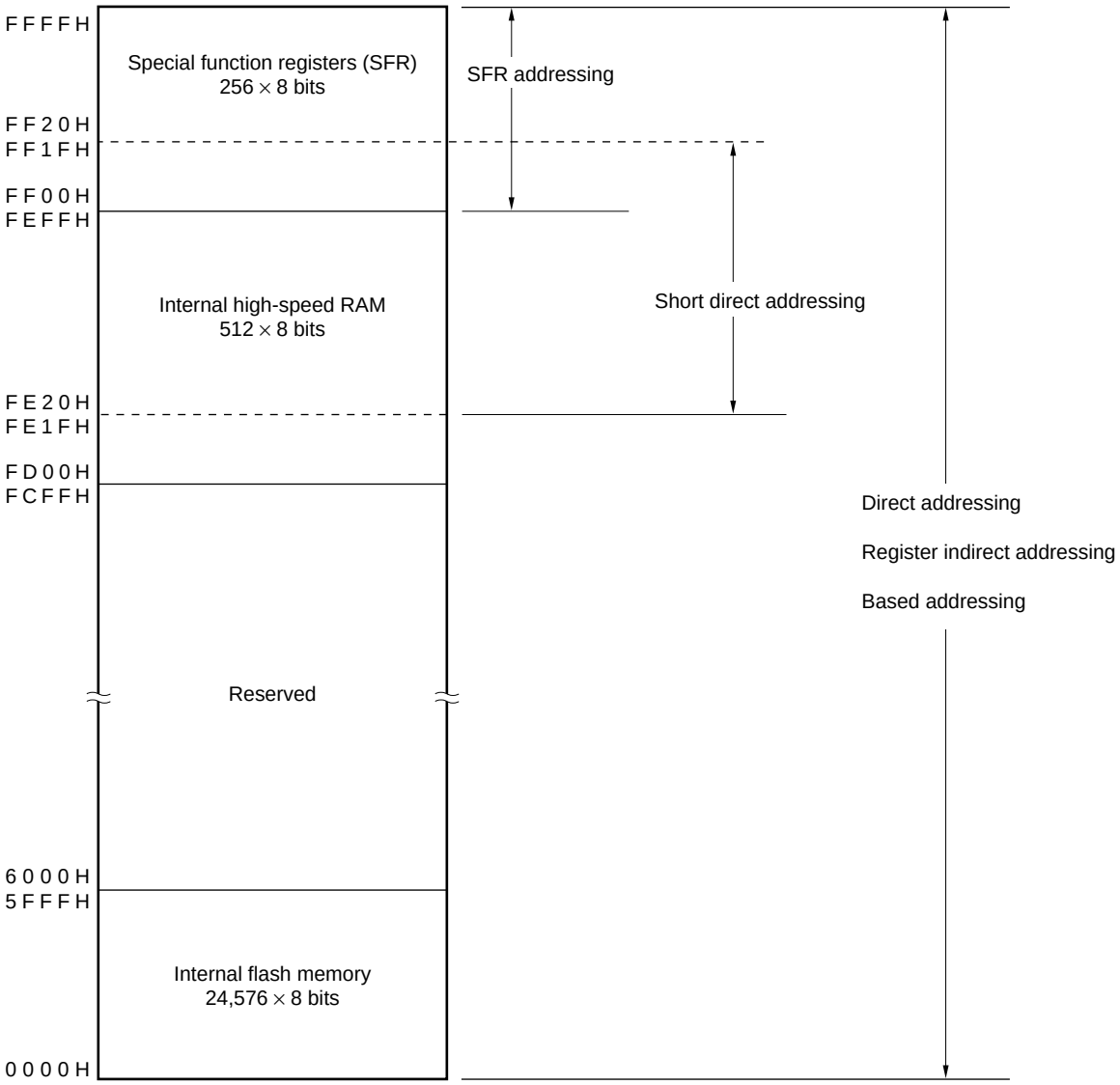


Figure 3-6. Data Memory Addressing Modes (μPD78F9189)



3.2 Processor Registers

The μPD789188 and 78F9189 Subseries provide the following on-chip processor registers:

3.2.1 Control registers

The control registers have special functions to control the program sequence statuses and stack memory. The control registers include a program counter, a program status word, and a stack pointer.

(1) Program counter (PC)

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

(a) Interrupt enable flag (IE)

This flag controls interrupt request acknowledge operations of CPU.

When IE = 0, the interrupt disabled (DI) status is set. All interrupt requests except non-maskable interrupt are disabled.

When IE = 1, the interrupt enabled (EI) status is set. Interrupt request acknowledgement is controlled with an interrupt mask flag for various interrupt sources.

This flag is reset to 0 upon DI instruction execution or interrupt acknowledgment 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) 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.

(d) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Figure 3-9. Stack Pointer Configuration

	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 writing (saving) to the stack memory and is incremented after reading (restoring) from the stack memory.

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

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

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

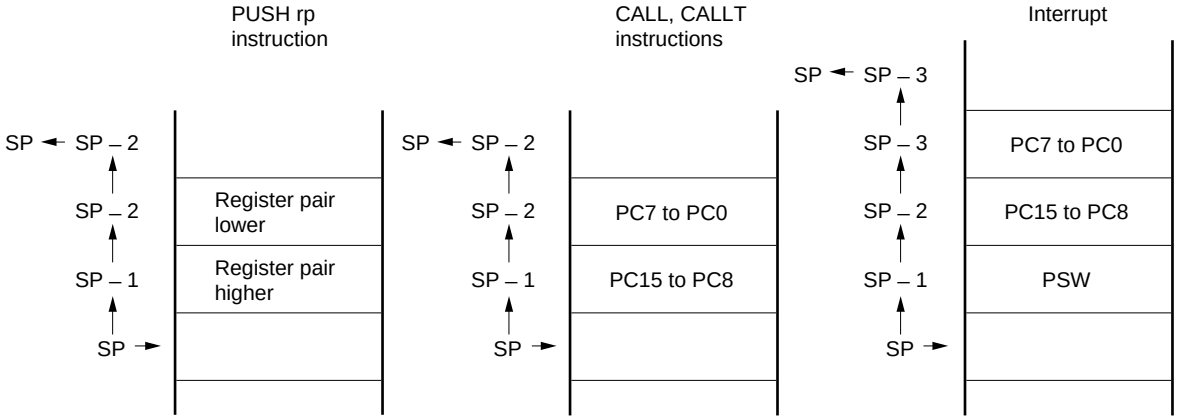
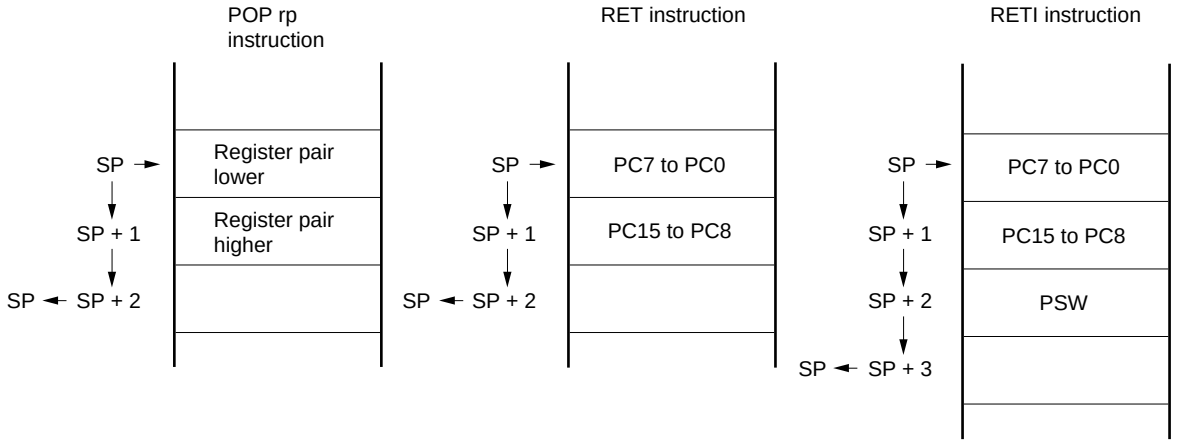


Figure 3-11. Data to be Restored from Stack Memory



3.2.2 General-purpose registers

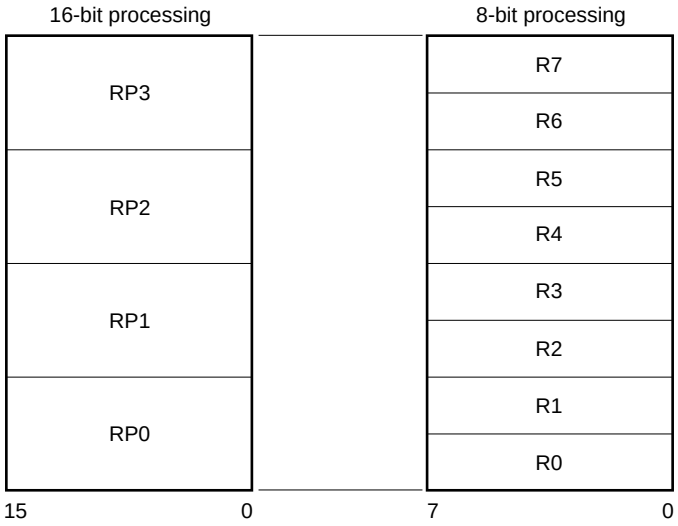
The general-purpose registers consist of eight 8-bit registers (X, A, C, B, E, D, L, and H).

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

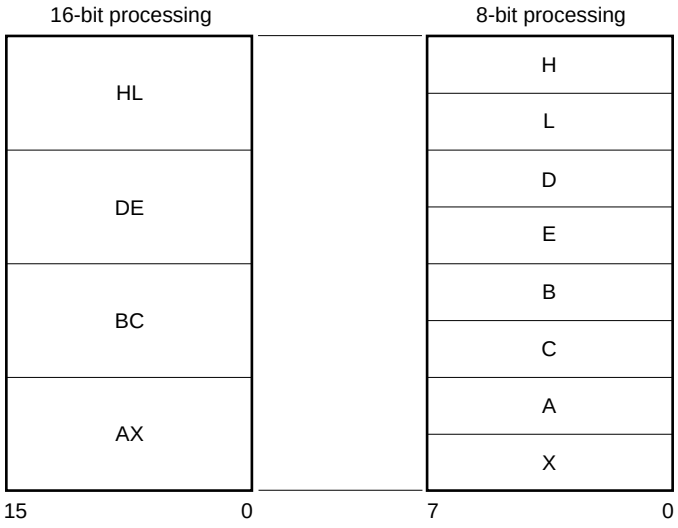
They can be described in terms of functional names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute names (R0 to R7 and RP0 to RP3).

Figure 3-12. General-Purpose Register Configuration

(a) Absolute Names



(b) Functional Names



3.2.3 Special function registers (SFR)

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

They are allocated to the 256-byte area FF00H to FFFFH.

The special function registers can be manipulated, like the general-purpose registers, with the operation, transfer, and bit manipulation instructions. Manipulatable bit units (1, 8, and 16) differ depending on the special function register type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describes a symbol reserved with assembler for the 1-bit manipulation instruction operand (sfr.bit). This manipulation can also be specified with an address.

- 8-bit manipulation
  Describes a symbol reserved with assembler for the 8-bit manipulation instruction operand (sfr). This manipulation can also be specified with an address.

- 16-bit manipulation
  Describes a symbol reserved with assembler for the 16-bit manipulation instruction operand. When specifying an address, describe an even address.

Table 5-3 lists the special function registers. The meanings of the symbols in this table are as follows:

- Symbol
  Indicates the addresses of the implemented special function registers. The symbols shown in this column are the reserved words of the assembler, and have already been defined in the header file called "sfrbit.h" of C compiler. Therefore, these symbols can be used as instruction operands if assembler or integrated debugger is used.

- R/W
  Indicates whether the special function register can be read or written.
  R/W: Read/write
  R: Read only
  W: Write only

- Bit units for manipulation
  Indicates the bit units (1, 8, and 16) in which the special function register can be manipulated.

- After reset
  Indicates the status of the special function register when the $\overline{\text{RESET}}$ signal is input.

Table 3-3. Special Function Registers (1/2)

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			√	√	–	
FF02H	Port 2	P2			√	√	–	
FF03H	Port 3	P3			√	√	–	
FF05H	Port 5	P5			√	√	–	
FF06H	Port 6	P6		R	√	√	–	
FF10H	16-bit multiplication result storage register 0	MUL0L	MUL0		–	–	√ **Notes 1, 2**	Undefined
FF11H		MUL0H						
FF15H	A/D conversion result register 0	ADCR0				√	–	
FF16H	16-bit compare register 90	CR90L	CR90	W	–	–	√ **Notes 1, 2**	FFFFH
FF17H		CR90H						
FF18H	16-bit timer counter 90	TM90L	TM90	R	–	–	√ **Notes 1, 2**	0000H
FF19H		TM90H						
FF1AH	16-bit capture register 90	TCP90L	TCP90		–	–	√ **Notes 1, 2**	Undefined
FF1BH		TCP90H						
FF20H	Port mode register 0	PM0		R/W	√	√	–	FFH
FF21H	Port mode register 1	PM1			√	√	–	
FF22H	Port mode register 2	PM2			√	√	–	
FF23H	Port mode register 3	PM3			√	√	–	
FF25H	Port mode register 5	PM5			√	√	–	
FF32H	Pull-up resistor option register B2	PUB2			√	√	–	00H
FF33H	Pull-up resistor option register B3	PUB3			√	√	–	
FF42H	Timer clock selection register 2	TCL2			–	√	–	
FF48H	16-bit timer mode control register 90	TMC90			√	√	–	
FF49H	Buzzer output control register 90	BZC90			√	√	–	
FF4AH	Watch timer mode control register	WTM			√	√	–	
FF50H	8-bit compare register 80	CR80		W	–	√	–	Undefined
FF51H	8-bit timer counter 80	TM80		R	–	√	–	00H
FF53H	8-bit timer mode control register 80	TMC80		R/W	√	√	–	

Notes 1. 16-bit access is allowed only with short direct addressing.

2. MUL0, CR90, TM90, and TCP90 are designed only for 16-bit access. With direct addressing, however, they can also be accessed in 8-bit mode.

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Bit Units for Manipulation			After Reset
					1 Bit	8 Bits	16 Bits	
FF54H	8-bit compare register 81	CR81		W	–	√	–	Undefined
FF55H	8-bit timer counter 81	TM81		R	–	√	–	00H
FF57H	8-bit timer mode control register 81	TMC81		R/W	√	√	–	
FF58H	8-bit compare register 82	CR82		W	–	√	–	Undefined
FF59H	8-bit timer counter 82	TM82		R	–	√	–	00H
FF5BH	8-bit timer mode control register 82	TMC82		R/W	√	√	–	
FF70H	Asynchronous serial interface mode register 20	ASIM20			√	√	–	
FF71H	Asynchronous serial interface status register 20	ASIS20		R	√	√	–	
FF72H	Serial operation mode register 20	CSIM20		R/W	√	√	–	
FF73H	Baud rate generator control register 20	BRGC20			–	√	–	
FF74H	Transmission shift register 20	TXS20	SIO20	W	–	√	–	FFH
	Reception buffer register 20	RXB20		R	–	√	–	Undefined
FF80H	A/D converter mode register 0	ADM0			√	√	–	00H
FF84H	A/D input selection register 0	ADS0			√	√	–	
FFD0H	Multiplication data register A0	MRA0		W	√	√	–	Undefined
FFD1H	Multiplication data register B0	MRB0			√	√	–	
FFD2H	Multiplier control register 0	MULC0		R/W	√	√	–	00H
FFE0H	Interrupt request flag register 0	IF0			√	√	–	
FFE1H	Interrupt request flag register 1	IF1			√	√	–	
FFE4H	Interrupt mask flag register 0	MK0			√	√	–	FFH
FFE5H	Interrupt mask flag register 1	MK1			√	√	–	
FFECH	External interrupt mode register 0	INTM0			–	√	–	00H
FFEDH	External interrupt mode register 1	INTM1			–	√	–	
FFF7H	Pull-up resistor option register 0	PU0			√	√	–	
FFF9H	Watchdog timer mode register	WDTM			√	√	–	
FFFAH	Oscillation stabilization time selection register	OSTS			–	√	–	04H
FFFBH	Processor clock control register	PCC			√	√	–	02H

Note

3.3 Instruction Address Addressing

An instruction address is determined by the program counter (PC) contents. The PC 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 to branch by the following addressing (For details of each instruction, refer to **78K/0S Series User's Manual — Instruction (U11047E)**).

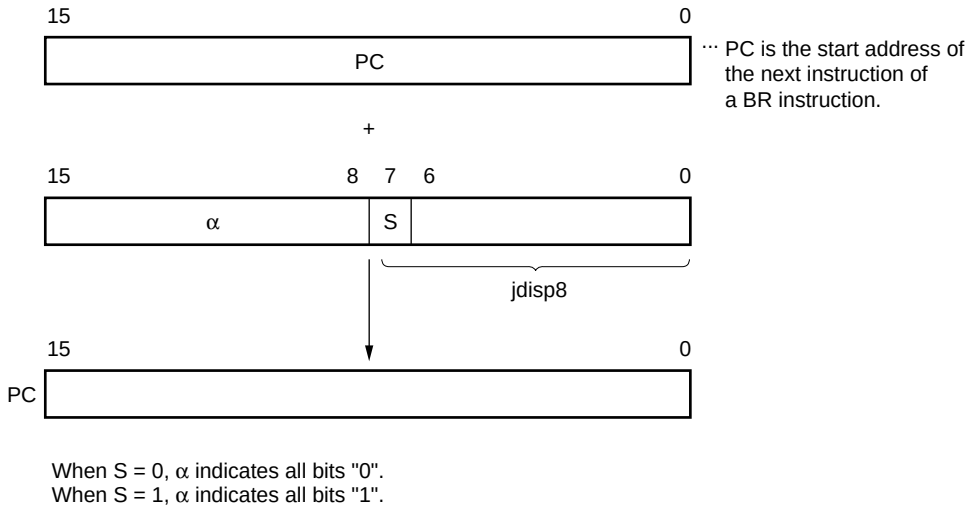
3.3.1 Relative addressing

[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) to branch. The displacement value is treated as signed two's complement data (–128 to +127) and bit 7 becomes a sign bit. In other words, the range of branch in relative addressing is between –128 and +127 of the start address of the following instruction.

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

[Illustration]



When S = 0, α indicates all bits "0".
When S = 1, α indicates all bits "1".

3.3.2 Immediate addressing

[Function]

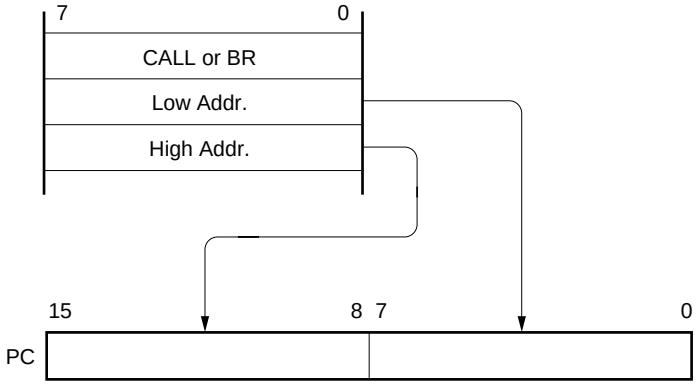
Immediate data in the instruction word is transferred to the program counter (PC) to branch.

This function is carried out when the CALL !addr16 and BR !addr16 instructions are executed.

CALL !addr16 and BR !addr16 instructions can be used to branch to all the memory spaces.

[Illustration]

In case of CALL !addr16 and BR !addr16 instructions



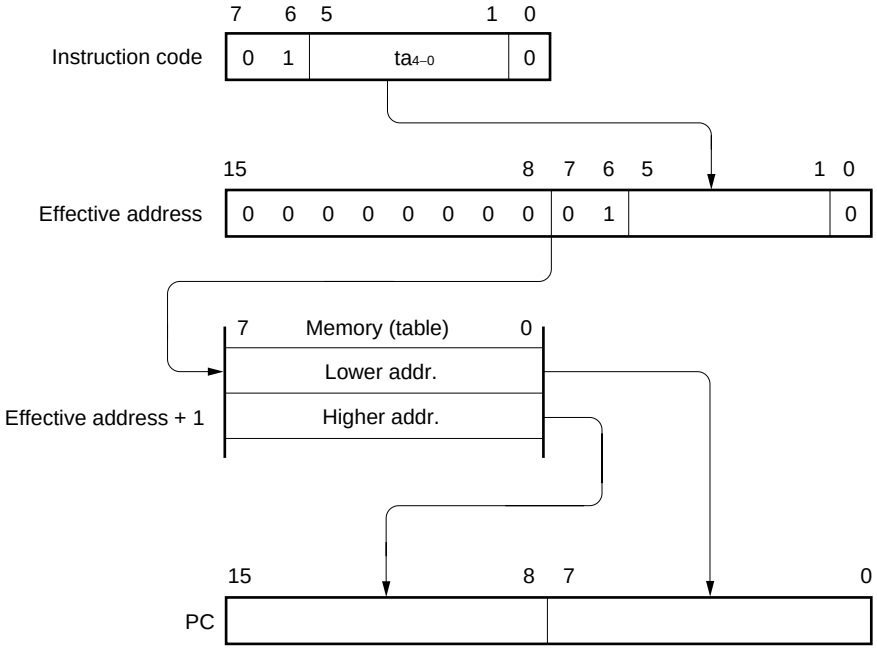
3.3.3 Table indirect addressing

[Function]

Table contents (branch destination address) of the particular location to be addressed by the immediate data of an instruction code from bit 1 to bit 5 are transferred to the program counter (PC) to branch.

Table indirect addressing is carried out when the CALLT [addr5] instruction is executed. This instruction can be used to branch to all the memory spaces according to the address stored in the memory table 40H to 7FH.

[Illustration]



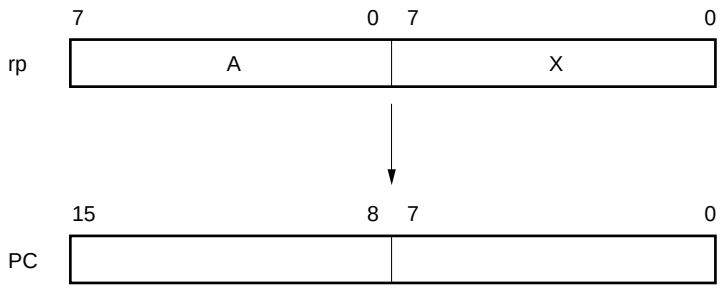
3.3.4 Register addressing

[Function]

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

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) which undergo manipulation during instruction execution.

3.4.1 Direct addressing

[Function]

The memory indicated by immediate data in an instruction word is directly addressed.

[Operand format]

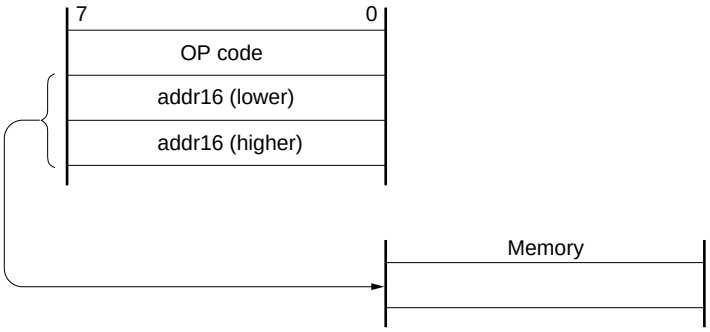
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

Instruction code	0	0	1	0	1	0	0	1	OP code
	0	0	0	0	0	0	0	0	00H
	1	1	1	1	1	1	1	0	FEH

[Illustration]



3.4.2 Short direct addressing

[Function]

The memory to be manipulated in the fixed space is directly addressed with 8-bit data in an instruction word.

The fixed space where this addressing is applied to is the 256-byte space FE20H to FF1FH. An internal high-speed RAM and a special function register (SFR) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.

The SFR area (FF00H to FF1FH) where short direct addressing is applied is a part of all SFR areas. In this area, ports which are frequently accessed in a program and a compare register of the timer 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. See [Illustration] below.

[Operand format]

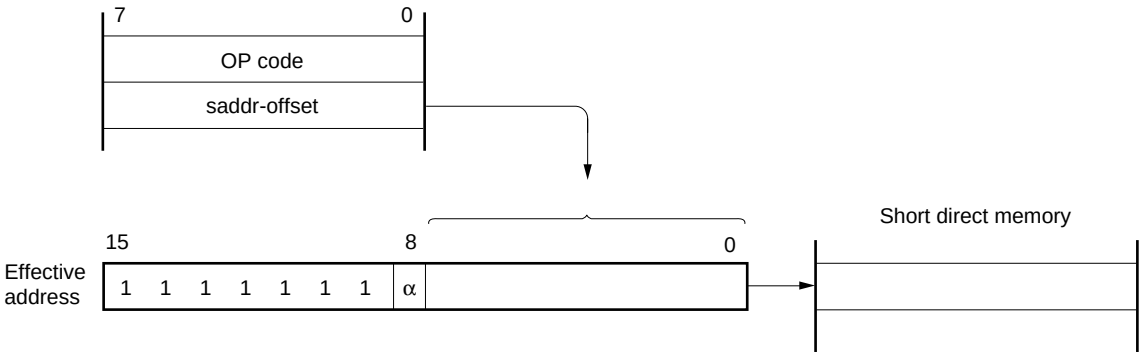
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

[Description example]

MOV FE90H, #50H; When setting saddr to FE90H and the immediate data to 50H

Instruction code	1	1	1	1	0	1	0	1	OP code
	1	0	0	1	0	0	0	0	90H (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, $\alpha = 1$.

3.4.3 Special function register (SFR) addressing

[Function]

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

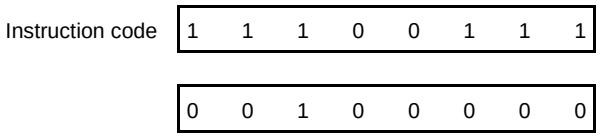
This addressing is applied to the 256-byte space FF00H to FFFFH. However, the SFR mapped at FF00H to FF1FH can also be accessed with short direct addressing.

[Operand format]

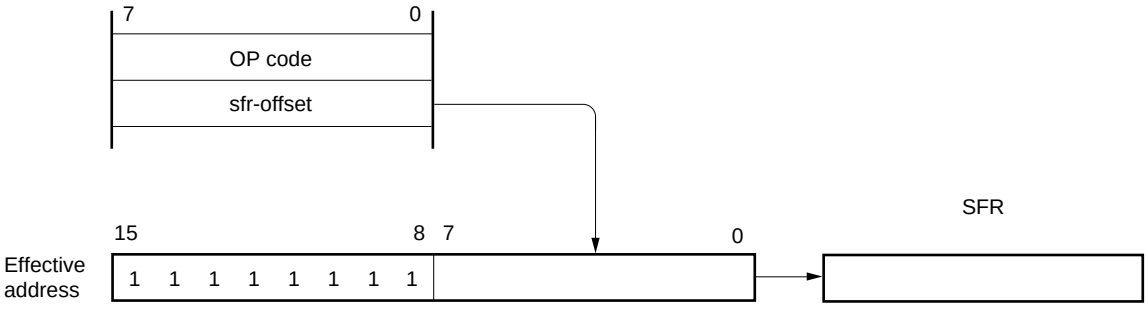
Identifier	Description
sfr	Special function register name

[Description example]

MOV PM0, A; When selecting PM0 for sfr



[Illustration]



3.4.4 Register addressing

[Function]

The general-purpose registers are accessed as operands. The general-purpose register to be accessed is specified with register specify code and functional name in the instruction code.

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

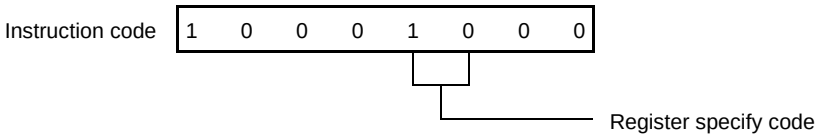
[Operand format]

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

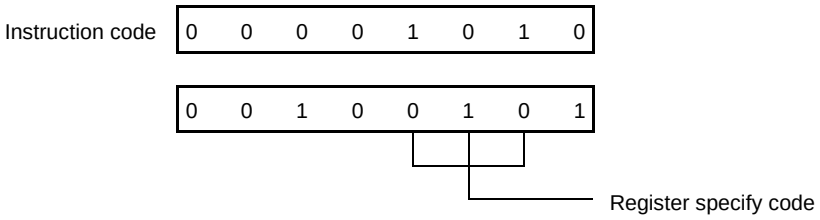
'r' and 'rp' can be described with absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

MOV A, C; When selecting the C register for r



INCW DE; When selecting the DE register pair for rp



3.4.5 Register indirect addressing

[Function]

The memory is addressed with the contents of the register pair specified as an operand. The register pair to be accessed is specified with the register pair specify code in the instruction code. This addressing can be carried out for all the memory spaces.

[Operand format]

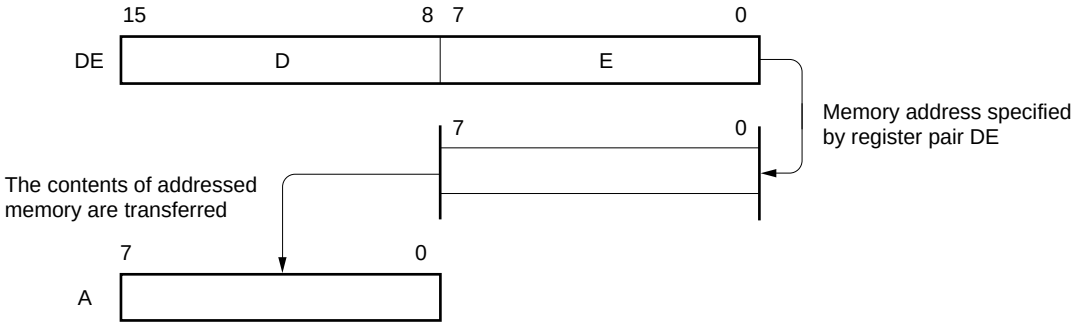
Identifier	Description
–	[DE], [HL]

[Description example]

MOV A, [DE]; When selecting register pair [DE]

Instruction code	0	0	1	0	1	0	1	1

[Illustration]



3.4.6 Based addressing

[Function]

8-bit immediate data is added to the contents of the base register, that is, the HL register pair, and the sum is used to address the memory. Addition is performed by expanding the offset data as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all the memory spaces.

[Operand format]

Identifier	Description
–	[HL+byte]

[Description example]

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

Instruction code	0	0	1	0	1	1	0	1
	0	0	0	1	0	0	0	0

3.4.7 Stack addressing

[Function]

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

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

Stack addressing can be used to access the internal high-speed RAM area only.

[Description example]

In the case of PUSH DE

Instruction code	1	0	1	0	1	0	1	0

[MEMO]

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

Table 4-1. Port Functions

Pin Name	I/O	Function	After Reset	Alternate Function
P00 to P04	I/O	Port 0 5-bit input/output port Input/output mode can be specified in 1-bit units When used as an input port, an on-chip pull-up resistor can be specified by means of pull-up resistor option register 0 (PU0).	Input	–
P10, P11	I/O	Port 1 2-bit input/output port Input/output mode can be specified in 1-bit units When used as an input port, an on-chip pull-up resistor can be specified by means of pull-up resistor option register 0 (PU0).	Input	–
P20	I/O	Port 2 7-bit input/output port Input/output mode can be specified in 1-bit units For P20 to P22, P25, and P26, an on-chip pull-up resistor can be specified by means of pull-up resistor option register B2 (PUB2). Only P23 and P24 can be used as N-ch open-drain input/output port pins.	Input	$\overline{\text{SCK20}}$/ASCK20
P21				SO20/TxD20
P22				SI20/RxD20
P23				-
P24				-
P25				TI80/$\overline{\text{SS20}}$
P26				TO80
P30	I/O	Port 3 4-bit input/output port Input/output mode can be specified in 1-bit units An on-chip pull-up resistor can be specified by means of pull-up resistor option register B3 (PUB3).	Input	INTP0/TI81/CPT90
P31				INTP1/TO81
P32				INTP2/TO90
P33				INTP3/TO82/BZO90
P50 to P53	I/O	Port 5 4-bit N-ch open-drain input/output port Input/output mode can be specified in 1-bit units For a mask ROM version, an on-chip pull-up resistor can be specified by a mask option.	Input	–
P60 to P63	Input	Port 4 4-bit input-only port	Input	ANI0 to ANI3

4.2 Port Configuration

Ports have the following hardware configuration.

Table 4-2. Configuration of Port

Parameter	Configuration
Control register	Port mode registers (PMm: m = 0 to 3, 5) Pull-up resistor option register 0 (PU0) Pull-up resistor option registers B2, B3 (PUB2, PUB3)
Port	Total: 26 (CMOS input/output: 16, CMOS input: 4, N-ch open-drain input/output: 6)
Pull-up resistor	• Mask ROM versions Total: 20 (software control: 16, mask option control: 4) • Flash memory versions Total: 16 (software control only)

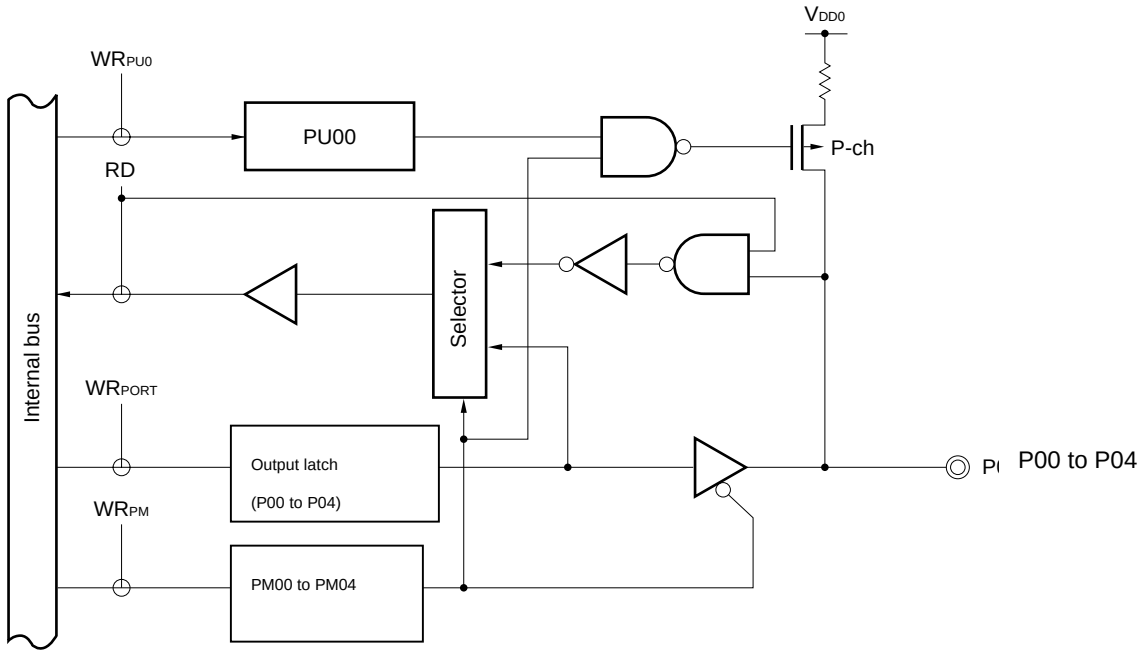
4.2.1 Port 0

This is a 5-bit I/O port with output latches. Port 0 can be set to input or output mode in 1-bit units by using port mode register 0 (PM0). When pins P00 to P04 are used as input port pins, on-chip pull-up resistors can be connected in 5-bit units by using pull-up resistor option register 0 (PU0).

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

Figure 6-2 shows a block diagram of port 0.

Figure 4-2. Block Diagram of P00 to P05



PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 0 read signal
WR: Port 0 write signal

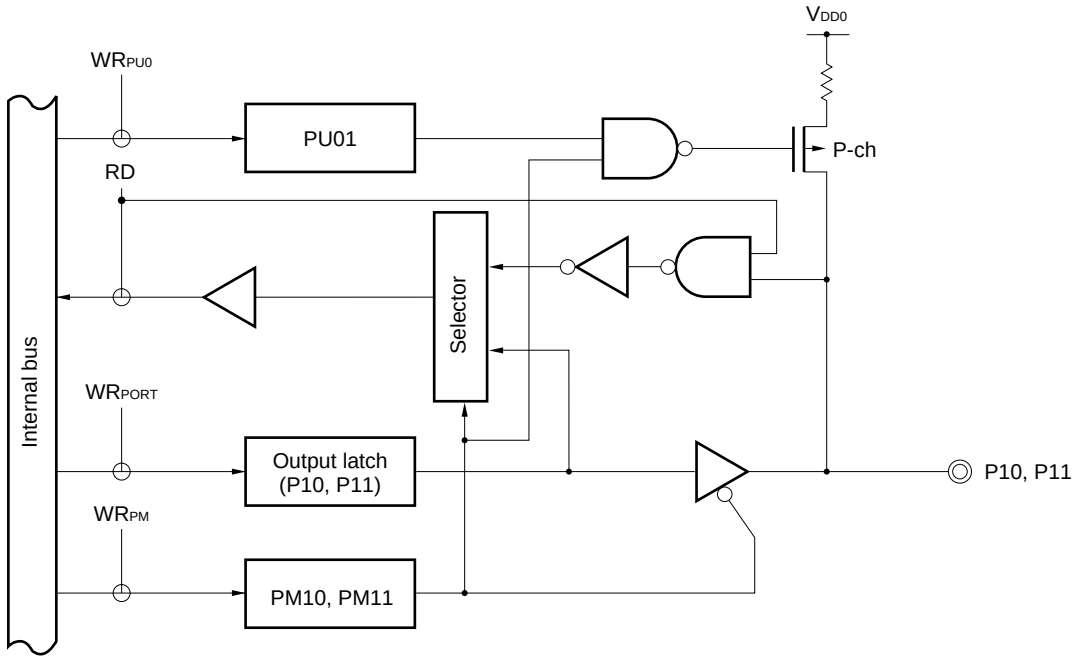
4.2.2 Port 1

This is a 2-bit I/O port with output latches. Port 1 can be set to input or output mode in 1-bit units by using the port mode register 1 (PM1). When the P10 and P11 pins are used as input port pins, on-chip pull-up resistors can be connected in 2-bit units by using pull-up resistor option register 0 (PU0).

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

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

Figure 4-3. Block Diagram of P10 and P11



PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 1 read signal
WR: Port 1 write signal

4.2.3 Port 2

This is a 7-bit I/O port with output latches. Port 2 can be set to input or output mode in 1-bit units by using port mode register 2 (PM2). For pins P20 to P22, P25, and P26, on-chip pull-up resistors can be connected in 1-bit units by using pull-up resistor option register B2 (PUB2).

The port is also used as a data I/O and clock I/O to and from the serial interface, and timer I/O.

RESET input sets port 2 to input mode.

Figures 4-4 through 6-8 show block diagrams of port 2.

Caution When using the pins of port 2 as the serial interface, the I/O and output latches must be set according to the function to be used. For details of the settings, see Table 14-2 Serial Interface 20 Operating Mode Settings.

Figure 4-4. Block Diagram of P20

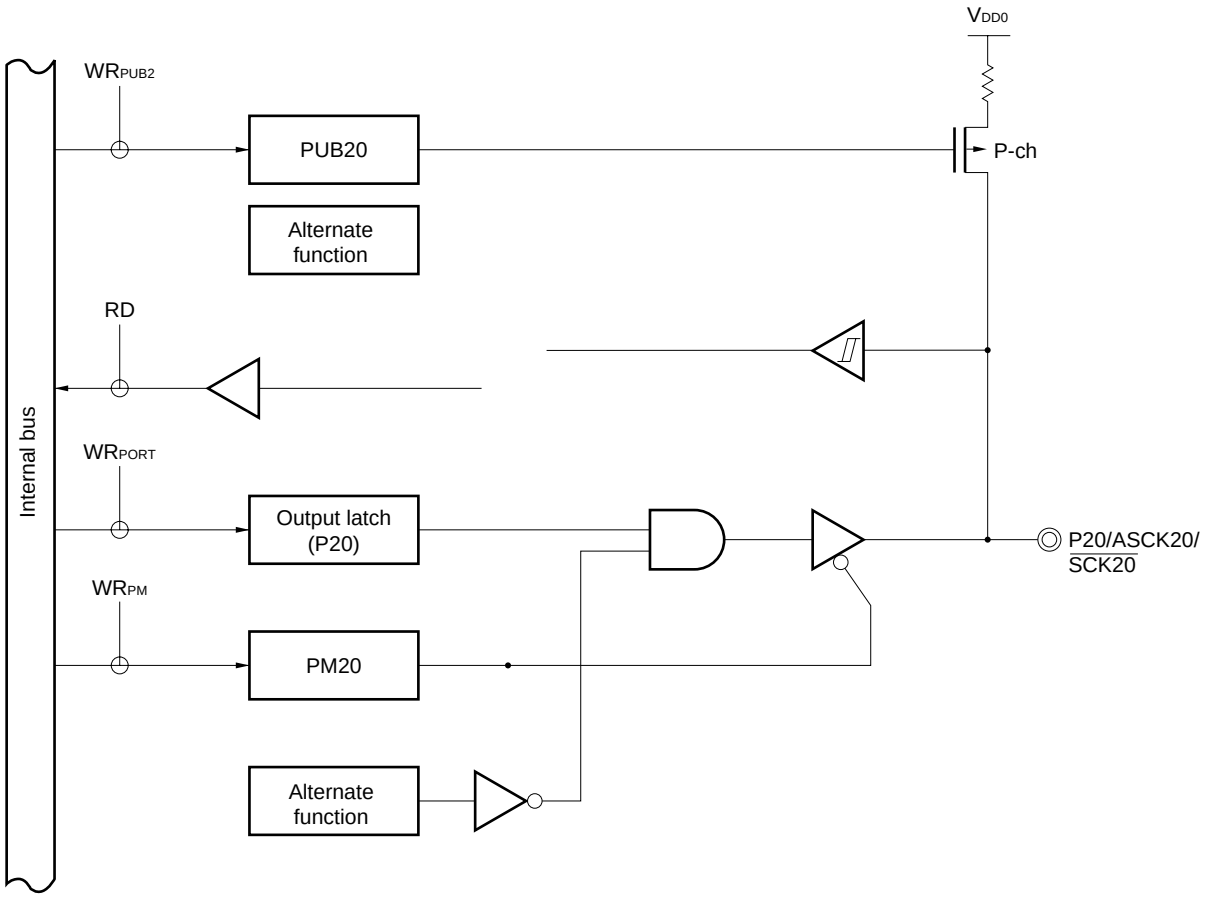
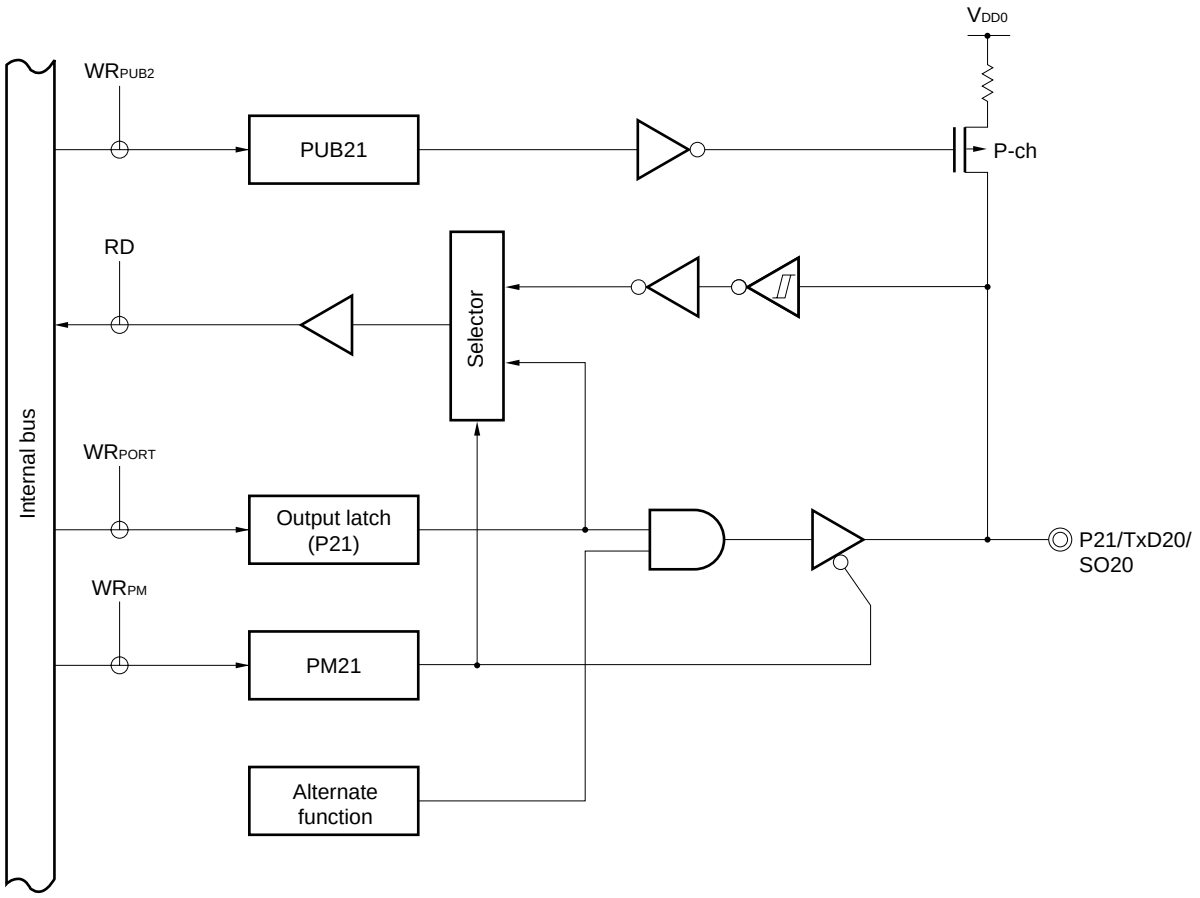


Figure 4-5. Block Diagram of P21



PUB2: Pull-up resistor option register B2
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

Figure 4-6. Block Diagram of P22 and P25

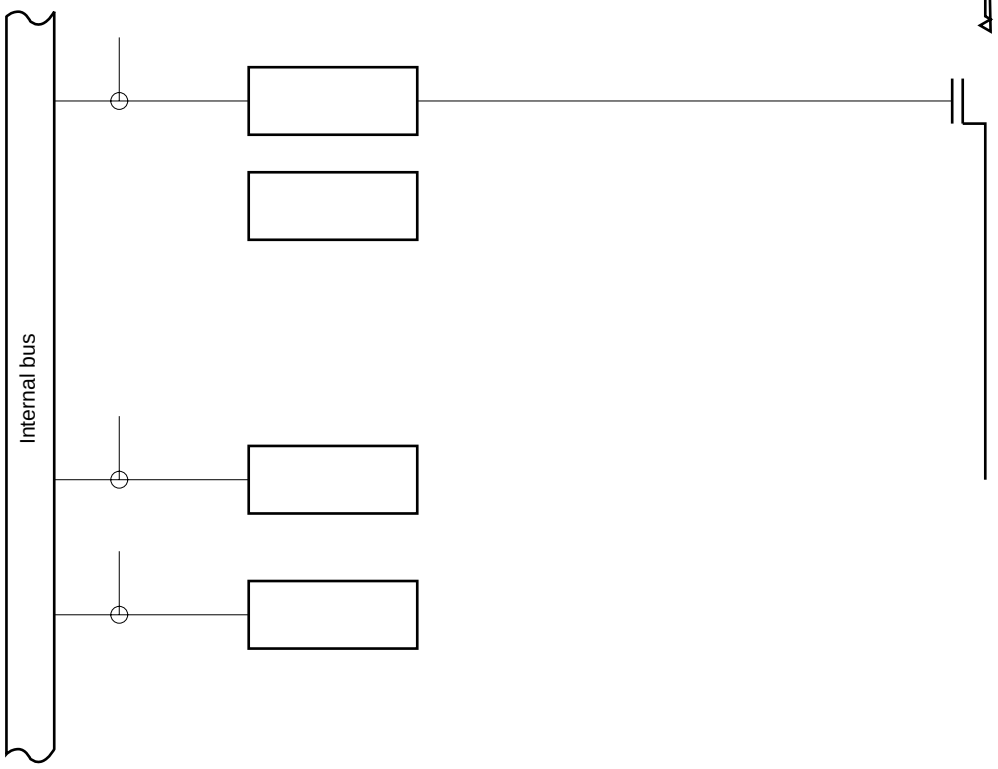
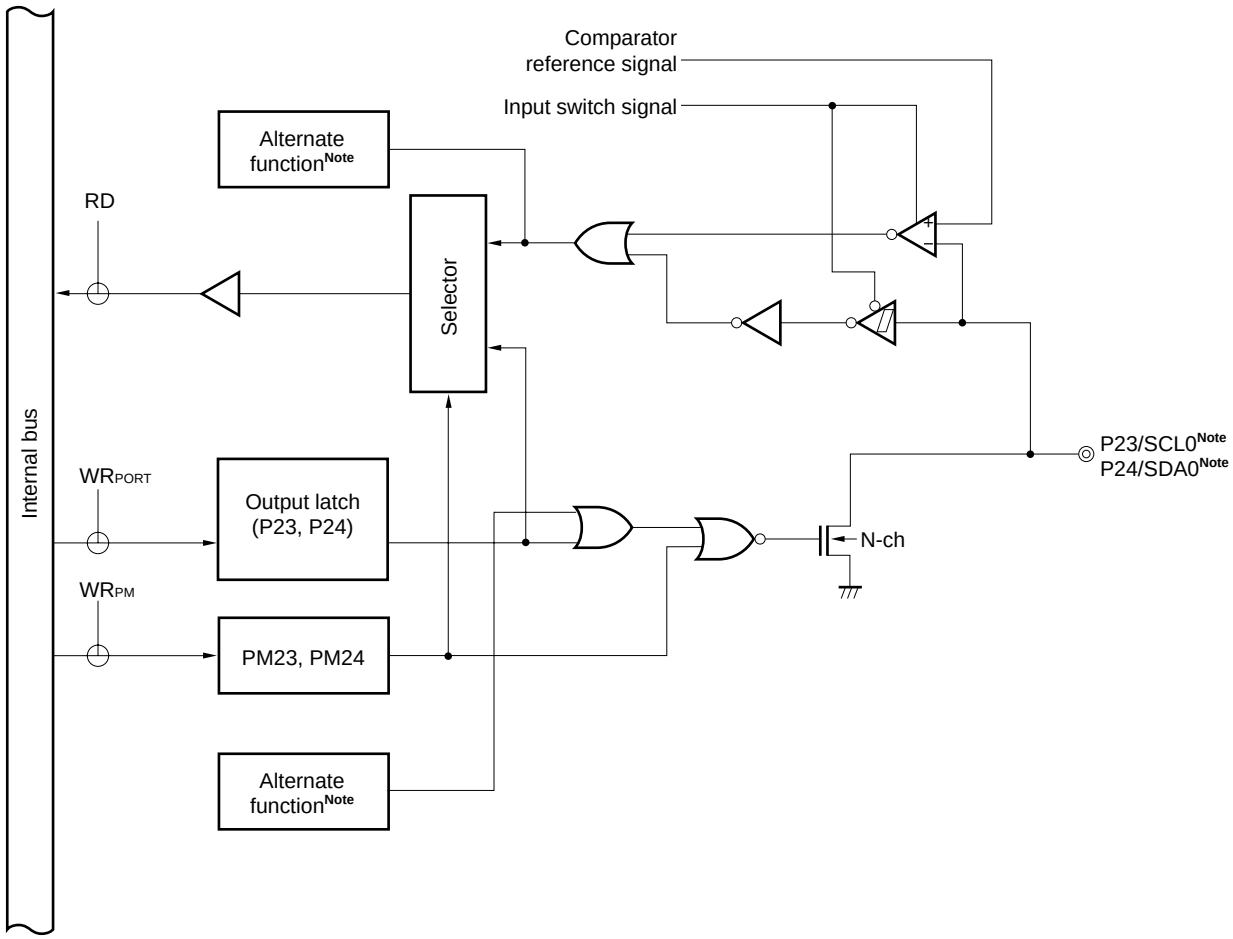
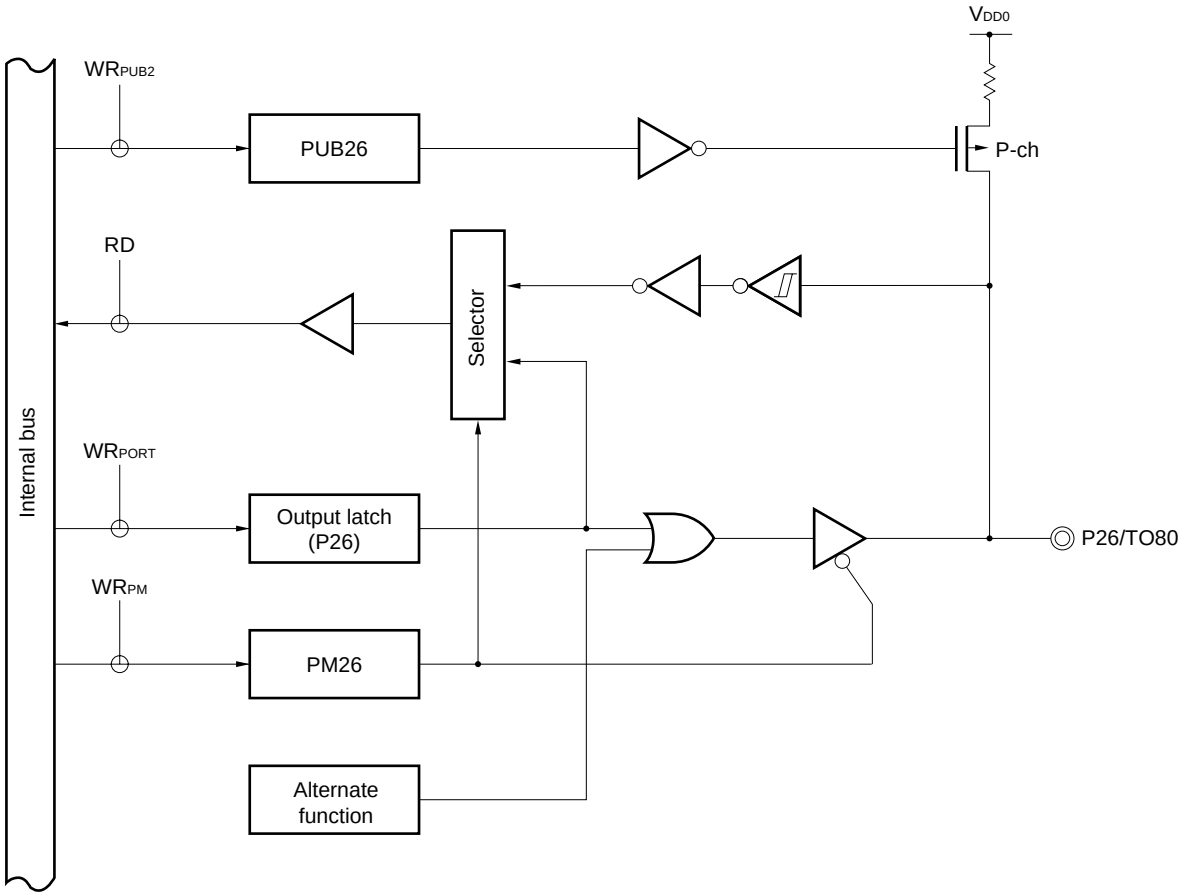


Figure 4-7. Block Diagram of P23 and P24



PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

Figure 4-8. Block Diagram of P26



PUB2: Pull-up resistor option register B2
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

4.2.4 Port 3

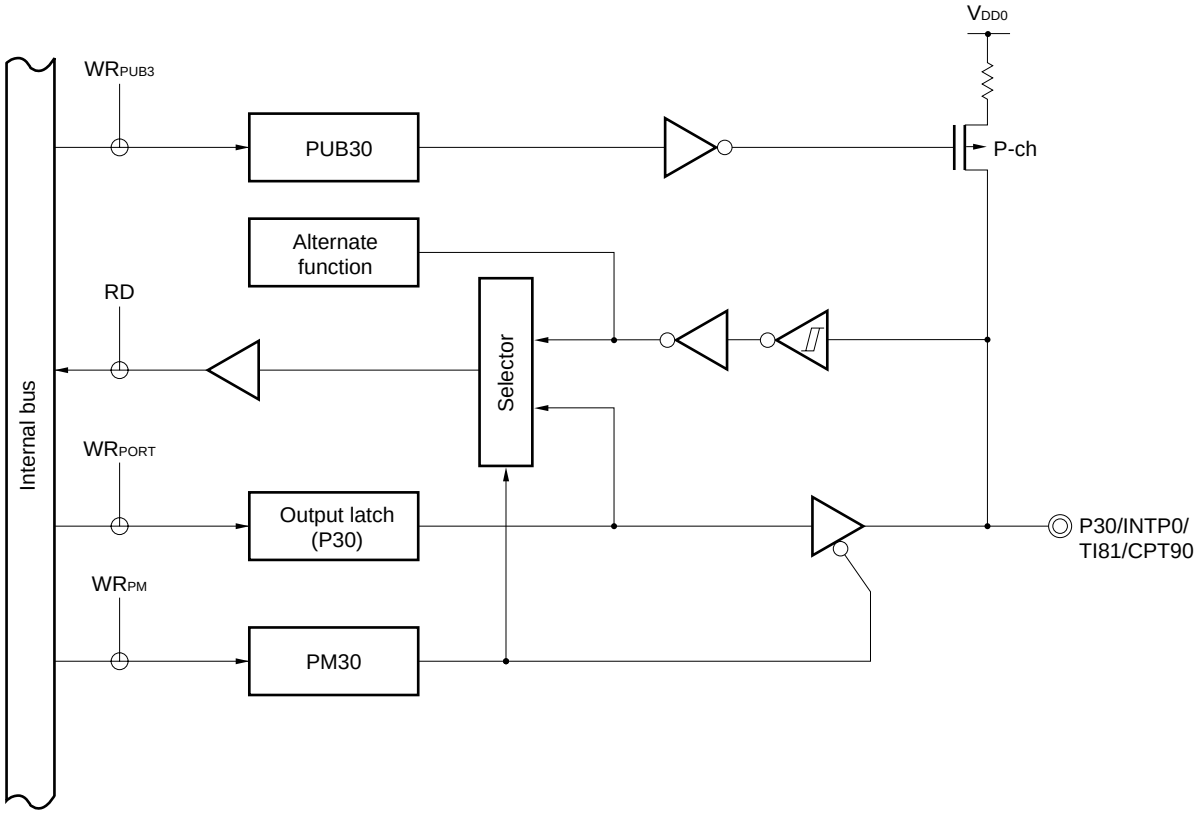
This is a 4-bit I/O port with output latches. Port 3 can be set to input or output mode in 1-bit units by using port mode register 3 (PM3). For pins P30 to P33, on-chip pull-up resistors can be connected in 1-bit units by using pull-up resistor option register B3 (PUB3).

The port is also used as an external interrupt input, capture input, timer output, and buzzer output.

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

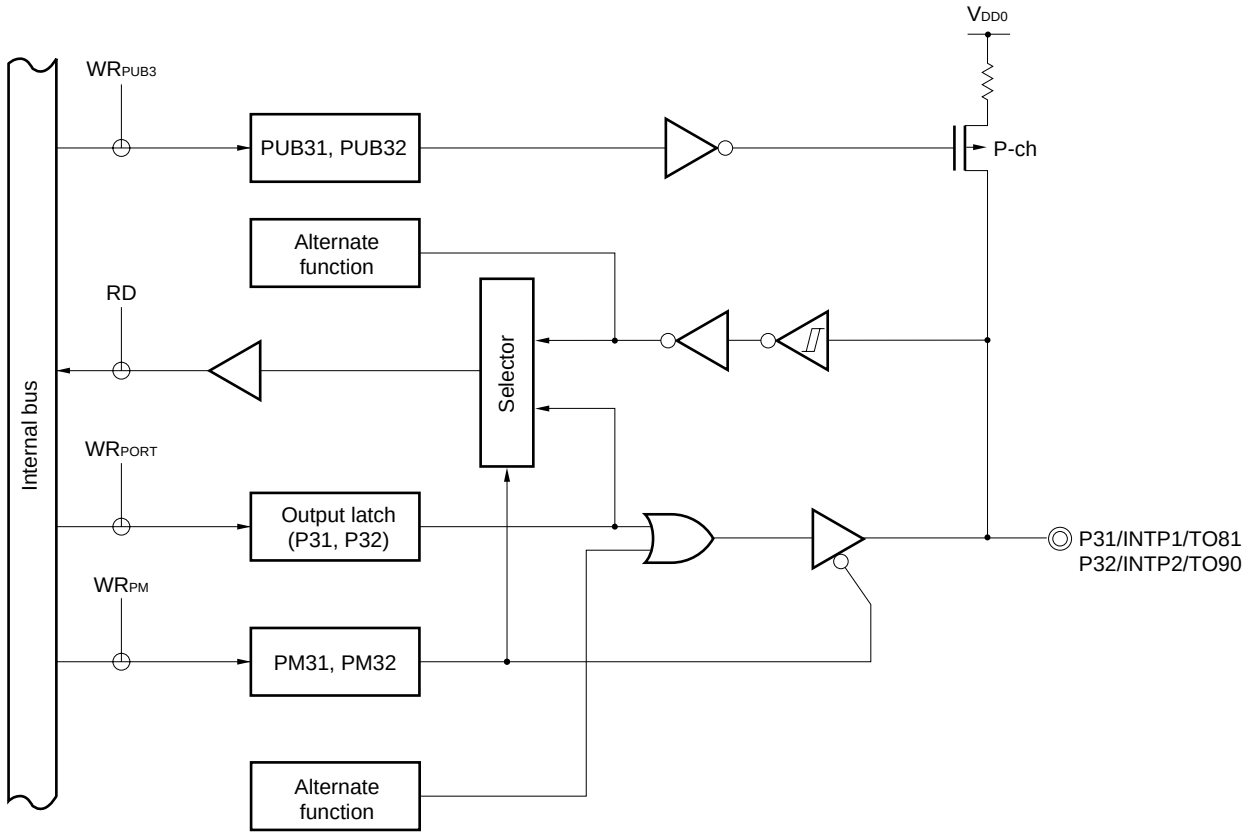
Figures 6-9 through 6-11 show block diagrams of port 3.

Figure 4-9. Block Diagram of P30



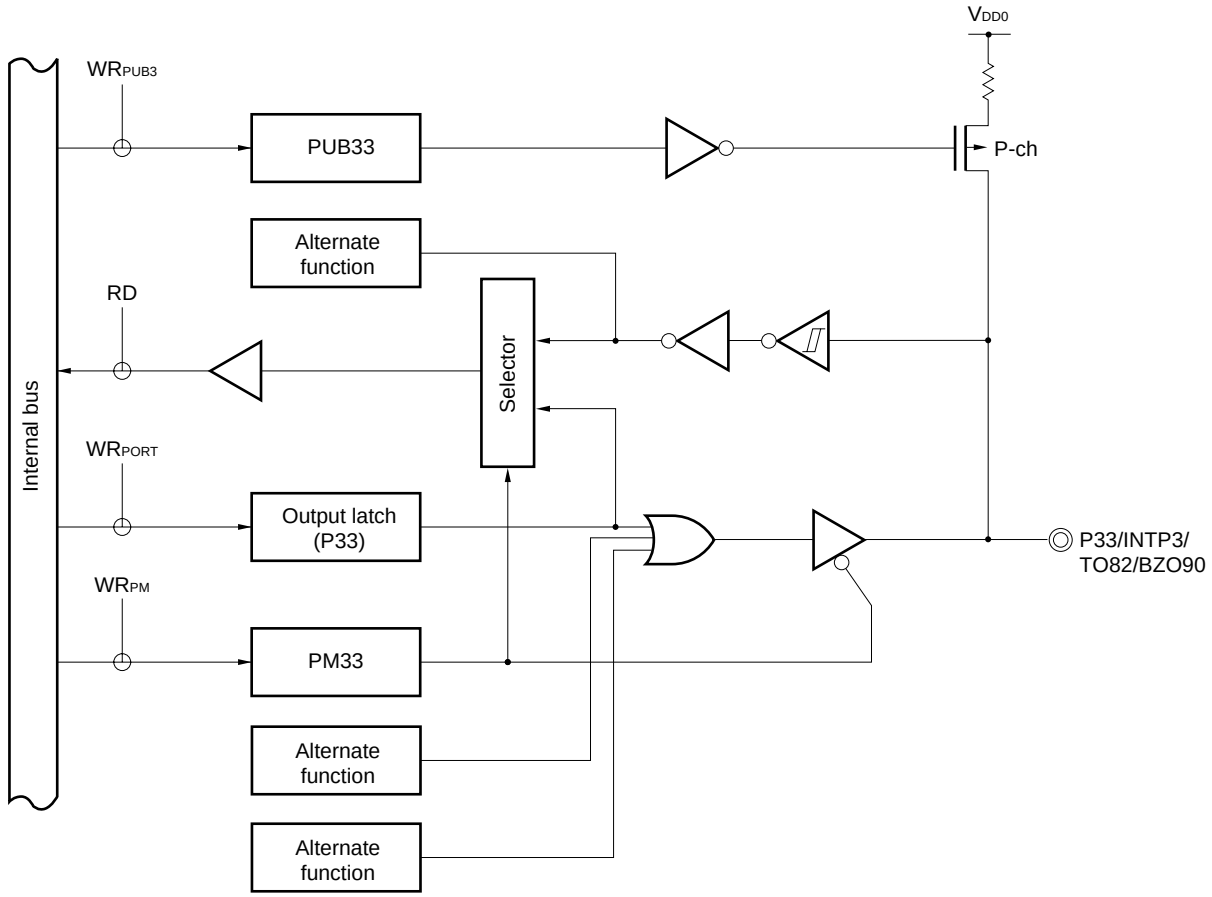
PUB3: Pull-up resistor option register B3
PM: Port mode register
RD: Port 3 read signal
WR: Port 3 write signal

Figure 4-10. Block Diagram of P31 and P32



PUB3: Pull-up resistor option register B3
PM: Port mode register
RD: Port 3 read signal
WR: Port 3 write signal

Figure 4-11. Block Diagram of P33



PUB3: Pull-up resistor option register B3
PM: Port mode register
RD: Port 3 read signal
WR: Port 3 write signal

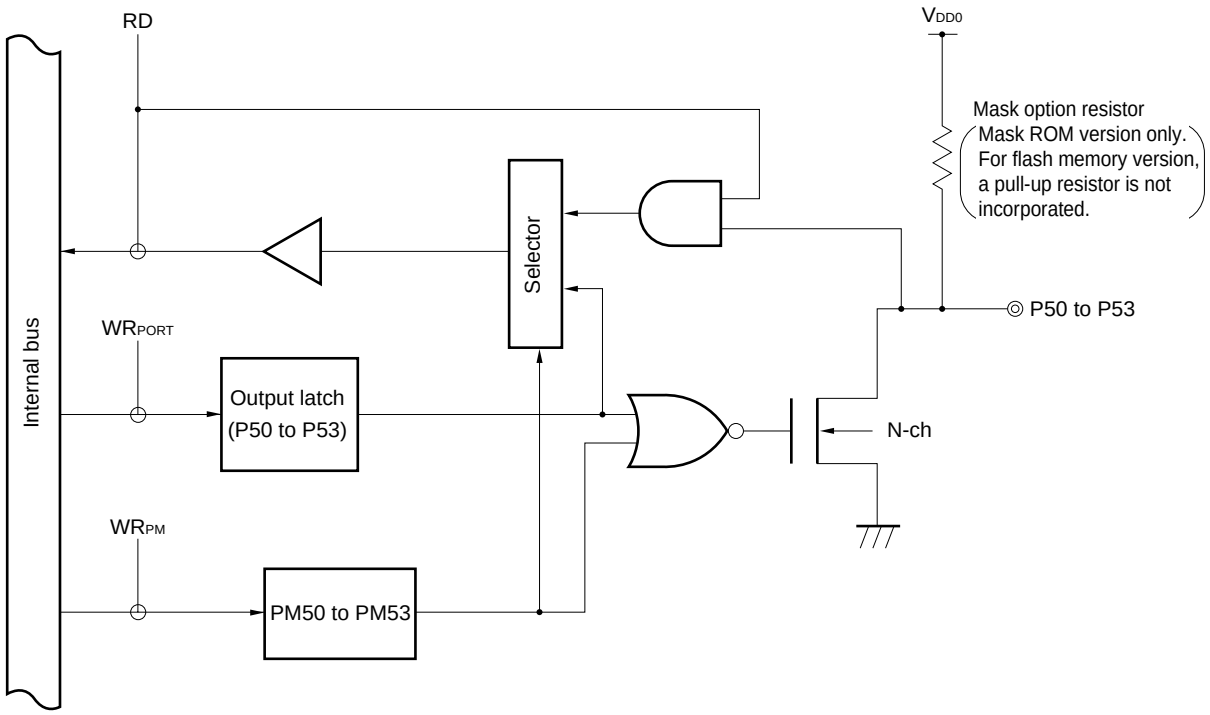
4.2.5 Port 5

This is a 4-bit N-ch open-drain I/O port with output latches. Port 5 can be set to input or output mode in 1-bit units by using port mode register 5 (PM5). For a mask ROM version, whether a pull-up resistor is to be incorporated can be specified by the mask option.

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

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

Figure 4-12. Block Diagram of P50 to P53



PM: Port mode register

RD: Port 5 read signal

WR: Port 5 write signal

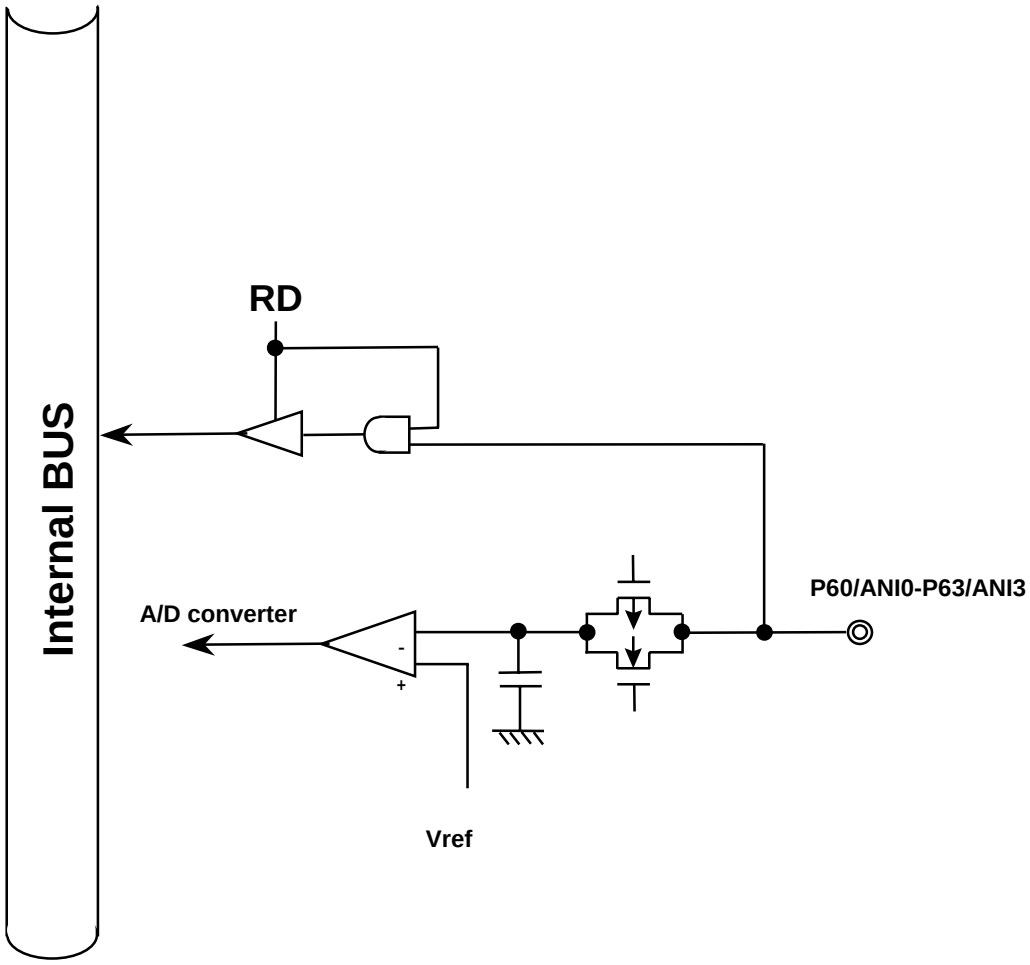
4.2.6 Port 6

This is an 4-bit input port.

The port is also used as an analog input to the A/D converter.

Figure 4-13 shows a block diagram of port 6.

Figure 4-13. Block Diagram of P60 to P63



4.3 Port Function Control Registers

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

- Port mode registers (PM0 to PM3, and PM5)
- Pull-up resistor option registers (PU0, PUB2, and PUB3)

(1) Port mode registers (PM0 to PM3, and PM5)

The port mode registers separately set each port bit to either input or output.

Each port mode register is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input writes FFH into the port mode registers.

When port pins are used for alternate functions, the corresponding port mode register and output latch must be set or reset as described in Table 6-3.

Caution When port 3 is acting as an output port, and its output level is changed, an interrupt request flag is set, because this port is also used as the input for an external interrupt. To use port 3 in output mode, therefore, the interrupt mask flag must be set to 1 in advance.

Figure 4-14. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
PM0	1	1	1	PM04	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
PM1	1	1	1	1	1	1	PM11	PM10	FF21H	FFH	R/W
PM2	1	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM5	1	1	1	1	PM53	PM52	PM51	PM50	FF25H	FFH	R/W

PMmn	Pmn pin input/output mode selection
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

Table 4-3. Port Mode Register and Output Latch Settings for Using Alternate Functions

Pin Name	Alternate Function		PM××	P××
	Name	Input/Output		
P25	TI80	Input	1	×
P26	TO80	Output	0	0
P30	INTP0	Input	1	×
	TI81	Input	1	×
	CPT90	Input	1	×
P31	INTP1	Input	1	×
	TO81	Output	0	0
P32	INTP2	Input	1	×
	TO90	Output	0	0
P33	INTP3	Input	1	×
	TO82	Output	0	0
	BZO90	Output	0	0
P60 to P63	ANI0 to ANI3	Input	1	×

Caution When using the pins of port 2 as the serial interface, the I/O or output latch must be set according to the function to be used. For details of the settings, see Table 14-2 Serial Interface 20 Operating Mode Settings.

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

(2) Pull-up resistor option register 0 (PU0)

The pull-up resistor option register (PU0) sets whether an on-chip pull-up resistor on each port is used. On the port which is specified to use the on-chip pull-up resistor in PU0, the pull-up resistor can be internally used only for the bits set to input mode. No on-chip pull-up resistors can be used for the bits set to output mode regardless of the setting of PU0. On-chip pull-up resistors cannot be used even when the pins are used as the alternate-function output pins.

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

$\overline{\text{RESET}}$ input clears PU0 to 00H.

Figure 4-15. Format of Pull-Up Resistor Option Register 0

Symbol	7	6	5	4	3	2	<1>	<0>	Address	After reset	R/W
PU0	0	0	0	0	0	0	PU01	PU00	FFF7H	00H	R/W

PU0m	Pm on-chip pull-up resistor selection (m = 0, 1)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

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

(3) Pull-up resistor option registers B2 and B3 (PUB2 and PUB3)

These registers specify whether an on-chip pull-up resistor is connected to each pin of ports 2 and 3. The pin specified by PUB2 or PUB3 is connected to on-chip pull-up resistor regardless of the setting of the port mode register.

PUB2 and PUB3 are set with a 1-bit or 8-bit manipulation instruction.

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

Figure 4-16. Format of Pull-Up Resistor Option Register B2

Symbol	7	<6>	<5>	4	3	<2>	<1>	<0>	Address	After reset	R/W
PUB2	0	PUB26	PUB25	0	0	PUB22	PUB21	PUB20	FF32H	00H	R/W

PUB2n	P2n on-chip pull-up resistor selection (n = 0 to 2, 5, 6)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

Caution Bits 3, 4, and 7 must all be set to 0.

Figure 4-17. Format of Pull-Up Resistor Option Register B3

Symbol	7	6	5	4	<3>	<2>	<1>	<0>	Address	After reset	R/W
PUB3	0	0	0	0	PUB33	PUB32	PUB31	PUB30	FF33H	00H	R/W

PUB3n	P3n on-chip pull-up resistor selection (n = 0 to 3)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

Caution Bits 4 to 7 must all be set to 0.

4.4 Operation of Port Functions

The operation of a port differs depending on whether the port is set to input or output mode, as described below.

4.4.1 Writing to I/O port

(1) In output mode

A value can be written to the output latch of a port by using a transfer instruction. The contents of the output latch can be output from the pins of the port.

The data once written to the output latch is retained until new data is written to the output latch.

(2) In input mode

A value can be written to the output latch by using a transfer instruction. However, the status of the port pin is not changed because the output buffer is OFF.

The data once written to the output latch is retained until new data is written to the output latch.

Caution A 1-bit memory manipulation instruction is executed to manipulate 1 bit of a port. However, this instruction accesses the port in 8-bit units. When this instruction is executed to manipulate a bit of a port consisting both of inputs and outputs, therefore, the contents of the output latch of the pin that is set to input mode and not subject to manipulation become undefined.

4.4.2 Reading from I/O port

(1) In output mode

The contents of the output latch can be read by using a transfer instruction. The contents of the output latch are not changed.

(2) In input mode

The status of a pin can be read by using a transfer instruction. The contents of the output latch are not changed.

4.4.3 Arithmetic operation of I/O port

(1) In output mode

An arithmetic operation can be performed with the contents of the output latch. The result of the operation is written to the output latch. The contents of the output latch are output from the port pins.

The data once written to the output latch is retained until new data is written to the output latch.

(2) In input mode

The contents of the output latch become undefined. However, the status of the pin is not changed because the output buffer is OFF.

Caution A 1-bit memory manipulation instruction is executed to manipulate 1 bit of a port. However, this instruction accesses the port in 8-bit units. When this instruction is executed to manipulate a bit of a port consisting both of inputs and outputs, therefore, the contents of the output latch of the pin that is set to input mode and not subject to manipulation become undefined.

[MEMO]

CHAPTER 5 CLOCK GENERATION CIRCUIT

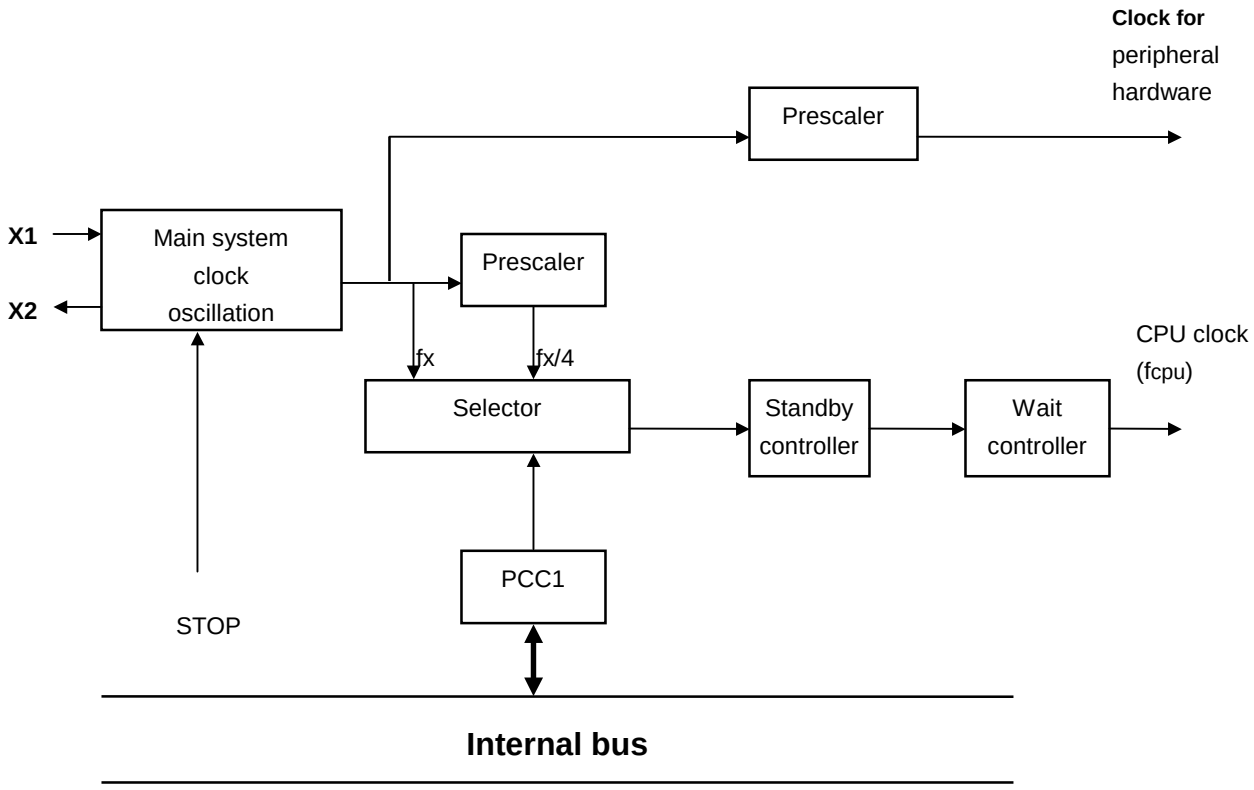
5.1 Clock Generation Circuit Functions

The clock generation circuit generates the clock to be supplied to the CPU and peripheral hardware. The following two types of system clock oscillators are used.

7(O)86(. X)625π 0.5733 –1.56 TΔ360.004 Tχ –0.0034 σεττι–0.δ τοπροχεσσιλΠΧ 1 χσ 0 0 04 T9467 1733 –1.56 Tμ 0 Tχ 0 Tω ( ]Tϑ /TT4 1 Tϕ 9.96 0 0 8.054

Figure 5-1. Block Diagram of Clock Generation Circuit

Figure 5-7. Clock Generator Block Diagram



5.3 Registers Controlling Clock Generation Circuit

The clock generation circuit is controlled by the following registers:

- Processor clock control register (PCC)

(1) Processor clock control register (PCC)

PCC selects the CPU clock and the ratio of division.

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

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

Figure 5-2. Format of Processor Clock Control Register

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
PCC	MCC	0	0	0	0	0	PCC1	0	FFFBH	02H	R/W

MCC	Control of main system clock oscillator operation
0	Operation enabled.
1	Operation disabled.

PCC1	CPU clock(fcpu) selection Note
0	fx(0.2 μ s)
1	Fx/2^2(0.8 μ s)

Note The CPU clock is selected according to a combination of the PCC1 flag in the processor clock control register (PCC) .

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

2. MCC can be set only when the subsystem clock has been selected as the CPU clock.

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

3. Minimum instruction execution time: 2 f_{CPU}

- f_{CPU} = 0.2 μs: 0.4 μs
- f_{CPU} = 0.8 μs: 1.6 μs

5.4 System Clock Oscillators

5.4.1 Main system clock oscillator

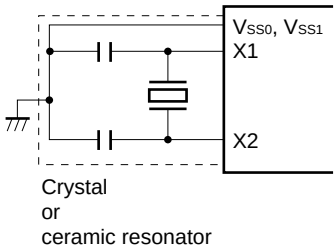
The main system clock oscillator is oscillated by the crystal or ceramic resonator (5.0 MHz TYP.) connected across the X1 and X2 pins.

An external clock can also be input to the circuit. In this case, input the clock signal to the X1 pin, and input the reversed signal to the X2 pin.

Figure 7-5 shows the external circuit of the main system clock oscillator.

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

(a) Crystal or ceramic oscillation



(b) External clock

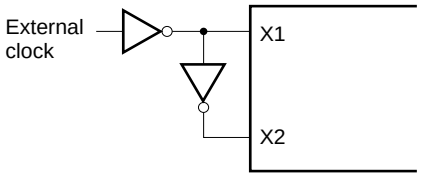
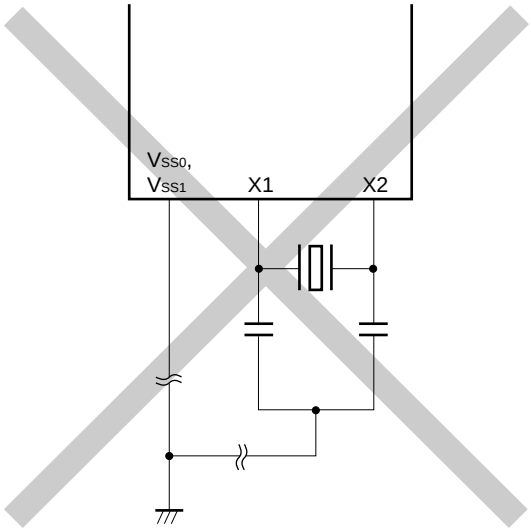
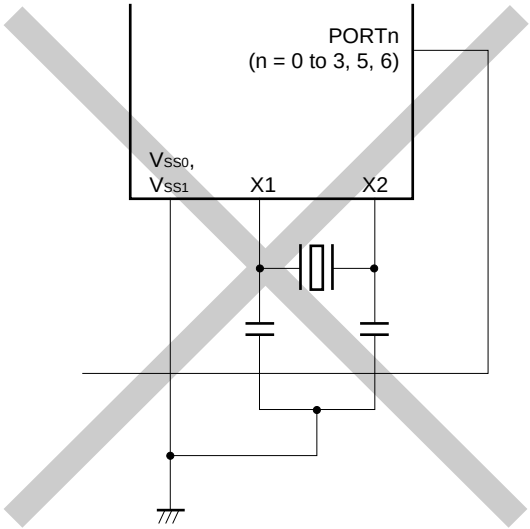


Figure 5-7. Examples of Incorrect Oscillator Connection (1/2)

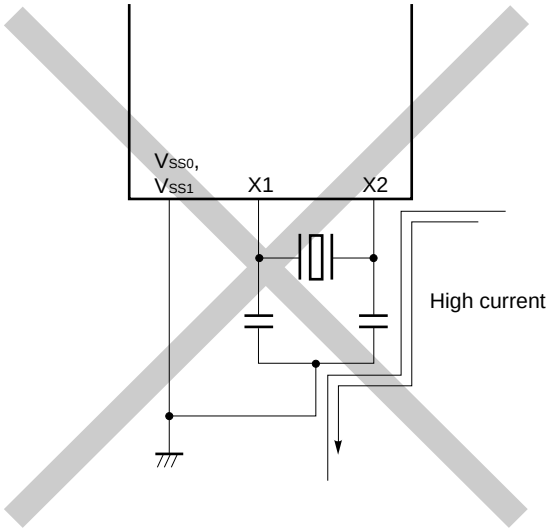
(a) Wiring too long



(b) Crossed signal line



(c) Wiring near high alternating current



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

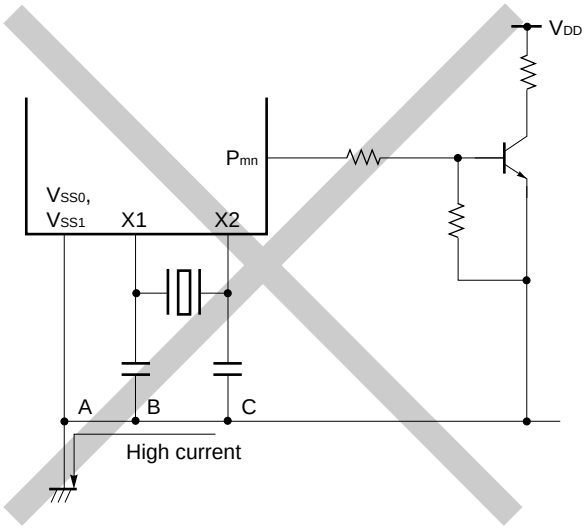
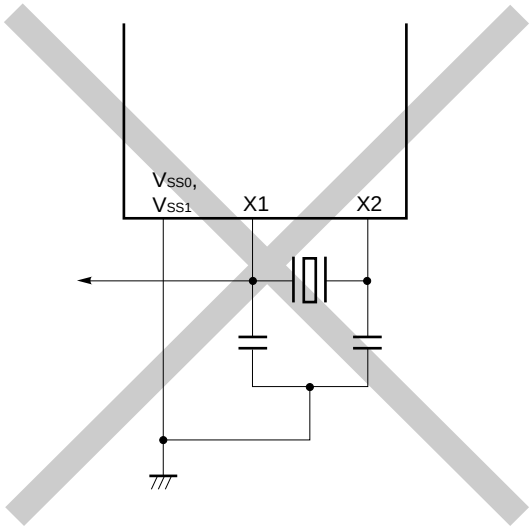


Figure 5-7. Examples of Incorrect Oscillator Connection (2/2)

(e) Signals are fetched



5.5 Clock Generation Circuit Operation

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

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

The operation of the clock generation circuit is determined by the processor clock control register (PCC), suboscillation mode register (SCKM), and subclock control register (CSS), as follows:

(a) The slow mode 2 f_{CPU} (1.6 μs: at 5.0-MHz operation) of the main system clock is selected when the $\overline{\text{RESET}}$ signal is generated (PCC = 02H). While a low level is input to the $\overline{\text{RESET}}$ pin, oscillation of the main system clock is stopped.

(b) Two types of CPU clocks f_{CPU} (0.2 μs and 0.8 μs: main system clock (at 5.0-MHz operation) can be selected by the PCC settings.

(c) Two standby modes, STOP and HALT, can be used with the main system clock selected.

5.6 Changing Setting of System Clock and CPU Clock

5.6.1 Time required for switching between system clock and CPU clock

The CPU clock can be selected by using bit 1 (PCC1) of the processor clock control register (PCC) and bit 4 (CSS0) of the subclock control register (CSS).

Actually, the specified clock is not selected immediately after the setting of PCC has been changed, and the old clock is used for the duration of several instructions after that (see Table 5-2).

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

Set Value before Switching	Set Value after Switching	
PCC1	PCC1	PCC1
	0	1
0	-	4 clocks
1	2 clocks	-

Remarks 1. Two clocks are the minimum instruction execution time of the CPU clock before switching.

2. The parenthesized values apply to operation at f_X = 5.0 MHz .

3. ×: don't care

CHAPTER 6 16-BIT TIMER

6.1 16-Bit Timer Functions

The 16-bit timer has the following functions.

- Timer interrupt
- Timer output
- Buzzer output
- Count value capture

(1) Timer interrupt

An interrupt is generated when a count value and compare value matches.

(2) Timer output

Timer output can be controlled when a count value and compare value matches.

(3) Buzzer output

Buzzer output can be controlled by software.

(4) Count value capture

A count value of 16-bit timer counter 90 (TM90) is latched into a capture register synchronizing with the capture trigger and retained.

6.2 16-Bit Timer Configuration

The 16-bit timer consists of the following hardware.

Table 6-1. Configuration of 16-Bit Timer

Item	Configuration
Timer counter	16 bits × 1 (TM90)
Register	Compare register: 16 bits × 1 (CR90) Capture register: 16 bits × 1 (TCP90)
Timer output	1 (TO90)
Control register	16-bit timer mode control register 90 (TMC90) Buzzer output control register 90 (BZC90) Port mode register 3 (PM3)

(1) 16-bit compare register 90 (CR90)

A value specified in CR90 is compared with the count in 16-bit timer register 90 (TM90). If they match, an interrupt request (INTTM90) is issued by CR90.

CR90 is set with an 8-bit or 16-bit memory manipulation instruction. Any value from 0000H to FFFFH can be set.

$\overline{\text{RESET}}$ input sets CR90 to FFFFH.

Cautions 1. CR90 is designed to be manipulated with a 16-bit memory manipulation instruction. It can also be manipulated with 8-bit memory manipulation instructions, however. When an 8-bit memory manipulation instruction is used to set CR90, it must be accessed in direct addressing.

2. To re-set CR90 during count operation, it is necessary to disable interrupts in advance, using interrupt mask flag register 1 (MK1). It is also necessary to disable inversion of the timer output data, using 16-bit timer mode control register 90 (TMC90).
If the value in CR90 is rewritten in the interrupt-enabled state, an interrupt request may occur at the moment of rewrite.

(2) 16-bit timer counter 90 (TM90)

TM90 is used to count the number of pulses.

The contents of TM90 are read with an 8-bit or 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TM90 to 0000H.

Cautions 1. The count becomes undefined when STOP mode is deselected, because the count operation is performed before oscillation settles.

2. TM90 is designed to be manipulated with a 16-bit memory manipulation instruction. It can also be manipulated with 8-bit memory manipulation instructions, however. When an 8-bit memory instruction is used to manipulate TM90, it must be accessed in direct addressing.

3. When an 8-bit memory manipulation instruction is used to manipulate TM90, the lower and upper bytes must be read as a pair, in this order.

(3) 16-bit capture register 90 (TCP90)

TCP90 captures the contents of TM90.

It is set with an 8-bit or 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes TCP90 undefined.

Caution TCP90 is designed to be manipulated with a 16-bit memory manipulation instruction. It can also be manipulated with 8-bit memory manipulation instructions, however. When an 8-bit memory manipulation instruction is used to manipulate TCP90, it must be accessed in direct addressing.

(4) 16-bit counter read buffer 90

This buffer is used to latch and hold the count for TM90.

6.3 Registers Controlling 16-Bit Timer

The following three types of registers control the 16-bit timer.

- 16-bit timer mode control register 90 (TMC90)
- Buzzer output control register 90 (BZC90)
- Port mode register 3 (PM3)

(1) 16-bit timer mode control register 90 (TMC90)

16-bit timer mode control register 90 (TMC90) controls the setting of a count clock, capture edge, etc.

TMC90 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TMC90 to 00H.

Figure 6-2. Format of 16-Bit Timer Mode Control Register 90

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
TMC90	TOD90	TOF90	CPT901	CPT900	TOC90	TCL901	TCL900	TOE90	FF48H	00H	R/W[Note]

TOD90	Timer output data
0	Timer output of 0
1	Timer output of 1

TOF90	Overflow flag control
0	Reset or cleared by software
1	Set when the 16-bit timer overflows

CPT901	CPT900	Capture edge selection
0	0	Capture operation disabled
0	1	Capture at the rising edge at the CPT90 pin
1	0	Capture at the falling edge at the CPT90 pin
1	1	Capture at both the rising and falling edges at the CPT90 pin

TOC90	Timer output data inversion control
0	Inversion disabled
1	Inversion enabled

TCL901	TCL900	16-bit timer counter 90 count clock selection
0	0	$Fx/2^2$ (1.25MHz)
0	1	$Fx/2^6$ (78.1kHz)
1	0	$Fx/2^7$ (39.1kHz)
1	1	-

TOE90	16-bit timer counter 90 output control
0	Output disabled(port mode)
1	Output enabled

Note Bit 7 is read-only.

Caution Disable the interrupt in advance by using the interrupt mask flag register (MK1) to change the data of TCL901 and TCL900. Also, prevent the timer output data from being inverted by setting TOC90 to1.

Remarks 1. fx: Main system clock oscillation frequency

2. The parenthesized values apply to operation at fx = 5.0 MHz.

(2) Buzzer output control register 90 (BZC90)

This register selects a buzzer frequency based on fcl selected with the count clock select bits (TCL901 and TCL900), and controls the output of a square wave.

BZC90 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears BZC90 to 00H.

Figure 6-3. Format of Buzzer Output Control Register 90

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
BZC90	0	0	0	0	BCS902	BCS901	BCS900	BZOE90	FF49H	00H	R/W Note1

BCS902	BCS901	BCS900	Buzzer frequency		
			fcl=fx/2^2	fcl=fx/2^6	fcl=fx/2^7
0	0	0	fcl/2^4 (78.1kHz)	fcl/2^4 (4.88kHz)	fcl/2^4 (2.44kHz)
0	0	1	fcl/2^5 (39.1kHz)	fcl/2^5 (2.44kHz)	fcl/2^5 (1.22kHz)
0	1	0	fcl/2^8 (4.88kHz)	fcl/2^8 (305Hz)	fcl/2^8 (153Hz)
0	1	1	fcl/2^9 (2.44kHz)	fcl/2^9 (153Hz)	fcl/2^9 (76Hz)
1	0	0	fcl/2^{10} (1.22kHz)	fcl/2^{10} (76Hz)	fcl/2^{10} (38Hz)
1	0	1	fcl/2^{11} (610Hz)	fcl/2^{11} (38Hz)	fcl/2^{11} (19Hz)
1	1	0	fcl/2^{12} (305Hz)	fcl/2^{12} (19Hz)	fcl/2^{12} (10Hz)
1	1	1	fcl/2^{13} (153Hz)	fcl/2^{13} (10Hz)	fcl/2^{13} (5Hz)

BZOE90	Buzzer port output control
0	Disables buzzer port output
1	Enables buzzer port output Note2

Notes 1. Bits 4 to 7 must all be set to 0.

2. When setting BZOE90 to 1, TOE82 must be set to 0. (See **Figure 9-6 Format of 8-Bit Timer Mode Control Register 82**.)

Caution If the subclock is selected as the count clock (TCL901 = 1, TCL900 = 1: see Figure 8-2 Format of 16-Bit Timer Mode Control Register 90), the subclock is not synchronized when buzzer port output is enabled. In this case, the capture function and TM90 read function are disabled. In addition, the count value of TM90 is undefined.

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

(3) Port mode register 3 (PM3)

PM3 is used to set each bit of port 3 to input or output.

When pin P32/INTP2/TO90 is used for timer output, reset the output latch of P32 and PM32 to 0; when pin P33/INTP3/TO82/BZO90 is used for buzzer output,[Note] reset the output latch of P33 and PM33 to 0.

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

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

Note Never output the TO82 and BZO90 signals at the same time.

Figure 6-4. Format of Port Mode Register 3

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM3n	P3n pin I/O mode (n = 2 or 3)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

6.4 16-Bit Timer Operation

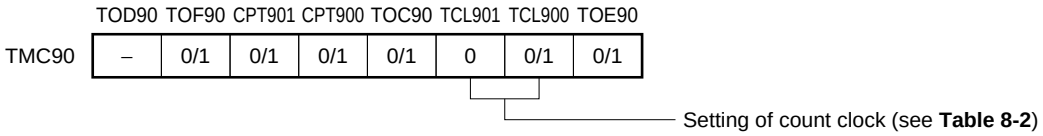
6.4.1 Operation as timer interrupt

In the timer interrupt function, interrupts are repeatedly generated at the count value set in 16-bit compare register 90 (CR90) in advance based on the intervals of the value set in TCL901 and TCL900.

To operate the 16-bit timer as a timer interrupt, the following settings are required.

- Set count values in CR90
- Set 16-bit timer mode control register 90 (TMC90) as shown in Figure 6-5.

Figure 6-5. Settings of 16-Bit Timer Mode Control Register 90 for Timer Interrupt Operation



Caution If 0 is set both for CPT901 and CPT900 flags, capture operation becomes prohibited.

When the count value of 16-bit timer counter 90 (TM90) matches the value set in CR90, counting of TM90 continues and an interrupt request signal (INTTM90) is generated.

Table 8-2 shows interval time, and Figure 8-6 shows timing of timer interrupt operation.

Caution When rewriting the value in CR90 during a count operation, be sure to execute the following processing.

<1> Set interrupt disabled (set TMMK90 (bit 4 of interrupt mask flag register 1 (MK1)) to 1).

<2> Disable inversion control of timer output data (set TOC90 to 0)

If the value in CR90 is rewritten in the interrupt-enabled state, an interrupt request may occur at the moment of rewrite.

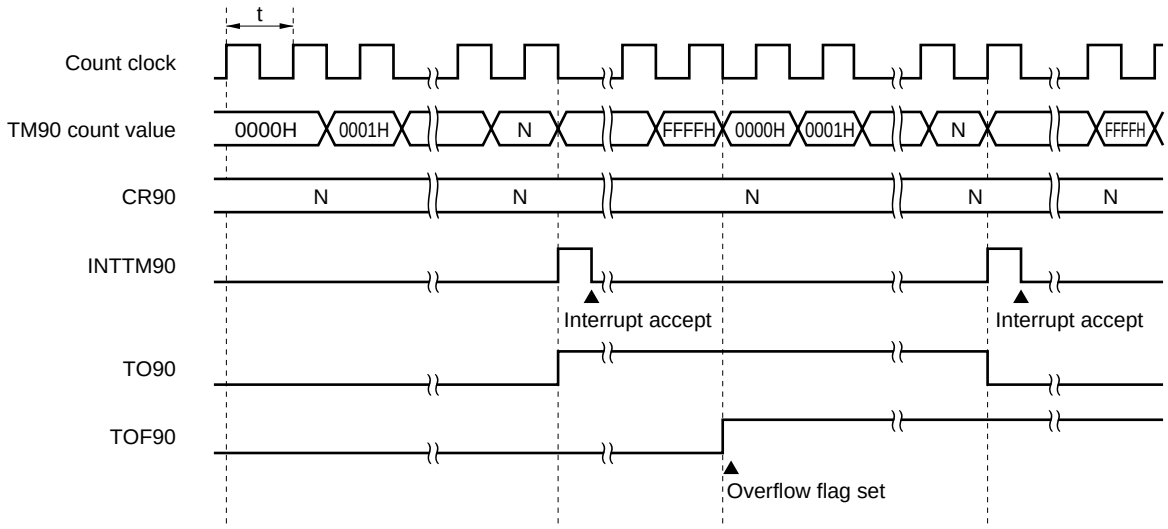
Table 6-2. Interval Time of 16-Bit Timer

TCL901	TCL900	Count Clock	Interval Time
0	0	$2^2/f_X$ (0.8 μs)	$2^{18}/f_X$ (52.4 ms)
0	1	$2^6/f_X$ (12.8 μs)	$2^{22}/f_X$ (838.9 ms)
1	0	$2^7/f_X$ (25.6 μs)	$2^{23}/f_X$ (1.68 s)
1	1	-	-

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

Figure 6-6. Timing of Timer Interrupt Operation



Remark N = 0000H to FFFFH

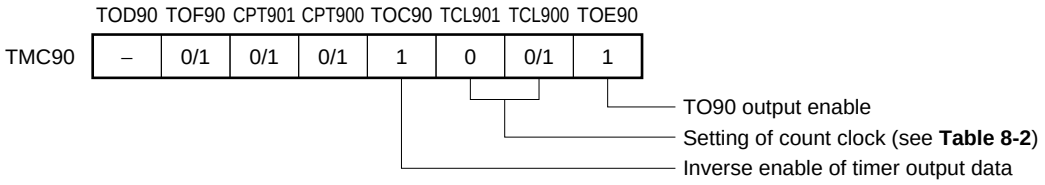
6.4.2 Operation as timer output

Timer outputs are repeatedly generated at the count value set in 16-bit compare register 90 (CR90) in advance based on the intervals of the value set in TCL901 and TCL900.

To operate the 16-bit timer as an timer output, the following settings are required.

- Set P32 to output mode (PM32 = 0).
- Reset output latch of P32 to 0.
- Set the count value in CR90.
- Set 16-bit timer mode control register 90 (TMC90) as shown in Figure 6-7.

Figure 6-7. Settings of 16-Bit Timer Mode Control Register 90 for Timer Output Operation

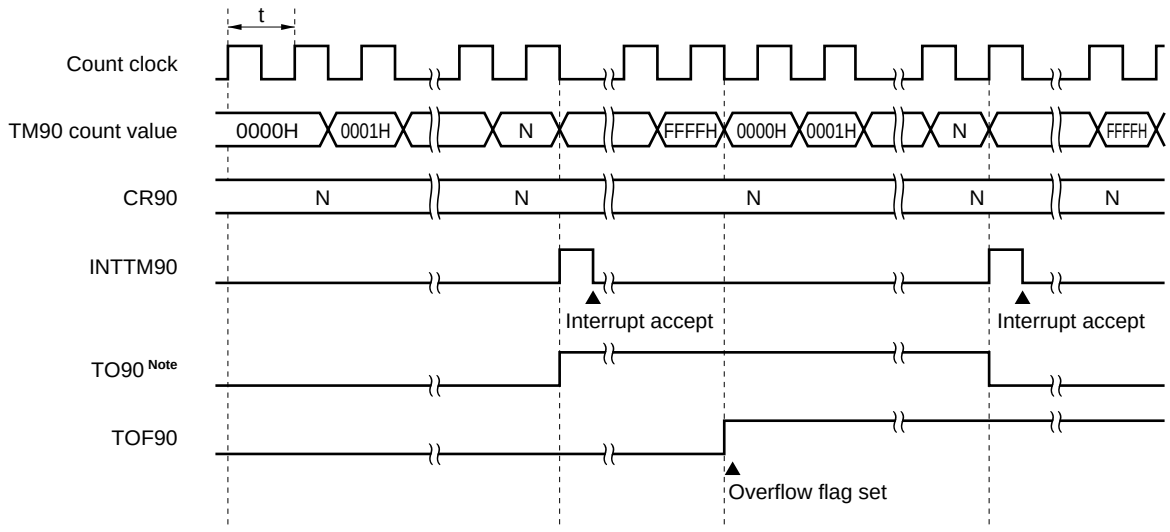


Caution If both the CPT901 flag and CPT900 flag are set to 0, the capture operation becomes prohibited.

When the count value of 16-bit timer counter 90 (TM90) matches the value set in CR90, the output status of the TO90/P32/INTP2 pin is inverted. This enables timer output. At that time, TM90 count is continued and an interrupt request signal (INTTM90) is generated.

Figure 6-8 shows the timing of timer output (see Table 6-2 for the interval time of the 16-bit timer).

Figure 6-8. Timer Output Timing



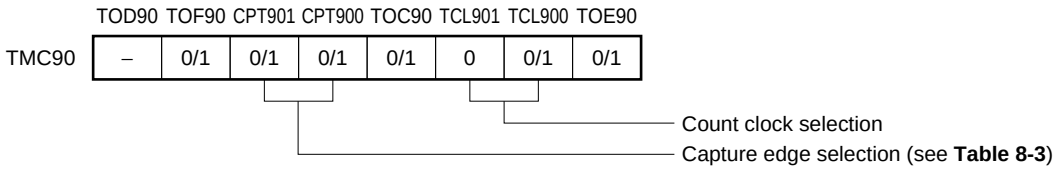
Note The TO90 initial value becomes low level during output enable (TOE90 = 1).

6.4.3 Capture operation

The capture operation consists of latching the count value of 16-bit timer register 90 (TM90) into a capture register synchronizing with a capture trigger, and retaining the count value.

Set TMC90 as shown in Figure 6-9 to allow the 16-bit timer to start the capture operation.

Figure 6-9. Settings of 16-Bit Timer Mode Control Register 90 for Capture Operation



16-bit capture register 90 (TCP90) starts capture operation after a CPT90 capture trigger edge is detected, and latches and retains the count value of 16-bit timer register 90. The TCP90 fetches count value within 2 clocks and retains the count value until the next capture edge detection.

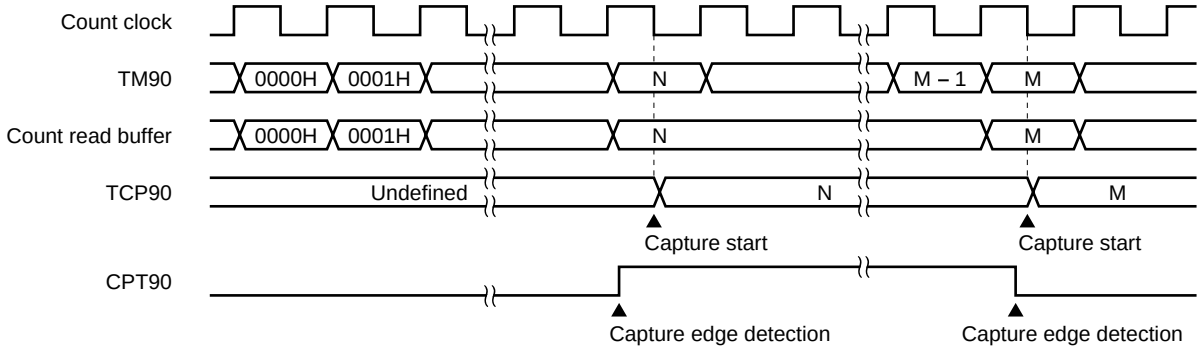
Table 8-3 and Figure 8-10 shows the settings of capture edge and capture operation timing, respectively.

Table 6-3. Settings of Capture Edge

CPT901	CPT900	Capture Edge Selection
0	0	Capture operation prohibited
0	1	CPT90 pin rising edge
1	0	CPT90 pin falling edge
1	1	CPT90 pin both edges

Caution Because TCP90 is rewritten when a capture trigger edge is detected during TCP90 read, disable the capture trigger edge detection during TCP90 read.

Figure 8-10. Capture Operation Timing (Both Edges of CPT90 Pin are Specified)



6.4.4 16-bit timer counter 90 readout

The count value of 16-bit timer counter 90 (TM90) is read out with a 16-bit manipulation instruction.

TM90 readout is performed through a counter read buffer. The counter read buffer latches the TM90 count value. And buffer operation is pended at the CPU clock falling edge after the read signal of the TM90 lower byte rises and the count value is retained. The counter read buffer value at the retention state can be read out as the count value.

Cancellation of pending is performed at the CPU clock falling edge after the read signal of the TM90 higher byte falls.

$\overline{\text{RESET}}$ input clears TM90 to 0000H and TM90 starts freerunning.

Figure 6-11 shows the timing of 16-bit timer counter 90 readout.

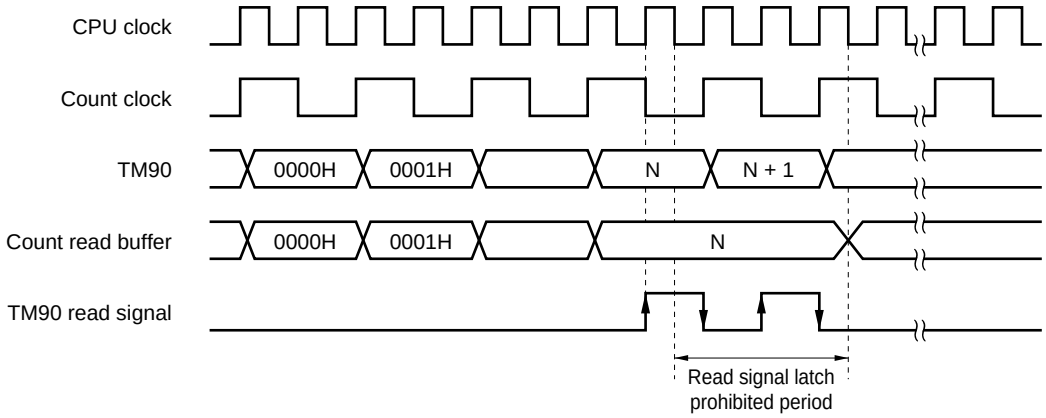
Cautions 1. The count value after releasing stop becomes undefined because the count operation is executed during oscillation stabilization time.

2. Though TM90 is designed for a 16-bit transfer instruction, 8-bit transfer instruction can also be used.

When using the 8-bit transfer instruction, execute it in direct addressing.

3. When using the 8-bit transfer instruction, execute in the order from lower byte to higher byte in pairs. If only the lower byte is read, the pending state of the counter read buffer is not canceled, and if only the higher byte is read, an undefined count value is read.

Figure 6-11. 16-Bit Timer Counter 90 Readout Timing



6.4.5 Buzzer output operation

The buzzer frequency is set using buzzer output control register 90 (BZC90) based on the count clock selected with TCL901 and TCL900 of TMC90 (source clock). A square wave of the set buzzer frequency is output.

Table 6-4 shows the buzzer frequency.

Set the 16-bit timer as follows to use it for buzzer output:

- Set P33 to output mode (PM33 = 0).
- Reset output latch of P33 to 0.
- Set a count clock by using TCL901 and TCL900.
- Set BZC90 as shown in Figure 8-12.
- Clear TOE82 of 8-bit timer mode control register 82 (TMC82) to 0 to disable the output of 8-bit timer 82.

Figure 6-12. Settings of Buzzer Output Control Register 90 for Buzzer Output Operation

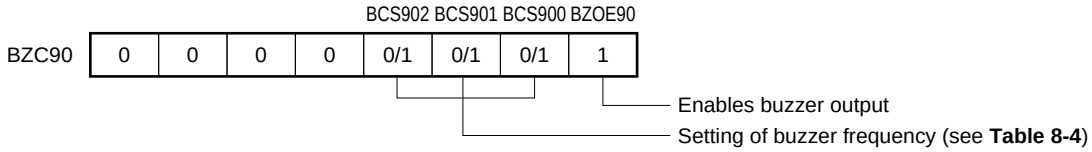


Table 6-4. Buzzer Frequency of 16-Bit Timer

BCS902	BCS901	BCS900	Buzzer Frequency		
			$fcl = f_X/2^2$	$fcl = f_X/2^6$	$fcl = f_X/2^7$
0	0	0	$fcl/2^4$ (78.1 kHz)	$fcl/2^4$ (4.88 kHz)	$fcl/2^4$ (2.44 kHz)
0	0	1	$fcl/2^5$ (39.1 kHz)	$fcl/2^5$ (2.44 kHz)	$fcl/2^5$ (1.22 kHz)
0	1	0	$fcl/2^8$ (4.88 kHz)	$fcl/2^8$ (305 Hz)	$fcl/2^8$ (153 Hz)
0	1	1	$fcl/2^9$ (2.44 kHz)	$fcl/2^9$ (153 Hz)	$fcl/2^9$ (76 Hz)
1	0	0	$fcl/2^{10}$ (1.22 kHz)	$fcl/2^{10}$ (76 Hz)	$fcl/2^{10}$ (38 Hz)
1	0	1	$fcl/2^{11}$ (610 Hz)	$fcl/2^{11}$ (38 Hz)	$fcl/2^{11}$ (19 Hz)
1	1	0	$fcl/2^{12}$ (305 Hz)	$fcl/2^{12}$ (19 Hz)	$fcl/2^{12}$ (10 Hz)
1	1	1	$fcl/2^{13}$ (153 Hz)	$fcl/2^{13}$ (10 Hz)	$fcl/2^{13}$ (5 Hz)

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz .

6.5 Notes on Using 16-Bit Timer

Usable functions differ according to the settings of the count clock selection, CPU clock operation, system clock oscillation status, and BZOE90 (bit 0 of buzzer output control register 90 (BZC90)).

Refer to the following table.

Count Clock	CPU Clock	System Clock		BZOE90	Capture	TM90 Read	Buzzer Output	Timer Output	Timer Interrupt
		Main System Clock	Subsystem Clock						
$f_X/2^2$, $f_X/2^6$, $f_X/2^7$	Main	Oscillating	Oscillating/Stopped	1/0	√	√ Note 1	**Note 2**	√	√
		Stopped			×	×	×	×	×

Notes 1. TM90 is enabled only when CPU clock is in high-speed mode.

2. Output is enabled when BZOE90 = 1.

Cautions 1. The capture function uses $f_X/2$ for control (refer to Figure 8-1 Block Diagram of 16-Bit Timer). Therefore, the capture function cannot be used when the main system clock is stopped.

2. The read function of TM90 uses the CPU clock for control (refer to Figure 8-1), and reads an undefined value when the CPU clock is slower than the count clock (values are not guaranteed). When reading TM90, set the count clock to the same speed as the CPU clock (when the CPU clock is main system clock, high-speed mode is set), or select a clock slower than the CPU clock.

CHAPTER 7 8-BIT TIMER/EVENT COUNTERS

7.1 Functions of 8-Bit Timer/Event Counters

The 8-bit timer/event counters (TM80 and TM81) and 8-bit timer (TM82) have the following functions:

- Interval timer (TM80, TM81, TM82)
- External event counter (TM80, TM81 only)
- Square wave output (TM80, TM81, TM82)
- PWM output (TM80, TM81, TM82)

(1) 8-bit interval timer

When an 8-bit timer/event counter is used as an interval timer, it generates an interrupt at any time intervals set in advance.

Table 7-1. Interval Time of 8-Bit Timer/Event Counter 80

Minimum Interval Time	Maximum Interval Time	Resolution
$1/f_X$ (200 ns)	$2^8/f_X$ (51.2 μs)	$1/f_X$ (200 ns)
$2^3/f_X$ (1.6 μs)	$2^{11}/f_X$ (409.6 μs)	$2^3/f_X$ (1.6 μs)

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

Table 7-2. Interval Time of 8-Bit Timer/Event Counter 81

Minimum Interval Time	Maximum Interval Time	Resolution
$2^4/f_X$ (3.2 μs)	$2^{12}/f_X$ (819.2 μs)	$2^4/f_X$ (3.2 μs)
$2^8/f_X$ (51.2 μs)	$2^{16}/f_X$ (13.1 ms)	$2^8/f_X$ (51.2 μs)

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

Table 7-3. Interval Time of 8-Bit Timer 82

Minimum Interval Time	Maximum Interval Time	Resolution
$2^5/f_X$ (6.4 μs)	$2^{13}/f_X$ (1.64 ms)	$2^5/f_X$ (6.4 μs)
$2^7/f_X$ (25.6 μs)	$2^{15}/f_X$ (6.55 ms)	$2^7/f_X$ (25.6 μs)

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

(2) External event counter

The number of pulses of an externally input signal can be counted.

(3) Square wave output

A square wave of arbitrary frequency can be output.

Table 7-4. Square Wave Output Range of 8-Bit Timer/Event Counter 80

Minimum Pulse Width	Maximum Pulse Width	Resolution
$1/f_X$ (200 ns)	$2^8/f_X$ (51.2 μs)	$1/f_X$ (200 ns)
$2^3/f_X$ (1.6 μs)	$2^{11}/f_X$ (409.6 μs)	$2^3/f_X$ (1.6 μs)

Remark f_X: Main system clock oscillation frequency. The parenthesized values apply to operation at f_X = 5.0 MHz.

Table 7-5. Square Wave Output Range of 8-Bit Timer/Event Counter 81

Minimum Pulse Width	Maximum Pulse Width	Resolution
$2^4/f_X$ (3.2 μs)	$2^{12}/f_X$ (819.2 μs)	$2^4/f_X$ (3.2 μs)
$2^8/f_X$ (51.2 μs)	$2^{16}/f_X$ (13.1 ms)	$2^8/f_X$ (51.2 μs)

Remark f_X: Main system clock oscillation frequency. The parenthesized values apply to operation at f_X = 5.0 MHz.

Table 7-6. Square Wave Output Range of 8-Bit Timer 82

Minimum Pulse Width	Maximum Pulse Width	Resolution
$2^5/f_X$ (6.4 μs)	$2^{13}/f_X$ (1.64 ms)	$2^5/f_X$ (6.4 μs)
$2^7/f_X$ (25.6 μs)	$2^{15}/f_X$ (6.55 ms)	$2^7/f_X$ (25.6 μs)

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

(4) PWM output

8-bit resolution PWM output can be produced.

7.2 8-Bit Timer/Event Counter Configuration

The 8-bit timer/event counter consists of the following hardware.

Table 7-7. 8-Bit Timer/Event Counter Configuration

Item	Configuration
Timer counter	8 bits × 3 (TM80 to TM82)
Register	Compare register: 8 bits × 3 (CR80 to CR82)
Timer output	3 (TO80 to TO82)
Control register	8-bit timer mode control register 80 to 82 (TMC80 to TMC82) Port mode register 2, 3 (PM2, PM3)

Figure 7-1. Block Diagram of 8-Bit Timer/Event Counter 80

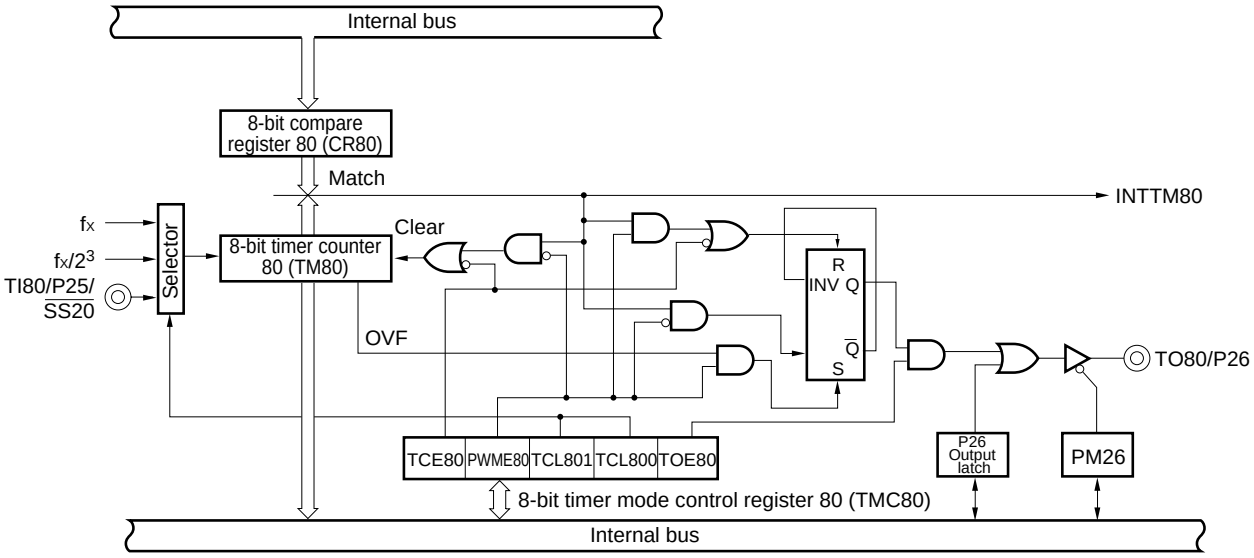


Figure 7-2. Block Diagram of 8-Bit Timer/Event Counter 81

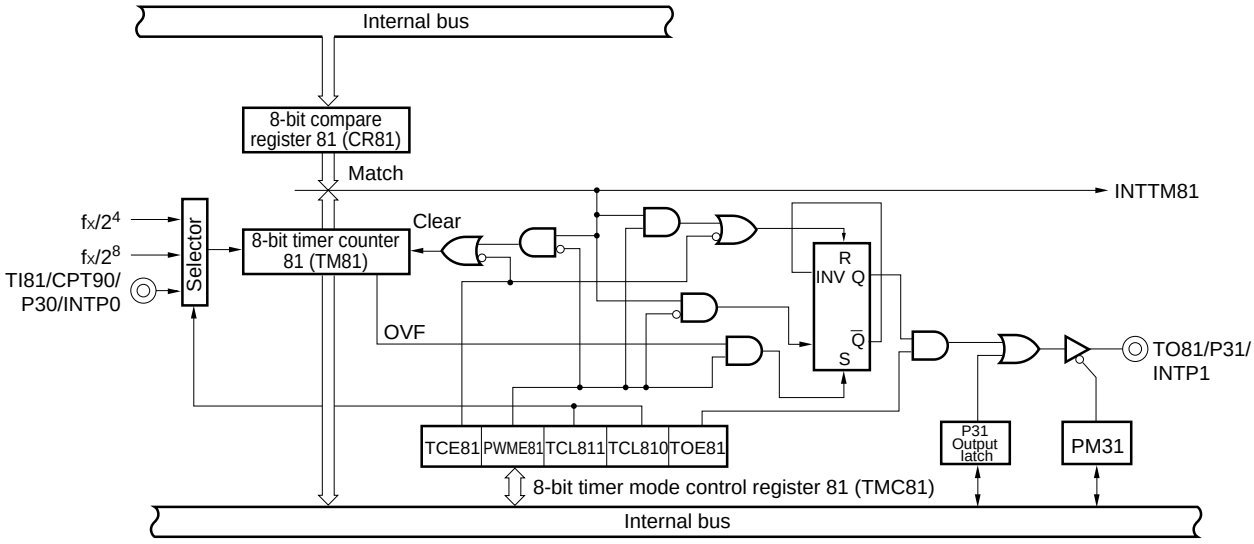
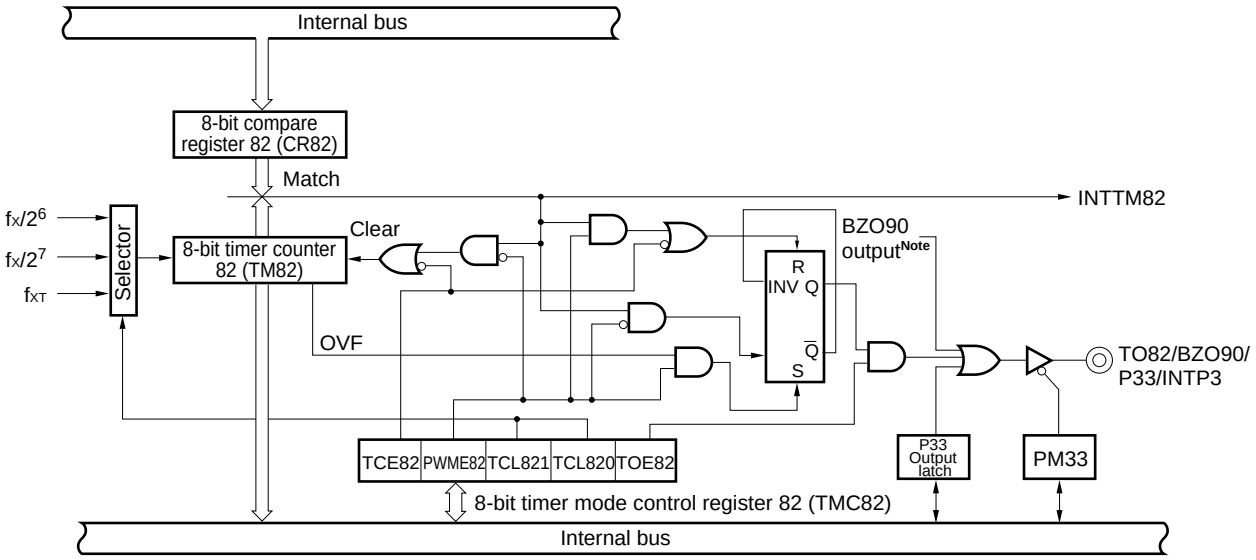


Figure 7-3. Block Diagram of 8-Bit Timer 82



Note See **Figure 7-1 Block Diagram of 16-Bit Timer**.

(1) 8-bit compare register 8n (CR8n)

A value specified in CR8n is compared with the count in 8-bit timer counter 8n (TM8n). If they match, an interrupt request (INTTM8n) is issued.

CR8n is set with an 8-bit memory manipulation instruction. Any value from 00H to FFH can be set.

$\overline{\text{RESET}}$ input makes CR8n undefined.

Cautions 1. Before rewriting CR8n, stop the timer operation once. If CR8n is rewritten in the timer operation-enabled state, a match interrupt request signal may occur at the moment of rewrite.

2. Do not clear CR8n to 00H in PWM output mode (when PWME8n = 1: bit 6 of 8-bit timer mode control register 8n (TMC8n)); otherwise, PWM output may not be produced normally.

Remark n = 0 to 2

(2) 8-bit timer counter 8n (TM8n)

TM8n is used to count the number of pulses.

Its contents are read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TM8n to 00H.

Remark n = 0 to 2

7.3 8-Bit Timer/Event Counter Control Registers

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

- 8-bit timer mode control registers 80, 81, and 82 (TMC80, TMC81, and TMC82)
- Port mode registers 2 and 3 (PM2 and PM3)

(1) 8-bit timer mode control register 80 (TMC80)

TMC80 determines whether to enable or disable 8-bit timer counter 80 (TM80), specifies the count clock for TM80, and controls the operation of the output control circuit of 8-bit timer/event counter 80.

TMC80 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TMC80 to 00H.

Figure 7-4. Format of 8-Bit Timer Mode Control Register 80

Symbol	<7>	<6>	5	4	3	2	1	<0>	Address	After reset	R/W
TMC80	TCE80	PWME80	0	0	0	TCL801	TCL800	TOE80	FF53H	00H	R/W

TCE80	TM80 operation control
0	Operation disabled (TM80 is cleared to 00H.)
1	Operation enabled

PWME80	PWM output selection
0	Timer counter operation mode
1	PWM output operation mode

TCL801	TCL800	8-bit timer counter 80 count clock selection
0	0	f_X (5.0 MHz)
0	1	$f_X/2^3$ (625 kHz)
1	0	Rising edge of TI80
1	1	Falling edge of TI80

TOE80	8-bit timer/event counter 80 output control
0	Output disabled (port mode)
1	Output enabled

Cautions 1. Always stop the timer before setting TMC80.

2. For PWM mode operation, the interrupt mask flag (TMMK80) must be set.

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

(2) 8-bit timer mode control register 81 (TMC81)

TMC81 determines whether to enable or disable 8-bit timer counter 81 (TM81), specifies the count clock for TM81, and controls the operation of the output control circuit of 8-bit timer/event counter 81.

TMC81 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TMC81 to 00H.

Figure 7-5. Format of 8-Bit Timer Mode Control Register 81

Symbol	<7>	<6>	5	4	3	2	1	<0>	Address	After reset	R/W
TMC81	TCE81	PWME81	0	0	0	TCL811	TCL810	TOE81	FF57H	00H	R/W

TCE81	TM81 operation control
0	Operation disabled (TM81 is cleared to 00H.)
1	Operation enabled

PWME81	PWM output selection
0	Timer counter operation mode
1	PWM output operation mode

TCL811	TCL810	8-bit timer counter 81 count clock selection
0	0	$f_X/2^4$ (312.5 kHz)
0	1	$f_X/2^8$ (19.5 kHz)
1	0	Rising edge of TI81
1	1	Falling edge of TI81

TOE81	8-bit timer/event counter 81 output control
0	Output disabled (port mode)
1	Output enabled

Cautions 1. Always stop the timer before setting TMC81.

2. For PWM mode operation, the interrupt mask flag (TMMK81) must be set.

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

(3) 8-bit timer mode control register 82 (TMC82)

TMC82 determines whether to enable or disable 8-bit timer counter 82 (TM82) and specifies the count clock for TM82. It also controls the operation of the output control circuit of 8-bit timer 82.

TMC82 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TMC82 to 00H.

Figure 7-6. Format of 8-Bit Timer Mode Control Register 82

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
TMC82	TCE82	PWME82	0	0	0	0	TCL820	TOE82	FF5BH	00H	R/W

TCE82	TM82 operation control
0	Operation disabled(TM82 is cleared to 00H)
1	Operation enabled

PWME82	PWM output selection
0	Timer counter operation mode
1	PWM output operation mode

TCL820	8-bit timer counter 82 count clock selection
0	$fx/2^5$(156.3kHz)
1	$fx/2^7$(39.1kHz)

TOE82	8-bit timer 82 output control
0	Output disabled(port mode)
1	Output enabled[Note]

Note When TOE82 is set to 1, BZOE90 must be set to 0 (see **Figure 8-3 Format of Buzzer Output Control Register 90**).

Cautions 1. Always stop the timer before setting TMC82.

2. For PWM mode operation, the interrupt mask flag (TMMK82) must be set.

Remarks 1. fx: Main system clock oscillation frequency

2. The parenthesized values apply to operation at fx = 5.0 MHz.

(4) Port mode registers 2 and 3 (PM2 and PM3)

PM2 and PM3 specify whether each bit of port 2 and port 3 is used for input or output.

To use the P26/TO80 pin for timer output, the PM26 and P26 output latch must be reset to 0.

To use the P31/TO81/INTP1 pin for timer output, the PM31 and P31 output latch must be reset to 0.

To use the P33/INTP3/TO82/BZO90 pin for timer output, the PM33 and P33 output latch must be reset to 0.

PM2 and PM3 are set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM2 and PM3 to FFH.

Figure 7-7. Format of Port Mode Register 2

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM2	1	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W

PM26	P26 pin input/output mode selection
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

Figure 7-8. Format of Port Mode Register 3

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM31	P31 pin input/output mode selection
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

PM33	P33 pin input/output mode selection
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

7.4 Operation of 8-Bit Timer/Event Counter

7.4.1 Operation as interval timer

The interval timer repeatedly generates an interrupt at time intervals specified by the count value set in 8-bit compare register 8n (CR8n) in advance.

To operate the 8-bit timer/event counter as an interval timer, the following settings are required.

<1> Set 8-bit timer counter 8n (TM8n) to operation disable (by setting TCE8n (bit 7 of 8-bit timer mode control register 8n (TMC8n)) to 0).
<2> Set the count clock of the 8-bit timer/event counter (see Tables 9-8 to 9-10).
<3> Set a count value in CR8n.
<4> Set TM8n to operation enabled (TCE8n = 1).

When the count value of 8-bit timer counter 8n (TM8n) matches the value set in CR8n, TM8n is cleared to 00H and continues counting. At the same time, an interrupt request signal (INTTM8n) is generated.

Tables 9-8 to 9-10 show interval time, and Figure 9-10 shows the timing of interval timer operation.

Cautions 1. Before rewriting CR8n, stop the timer operation once. If CR8n is rewritten in the timer operation-enabled state, a match interrupt request signal may occur at the moment of rewrite.

2. If the count clock setting and TM8n operation-enabled are set in TCM8n simultaneously using an 8-bit memory manipulation instruction, an error of more than a clock in one cycle may occur after the timer start. Therefore, always follow the above procedure when operating the 8-bit timer/event counter as an interval timer.

Remark n = 0 to 2

Table 7-8. Interval Time of 8-Bit Timer/Event Counter 80

TCL801	TCL800	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	$1/f_X$ (200 ns)	$2^8/f_X$ (51.2 μs)	$1/f_X$ (200 ns)
0	1	$2^3/f_X$ (1.6 μs)	$2^{11}/f_X$ (409.6 μs)	$2^3/f_X$ (1.6 μs)
1	0	TI80 input cycle	$2^8 \times$ TI80 input cycle	TI80 input edge cycle
1	1	TI80 input cycle	$2^8 \times$ TI80 input cycle	TI80 input edge cycle

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

Table 7-9. Interval Time of 8-Bit Timer/Event Counter 81

TCL811	TCL810	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	$2^4/f_X$ (3.2 μs)	$2^{12}/f_X$ (819.2 μs)	$2^4/f_X$ (3.2 μs)
0	1	$2^8/f_X$ (51.2 μs)	$2^{16}/f_X$ (13.1 ms)	$2^8/f_X$ (51.2 μs)
1	0	TI81 input cycle	$2^8 \times$ TI81 input cycle	TI81 input edge cycle
1	1	TI81 input cycle	$2^8 \times$ TI81 input cycle	TI81 input edge cycle

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

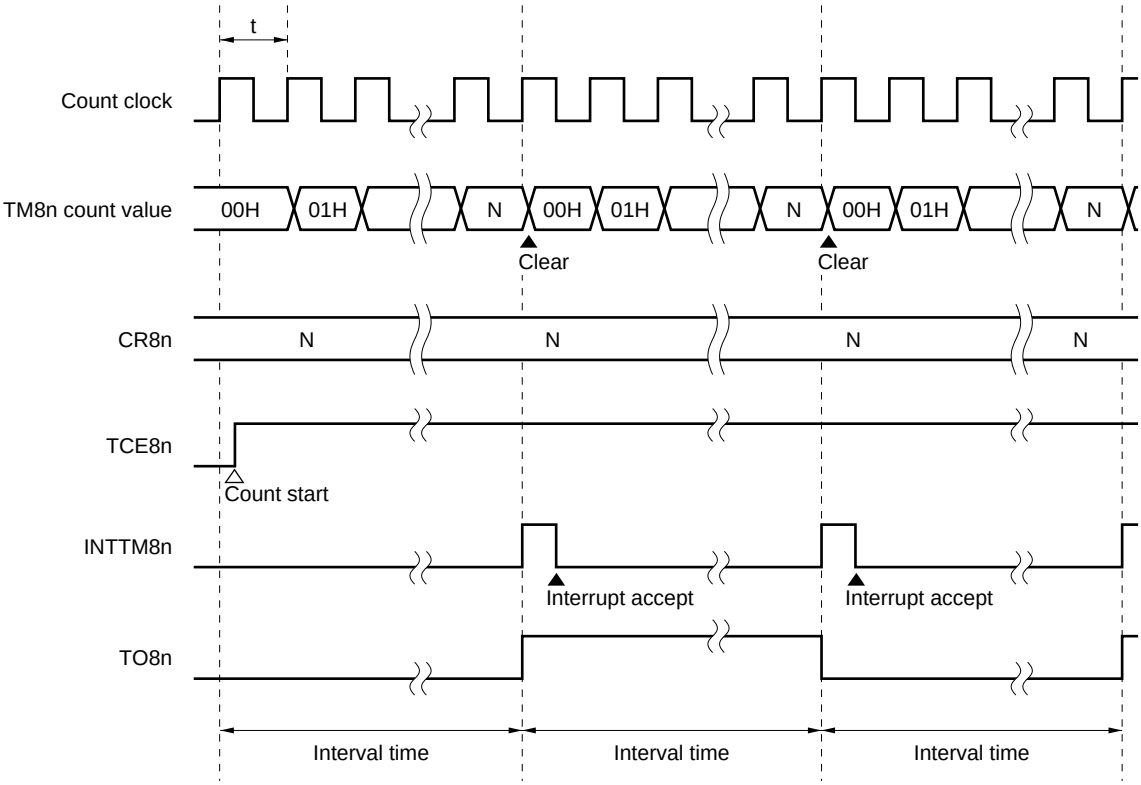
Table 7-10. Interval Time of 8-Bit Timer 82

TCL820	Minimum Interval Time	Maximum Interval Time	Resolution
0	$2^5/f_X$ (6.4 μs)	$2^{13}/f_X$ (1.64 ms)	$2^5/f_X$ (6.4 μs)
1	$2^7/f_X$ (25.6 μs)	$2^{15}/f_X$ (6.55 ms)	$2^7/f_X$ (25.6 μs)

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

Figure 7-9. Interval Timer Operation Timing



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

2. n = 0 to 2

7.4.2 Operation as external event counter[Note]

The external event counter counts the number of external clock pulses input to the TI80/P25/$\overline{SS20}$ or TI81/P30/INTP0/CPT90 pin by using 8-bit timer counters 80 and 81 (TM80 and TM81).

To operate the 8-bit timer/event counter as an external event counter, the following settings are required.

<1> Set P25 and P30 to input mode (PM25 = 1, PM30 = 1).

<2> Set 8-bit timer register 8n (TM8n) to operation disable (by setting TCE8n (bit 7 of 8-bit timer mode control register 8n (TMC8n)) to 0).

<3> Specify the rising/falling edges of TI8n (see Tables 9-8 and 9-9).

<4> Set a count value in CR8n.

<5> Set TM8n to operation enabled (TCE8n = 1).

Note Only TM80 and TM81 have this function.

Each time the valid edge specified by bit 1 (TCL8n0) of TMC8n is input, the value TM8n is incremented.

When the count value of TM8n matches the value set in CR8n, TM8n is cleared to 0 and continues counting. At the same time, an interrupt request signal (INTTM8n) is generated.

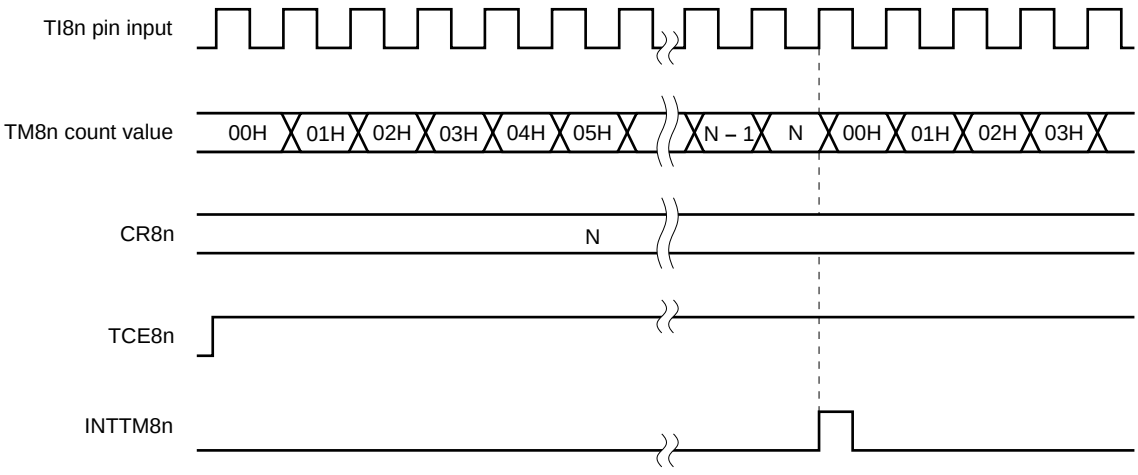
Figure 9-10 shows the timing of the external event counter operation (with rising edge specified).

Cautions 1. Before rewriting CR8n, stop the timer operation once. If CR8n is rewritten in the timer operation-enabled state, a match interrupt request signal may occur at the moment of rewrite.

2. If the count clock setting and TM8n operation-enabled are set in TCM8n simultaneously using an 8-bit memory manipulation instruction, an error of more than a clock in one cycle may occur after the timer start. Therefore, always follow the above procedure when operating the 8-bit timer/event counter as an external event counter.

Remark n = 0, 1

Figure 9-10. External Event Counter Operation Timing (with Rising Edge Specified)



Remarks 1. N = 00H to FFH

2. n = 0, 1

7.4.3 Operation as square wave output

The 8-bit timer/event counter can generate output square waves of an arbitrary frequency at intervals specified by the count value set in 8-bit compare registers 8n (CR8n) in advance.

To operate 8-bit timer/event counters 8n for square wave output, the following settings are required.

<1> Set P26, P31, and P33 to output mode (PM26 = 0, PM31 = 0, PM33 = 0).

<2> Reset the output latches of P26, P31, and P33 to 0.

<3> Set 8-bit timer counter 8n (TM8n) to operation disable (by setting TCE8n (bit 7 of 8-bit timer mode control register 8n (TMC8n)) to 1).

<4> Set the count clock of 8-bit timer/event counter 8n and set TO8n to output enable (TOE8n (bit 0 of TMC8n) = 1).

<5> Set count value in CR8n.

<6> Set TM8n to operation enable (TCE8n = 1).

When the count value of TM8n matches the value set in CR8n, the TO8n pin output will be inverted. Through application of this mechanism, square waves of any frequency can be output. As soon as a match occurs, TM8n will be cleared to 00H and resumes to count, generating an interrupt request signal (INTTM8n).

Setting 0 for bit 7 (TCE8n) of TMC8n clears the square-wave output to 0.

Tables 7-11 through 7-13 show square wave output range, and Figure 7-11 shows timing of square wave output.

Cautions 1. Before rewriting CR8n, stop the timer operation once. If CR8n is rewritten in the timer operation-enabled state, a match interrupt request signal may occur at the moment of rewrite.

2. If the count clock setting and TM8n operation-enabled are set in TCM8n simultaneously using an 8-bit memory manipulation instruction, an error of more than a clock in one cycle may occur after the timer start. Therefore, always follow the above procedure when operating the 8-bit timer/event counter for square wave output.

Remark n = 0 to 2

Table 7-11. Square Wave Output Range of 8-Bit Timer/Event Counter 80

TCL801	TCL800	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	$1/f_X$ (200 ns)	$2^8/f_X$ (51.2 μs)	$1/f_X$ (200 ns)
0	1	$2^3/f_X$ (1.6 μs)	$2^{11}/f_X$ (409.6 μs)	$2^3/f_X$ (1.6 μs)

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

Table 7-12. Square Wave Output Range of 8-Bit Timer/Event Counter 81

TCL811	TCL810	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	$2^4/f_X$ (3.2 μs)	$2^{12}/f_X$ (819.2 μs)	$2^4/f_X$ (3.2 μs)
0	1	$2^8/f_X$ (51.2 μs)	$2^{16}/f_X$ (13.1 ms)	$2^8/f_X$ (51.2 μs)

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

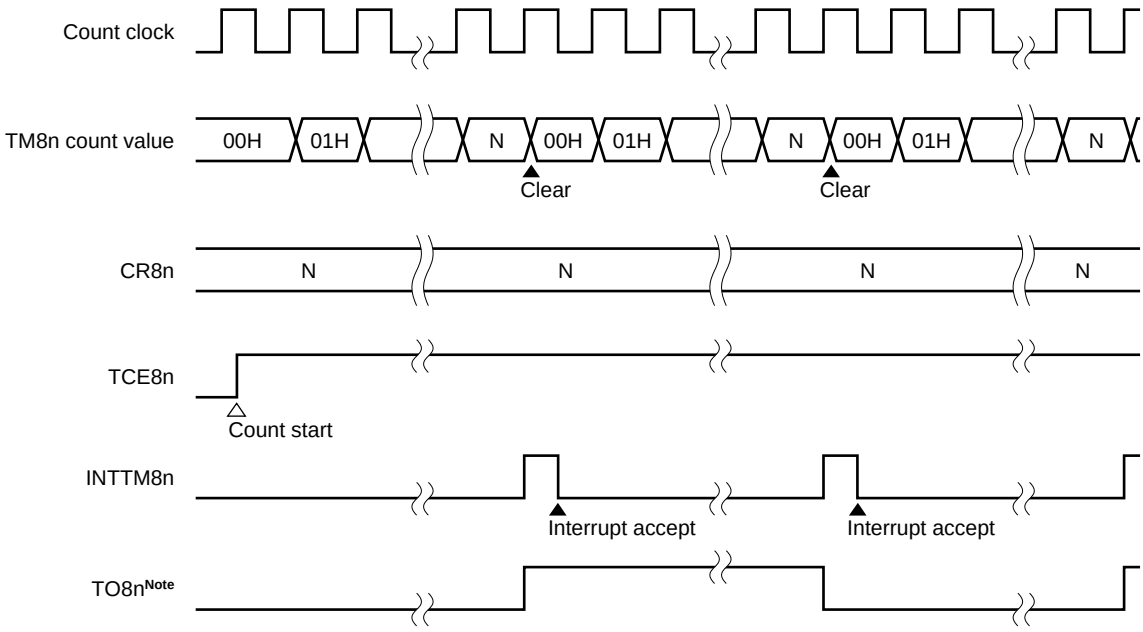
Table 7-13. Square Wave Output Range of 8-Bit Timer 82

TCL820	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	$2^5/f_X$ (6.4 μs)	$2^{13}/f_X$ (1.64 ms)	$2^5/f_X$ (6.4 μs)
1	$2^7/f_X$ (25.6 μs)	$2^{15}/f_X$ (6.55 ms)	$2^7/f_X$ (25.6 μs)

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

Figure 7-11. Square Wave Output Timing



Note The initial value of TO8n is low for output enable (TOE8n = 1).

Remark n = 0 to 2

7.4.4 PWM output operation

PWM output enables an interruption repeatedly at intervals specified by the count value set in 8-bit compare register 8n (CR8n) in advance.

To use 8-bit timer/event counter 8n for PWM output, the following settings are required.

<1> Set P26, P31, and P33 to output mode (PM26 = 0, PM31 = 0, PM33 = 0).

<2> Reset the output latches of P26, P31, and P33 to 0.

<3> Set 8-bit timer counter 8n (TM8n) to operation disable (by setting TCE8n (bit 7 of 8-bit timer mode control register 8n (TMC8n)) to 0).

<4> Set the count clock of 8-bit timer/event counter 8n, and set TO8n to output enable (TOE8n (bit 0 of TMC8n) = 1), and to PWM output mode (PWME8n = 1).

<5> Set a count value in CR8n.

<6> Set TM8n to operation enable (TCE8n = 1).

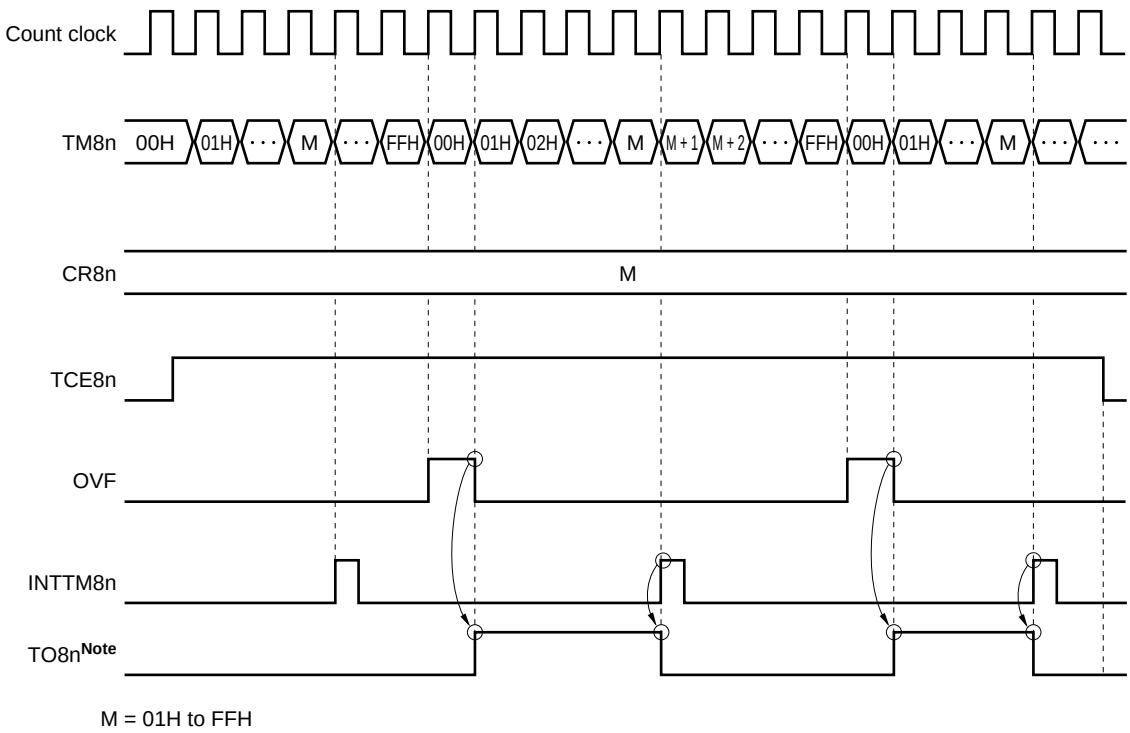
When the count value of TM8n matches the value set in CR8n, TM8n continues counting, and an interrupt request signal (INTTM8n) is generated.

Cautions 1. Before rewriting CR8n, stop the timer. If CR8n is rewritten in the timer operation-enabled state, a high-level signal may be output for the next cycle (256 count pulses) (for details, see 9.5 (3) Timer operation after compare register is rewritten during PWM output).

2. If the count clock setting and TM8n operation-enabled are set in TCM8n simultaneously using an 8-bit memory manipulation instruction, an error of more than a clock in one cycle may occur after the timer start. Therefore, always follow the above procedure when operating the 8-bit timer/event counter for PWM output.

Remark n = 0 to 2

Figure 7-12. PWM Output Timing



Note The initial value of TO8n is low for output enable (TOE8n = 1).

Caution Do not set CR8n to 00H in PWM output mode; otherwise, PWM may not be output normally.

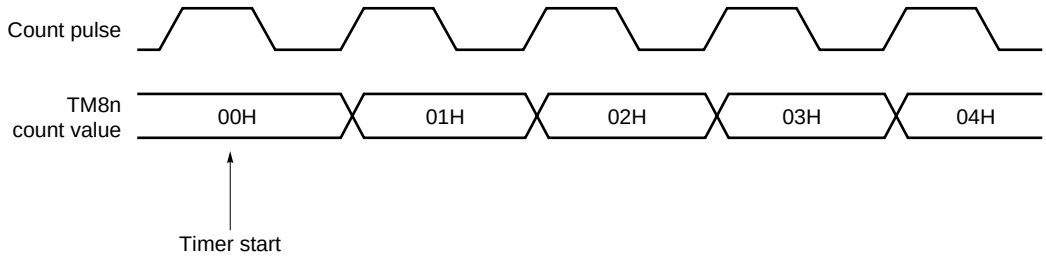
Remark n = 0 to 2

7.5 Notes on Using 8-Bit Timer/Event Counters

(1) Error on starting timer

An error of up to 1 clock is included in the time between the timer being started and a coincidence signal being generated. This is because 8-bit timer counter 8n (TM8n) is started in asynchronization with the count pulse.

Figure 7-13. Start Timing of 8-Bit Timer Counter



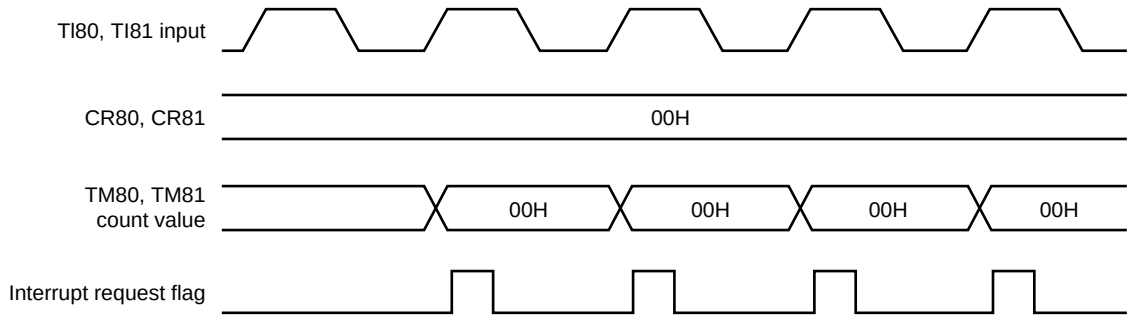
Remark n = 0 to 2

(2) Setting of 8-bit compare register 8n

8-bit compare register 8n (CR8n) can be set to 00H.

Therefore, one pulse can be counted when an 8-bit timer/event counter operates as an event counter.

Figure 7-14. External Event Counter Operation Timing



Cautions 1. When CR8n is rewritten to timer counter operation mode (PWME8n (8-bit timer mode control register 8n (TMC8n)) = 0), be sure to stop the timer operation beforehand. If CR8n is rewritten in the timer operation-enabled state, a match interrupt request signal may occur at the moment of rewrite.

2. If CR8n is rewritten during timer operation in the PWM operation mode (PWME8n = 1), pulses may not be generated for one cycle after the rewrite.

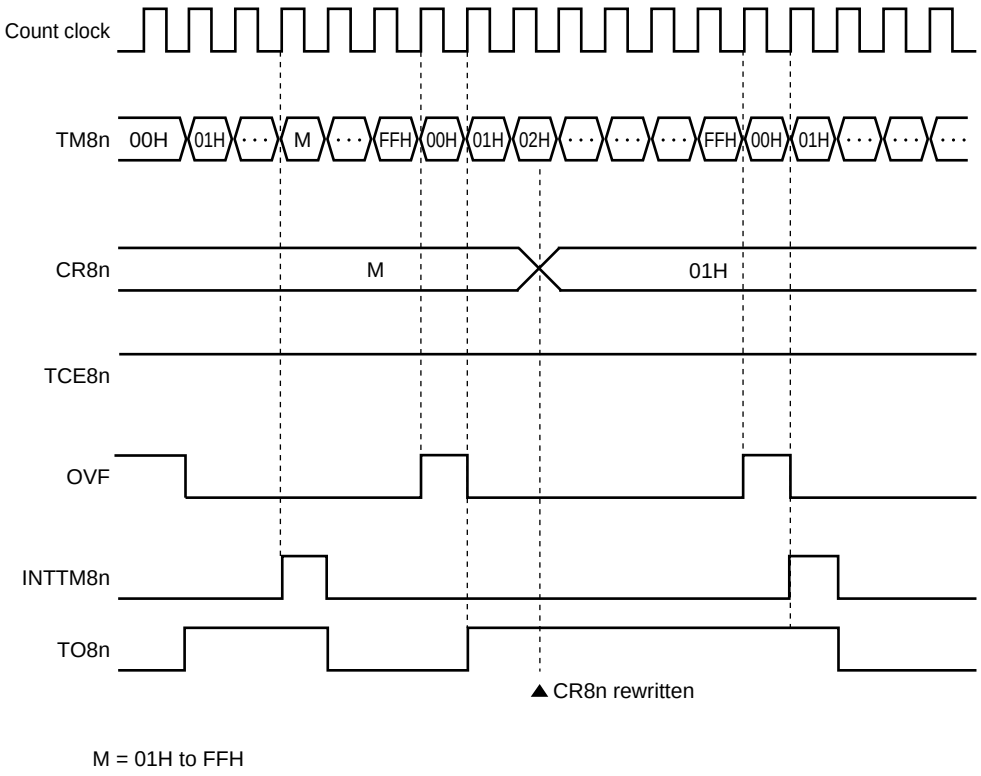
3. Do not set CR8n to 00H in PWM operation mode; otherwise, PWM may not be output normally.

Remark n = 0 to 2

(3) Timer operation after compare register is rewritten during PWM output

When 8-bit compare register 8n (CR8n) is rewritten during PWM output, if the new value is smaller than that of 8-bit timer/counter 8n (TM8n), a high-level signal may be output for the next cycle (256 count pulses) after the CR8n value is rewritten. Figure 9-15 shows the timing at which the high-level signal is output.

Figure 7-15. Operation Timing after Compare Register is Rewritten during PWM Output



M = 01H to FFH

Remark n = 0 to 2

CHAPTER 8 WATCH TIMER

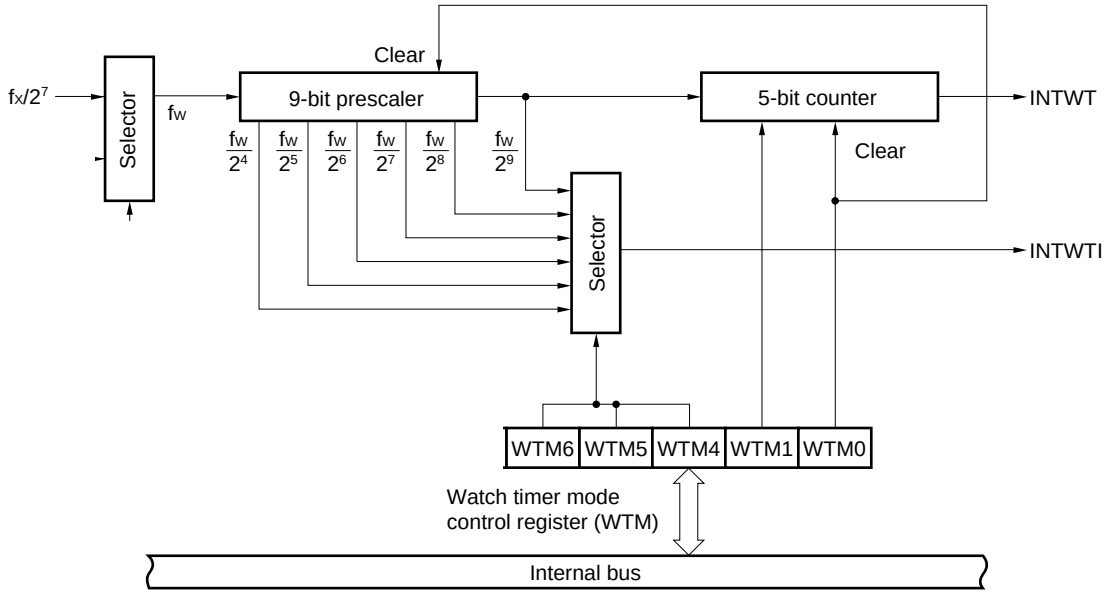
8.1 Watch Timer Functions

The watch timer has the following functions:

- Watch timer
- Interval timer

The watch and interval timers can be used at the same time.
Figure 8-1 is a block diagram of the watch timer.

Figure 8-1. Block Diagram of Watch Timer



(1) Watch timer

The 4.19-MHz main system clock is used to issue an interrupt request (INTWT) at 0.5-second intervals.

Caution When the main system clock is operating at 5.0 MHz, it cannot be used to generate a 0.5-second interval.

(2) Interval timer

The interval timer is used to generate an interrupt request (INTWTI) at specified intervals.

Table 8-1. Interval Generated Using the Interval Timer

Interval	At f_X = 5.0 MHz	At f_X = 4.19 MHz
$2^4 \times 1/f_W$	409.6 μs	489 μs
$2^5 \times 1/f_W$	819.2 μs	978 μs
$2^6 \times 1/f_W$	1.64 ms	1.96 ms
$2^7 \times 1/f_W$	3.28 ms	3.91 ms
$2^8 \times 1/f_W$	6.55 ms	7.82 ms
$2^9 \times 1/f_W$	13.1 ms	15.6 ms

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

2. f_X: Main system clock oscillation frequency

8.2 Watch Timer Configuration

The watch timer consists of the following hardware.

Table 8-2. Watch Timer Configuration

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

8.3 Watch Timer Control Register

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

- Watch timer mode control register (WTM)
  WTM selects a count clock for the watch timer and specifies whether to enable operation of the timer. It also specifies the prescaler interval and how the 5-bit counter is controlled.
  WTM is set with a 1-bit or 8-bit memory manipulation instruction.
  $\overline{\text{RESET}}$ input sets WTM to 00H.

Figure 8-2. Format of Watch Timer Mode Control Register

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
WTM	0	WTM6	WTM5	WTM4	0	0	WTM1	WTM0	FF4AH	00H	R/W

WTM6	WTM5	WTM4	Prescaler interval selection
0	0	0	2^4/fw(488 μs)
0	0	1	2^5/fw(977 μs)
0	1	0	2^6/fw(1.95ms)
0	1	1	2^7/fw(3.91ms)
1	0	0	2^8/fw(7.81ms)
1	0	1	2^9/fw(15.6ms)
Other than above			Setting prohibited

WTM1	Control of 5-bit counter operation
0	Cleared after stop
1	Started

WTM0	Watch timer operation
0	Operation disabled(both prescaler and timer cleard)
1	Operation enabled

Remarks 1. fw: Watch timer clock frequency ($f_X/2^7$)

2. fx: Main system clock oscillation frequency

8.4 Watch Timer Operation

8.4.1 Operation as watch timer

The main system clock (4.19 MHz) is used to enable the watch timer to operate at 0.5-second intervals.

The watch timer is used to generate an interrupt request at specified intervals.

By setting bits 0 and 1 (WTM0 and WTM1) of the watch timer mode control register (WTM) to 1, the watch timer starts counting. By setting them to 0, the 5-bit counter is cleared and the watch timer stops counting.

Only the watch timer can be started form zero seconds by clearing WTM1 to 0 when the interval timer and watch timer operate at the same time. In this case, however, an error of up to $2^9 \times 1/f_W$ seconds may occur in the overflow (INTWT) after the zero-second start of the watch timer because the 9-bit prescaler is not cleared to 0.

8.4.2 Operation as interval timer

The interval timer is used to repeatedly generate an interrupt request at the interval specified by a count value set in advance.

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

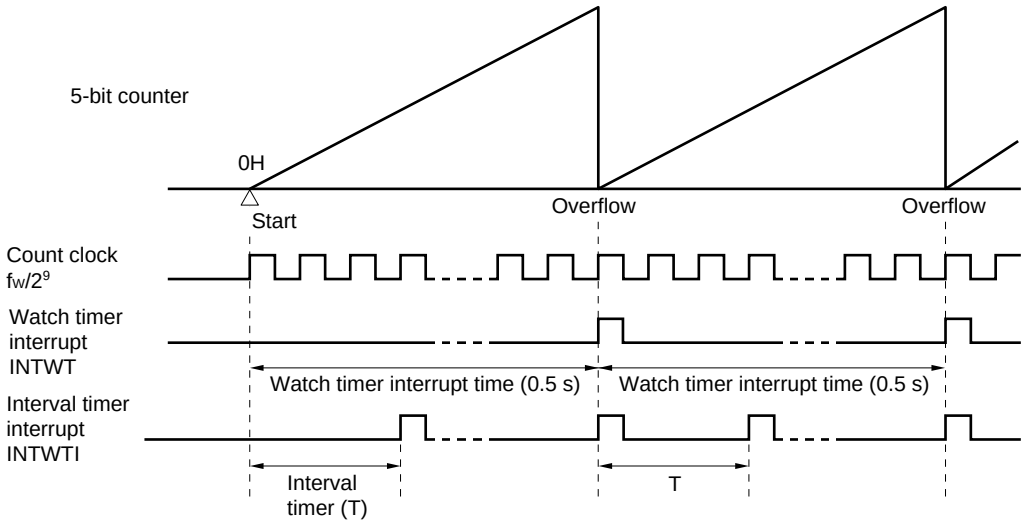
Table 8-3. Interval Generated Using the Interval Timer

WTM6	WTM5	WTM4	Interval	At f_X = 5.0 MHz	At f_X = 4.19 MHz
0	0	0	$2^4 \times 1/f_W$	409.6 μs	489 μs
0	0	1	$2^5 \times 1/f_W$	819.2 μs	978 μs
0	1	0	$2^6 \times 1/f_W$	1.64 ms	1.96 ms
0	1	1	$2^7 \times 1/f_W$	3.28 ms	3.91 ms
1	0	0	$2^8 \times 1/f_W$	6.55 ms	7.82 ms
1	0	1	$2^9 \times 1/f_W$	13.1 ms	15.6 ms
Other than above			Setting prohibited		

Remarks 1. f_X: Main system clock oscillation frequency

2. f_W: Watch timer clock frequency

Figure 8-3. Watch Timer/Interval Timer Operation Timing



Caution When operation of the watch timer and 5-bit counter operation is enabled by setting bit 0 (WTM0) of the watch mode timer mode control register (WTM) to 1, the interval until the first interrupt request (INTWT) is generated after the register is set does not exactly match the specification made with WTM3 (bit 3 of WTM). This is because there is a delay of one 9-bit prescaler output cycle until the 5-bit counter starts counting. Subsequently, however, the INTWT signal is generated at the specified intervals.

Remarks 1. fw: Watch timer clock frequency

[MEMO]

CHAPTER 9 WATCHDOG TIMER

9.1 Watchdog Timer Functions

The watchdog timer has the following functions:

- Watchdog timer
- Interval timer

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

(1) Watchdog timer

The watchdog timer is used to detect inadvertent program loops. When an inadvertent loop is detected, a non-maskable interrupt or a $\overline{\text{RESET}}$ signal can be generated.

Table 9-1. Inadvertent Loop Detection Time of Watchdog Timer

Inadvertent Loop Detection Time	At f_X = 5.0 MHz
$2^{11} \times 1/f_X$	410 μs
$2^{13} \times 1/f_X$	1.64 ms
$2^{15} \times 1/f_X$	6.55 ms
$2^{17} \times 1/f_X$	26.2 ms

f_X: Main system clock oscillation frequency

(2) Interval timer

The interval timer generates an interrupt at an arbitrary interval set in advance.

Table 9-2. Interval Time

Interval	At f_X = 5.0 MHz
$2^{11} \times 1/f_X$	410 μs
$2^{13} \times 1/f_X$	1.64 ms
$2^{15} \times 1/f_X$	6.55 ms
$2^{17} \times 1/f_X$	26.2 ms

f_X: Main system clock oscillation frequency

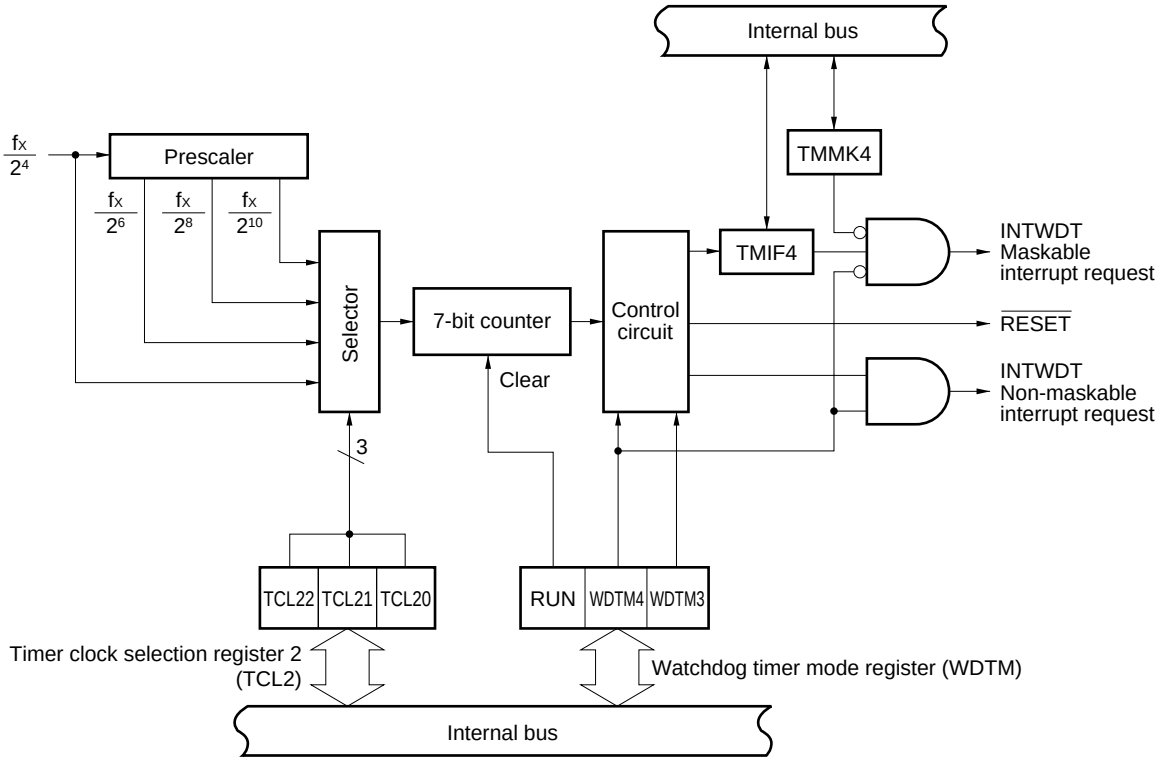
9.2 Watchdog Timer Configuration

The watchdog timer consists of the following hardware.

Table 9-3. Configuration of Watchdog Timer

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

Figure 9-1. Block Diagram of Watchdog Timer



9.3 Watchdog Timer Control Registers

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

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

(1) Timer clock selection register 2 (TCL2)

This register sets the watchdog timer count clock.

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

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

Figure 9-2. Format of Timer Clock Selection Register 2

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL2	0	0	0	0	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Watchdog timer count clock selection	Interval
0	0	0	$f_X/2^4$ (312.5 kHz)	$2^{11}/f_X$ (410 μs)
0	1	0	$f_X/2^6$ (78.1 kHz)	$2^{13}/f_X$ (1.64 ms)
1	0	0	$f_X/2^8$ (19.5 kHz)	$2^{15}/f_X$ (6.55 ms)
1	1	0	$f_X/2^{10}$ (4.88 kHz)	$2^{17}/f_X$ (26.2 ms)
Other than above			Setting prohibited	

Remarks 1. f_X : Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

(2) Watchdog timer mode register (WDTM)

This register sets an operation mode of the watchdog timer, and enables/disables counting of the watchdog timer.

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

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

Figure 9-3. Format of Watchdog Timer Mode Register

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 operation selection[Note 1]
0	Stops counting.
1	Clears counter and starts counting.

WDTM4	WDTM3	Watchdog timer operation mode selection[Note 2]
0	0	Operation stop
0	1	Interval timer mode (Generates a maskable interrupt upon overflow occurrence.)[Note 3]
1	0	Watchdog timer mode 1 (Generates a non-maskable interrupt upon overflow occurrence.)
1	1	Watchdog timer mode 2 (Starts reset operation upon overflow occurrence.)

Notes 1. Once RUN has been set to 1, it cannot be cleared to 0 by software. Therefore, when counting is started, it cannot be stopped by any means other than $\overline{\text{RESET}}$ input.

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

3. The watchdog timer starts operations as an interval timer when RUN is set to 1.

Cautions 1. When the watchdog timer is cleared by setting RUN to 1, the actual overflow time is up to 0.8% shorter than the time set by the timer clock selection register 2 (TCL2).

2. To set watchdog timer mode 1 or 2, set WDTM4 to 1 after confirming TMIF4 (bit 0 of the interrupt request flag register 0 (IF0)) being set to 0. When watchdog timer mode 1 or 2 is selected with TMIF4 set to 1, a non-maskable interrupt is generated upon the completion of rewriting WDTM4.

9.4 Watchdog Timer Operation

9.4.1 Operation as watchdog timer

The watchdog timer detects an inadvertent program loop when bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 1.

The count clock (inadvertent loop detection time interval) of the watchdog timer can be selected by bits 0 to 2 (TCL20 to TCL22) of timer clock selection register 2 (TCL2). By setting bit 7 (RUN) of WDTM to 1, the watchdog timer is started. Set RUN to 1 within the set inadvertent loop detection time interval after the watchdog timer has been started. By setting RUN to 1, the watchdog timer can be cleared and start counting. If RUN is not set to 1, and the inadvertent loop detection time is exceeded, a system reset signal or a non-maskable interrupt is generated, depending on the value of bit 3 (WDTM3) of WDTM.

The watchdog timer continues operation in HALT mode, but stops in STOP mode. Therefore, first set RUN to 1 to clear the watchdog timer before executing the STOP instruction.

Cautions 1. The actual inadvertent loop detection time may be up to 0.8% shorter than the set time.

2. When the subsystem clock is selected as the CPU clock, watchdog timer count operation is stopped. Even when the main system clock continues oscillating in this case, watchdog timer count operation is stopped.

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

TCL22	TCL21	TCL20	Inadvertent Loop Detection Time	At f_X = 5.0 MHz
0	0	0	$2^{11} \times 1/f_X$	410 μs
0	1	0	$2^{13} \times 1/f_X$	1.64 ms
1	0	0	$2^{15} \times 1/f_X$	6.55 ms
1	1	0	$2^{17} \times 1/f_X$	26.2 ms

f_X: Main system clock oscillation frequency

9.4.2 Operation as interval timer

When bits 4 and 3 (WDTM4, WDTM3) of watchdog timer mode register (WDTM) are set to 0 and 1, respectively, the watchdog timer operates as an interval timer that repeatedly generates an interrupt at intervals specified by a count value set in advance.

Select a count clock (or interval) by setting bits 0 to 2 (TCL20 to TCL22) of timer clock selection register 2 (TCL2). The watchdog timer starts operation as an interval timer when the RUN bit (bit 7 of WDTM) is set to 1.

In interval timer mode, the interrupt mask flag (TMMK4) is valid, and a maskable interrupt (INTWDT) can be generated. The priority of INTWDT is set as the highest of all the maskable interrupts.

The interval timer continues operation in HALT mode, but stops in STOP mode. Therefore, first set RUN to 1 to clear the interval timer before executing the STOP instruction.

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

2. The interval time may be up to 0.8% shorter than the set time when WDTM has just been set.

Table 9-5. Interval Time of Interval Timer

TCL22	TCL21	TCL20	Interval	At f_X = 5.0 MHz
0	0	0	$2^{11} \times 1/f_X$	410 μs
0	1	0	$2^{13} \times 1/f_X$	1.64 ms
1	0	0	$2^{15} \times 1/f_X$	6.55 ms
1	1	0	$2^{17} \times 1/f_X$	26.2 ms

f_X: Main system clock oscillation frequency

CHAPTER 10 8-BIT A/D CONVERTER

10.1 8-Bit A/D Converter Functions

The 8-bit A/D converter is an 8-bit resolution converter to convert an analog input to digital signals. This converter can control eight channels (ANI0 to ANI3) of analog inputs.

A/D conversion can be started only by software.

One of analog inputs ANI0 to ANI3 is selected for A/D conversion. A/D conversion is performed repeatedly, with an interrupt request (INTAD0) being issued each time A/D conversion is completed.

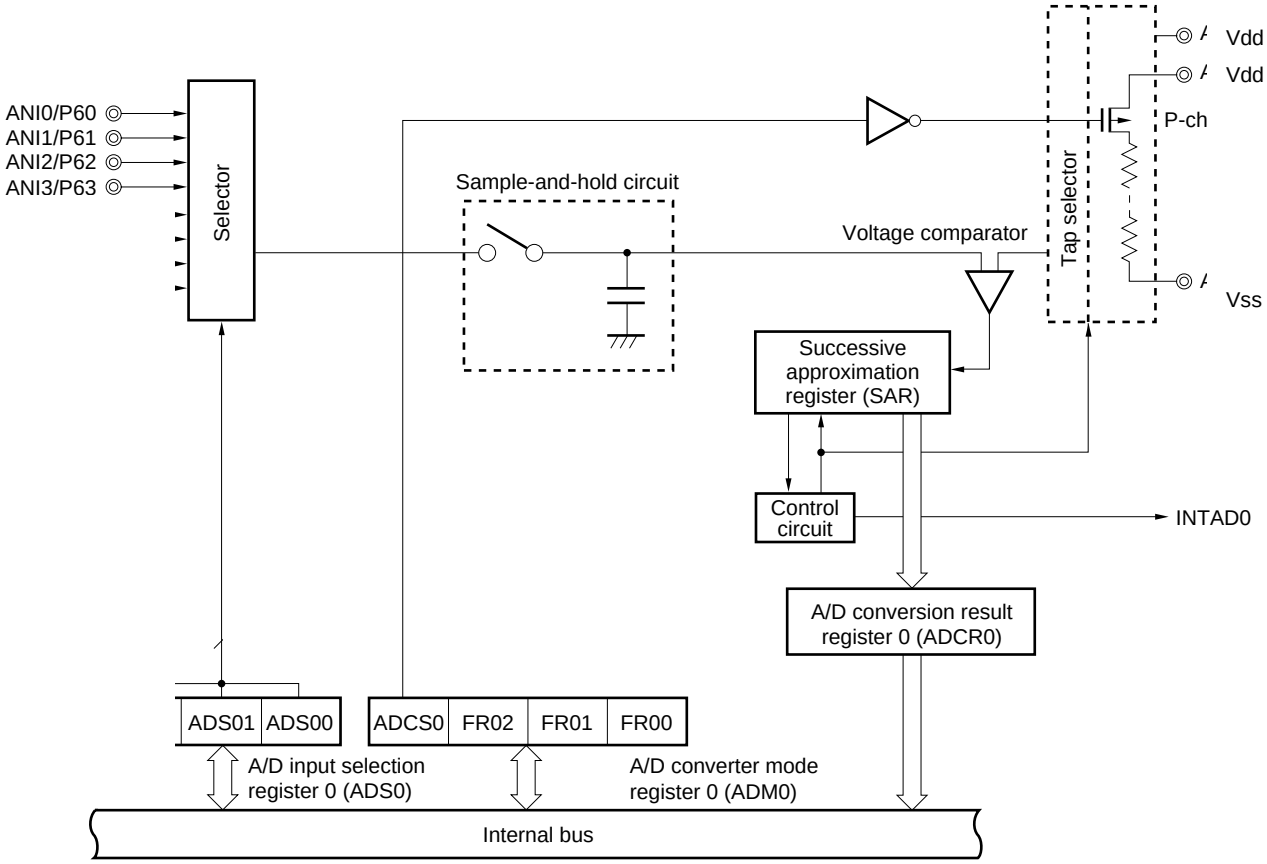
10.2 8-Bit A/D Converter Configuration

The 8-bit A/D converter consists of the following hardware.

Table 10-1. Configuration of 8-Bit A/D Converter

Item	Configuration
Analog input	4 channels (ANI0 to ANI3)
Register	Successive approximation register (SAR) A/D conversion result register 0 (ADCR0)
Control register	A/D converter mode register 0 (ADM0) A/D input selection register 0 (ADS0)

Figure 10-1. Block Diagram of 8-Bit A/D Converter



(1) Successive approximation register (SAR)

SAR receives the result of comparing an analog input voltage and a voltage at a voltage tap (comparison voltage), received from the serial resistor string, starting from the most significant bit (MSB).

Upon receiving all the bits, down to the least significant bit (LSB), that is, upon the completion of A/D conversion, SAR sends its contents to A/D conversion result register 0 (ADCR0).

(2) A/D conversion result register 0 (ADCR0)

ADCR0 holds the result of A/D conversion. Each time A/D conversion ends, the conversion result in the successive approximation register is loaded into ADCR0, which is an 8-bit register.

ADCR0 can be read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes this register undefined.

(3) Sample-and-hold circuit

The sample-and-hold circuit samples consecutive analog inputs from the input circuit, one by one, and sends them to the voltage comparator. The sampled analog input voltage is held during A/D conversion.

(4) Voltage comparator

The voltage comparator compares an analog input with the voltage output by the serial resistor string.

(6) ANI0 to ANI3

Pins ANI0 to ANI3 are the 8-channel analog input pins for the A/D converter. They are used to receive the analog signals for A/D conversion.

Caution Do not supply pins ANI0 to ANI3 with voltages that fall outside the rated range.

10.3 8-Bit A/D Converter Control Registers

The following two registers are used to control the 8-bit A/D converter.

- A/D converter mode register 0 (ADM0)
- A/D input selection register 0 (ADS0)

(1) A/D converter mode register 0 (ADM0)

ADM0 specifies the conversion time for analog inputs. It also specifies whether to enable conversion.

ADM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ADM0 to 00H.

Figure 10-2. Format of A/D Converter Mode Register 0

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
ADM0	ADCS0	0	FR02	FR01	FR00	0	0	0	FF80H	00H	R/W

ADCS0	A/D conversion control
0	Conversion disabled
1	Conversion enabled

FR02	FR01	FR00	A/D conversion time selection[Note 1]
0	0	0	144/fx (28.8 μs)
0	0	1	120/fx (24 μs)
0	1	0	96/fx (19.2 μs)
1	0	0	72/fx (14.4 μs)
1	0	1	60/fx (Setting prohibited[Note 2])
1	1	0	48/fx (Setting prohibited[Note 2])
Other than above			Setting prohibited

Notes 1. The specifications of FR02, FR01, and FR00 must be such that the A/D conversion time is at least 14 μs.

2. These bit combinations must not be used, as the A/D conversion time will fall below 14 μs.

Cautions 1. The result of conversion performed immediately after bit 7 (ADCS0) is set is undefined.

2. The conversion result may be undefined after ADCS0 has been cleared to 0 (For details, see 12.5 (5) Timing of undefined A/D conversion result).

Remarks 1. fx: Main system clock oscillation frequency

2. The parenthesized values apply to operation at fx = 5.0 MHz.

(2) A/D input selection register 0 (ADS0)

ADS0 specifies the port used to input the analog voltage to be converted to a digital signal.

ADS0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ADS0 to 00H.

Figure 10-3. Format of A/D Input Selection Register 0

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
ADS0	0	0	0	0	0	0	ADS01	ADS00	FF84H	00H	R/W

ADS01	ADS00	Analog input channel specification
0	0	ANI0
0	1	ANI1
1	0	ANI2
1	1	ANI3

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

10.4 8-Bit A/D Converter Operation

10.4.1 Basic operation of 8-bit A/D converter

<1> Select a channel for A/D conversion, using A/D input selection register 0 (ADS0).

<2> The voltage supplied to the selected analog input channel is sampled using the sample and hold circuit.

<3> After sampling continues for a certain period of time, the sample and hold circuit is put on hold to keep the input analog voltage until A/D conversion is completed.

<4> Bit 7 of the successive approximation register (SAR) is set. The series resistor string tap voltage at the tap selector is set to half of Vdd.

<5> The series resistor string tap voltage is compared with the analog input voltage using the voltage comparator. If the analog input voltage is higher than half of Vdd, the MSB of SAR is left set. If it is lower than half of Vdd, the MSB is reset.

<6> Bit 6 of SAR is set automatically, and comparison shifts to the next stage. The next tap voltage of the series resistor string is selected according to bit 7, which reflects the previous comparison result, as follows:

- Bit 7 = 1: Three quarters of Vdd
- Bit 7 = 0: One quarter of Vdd

The tap voltage is compared with the analog input voltage. Bit 6 is set or reset according to the result of comparison.

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

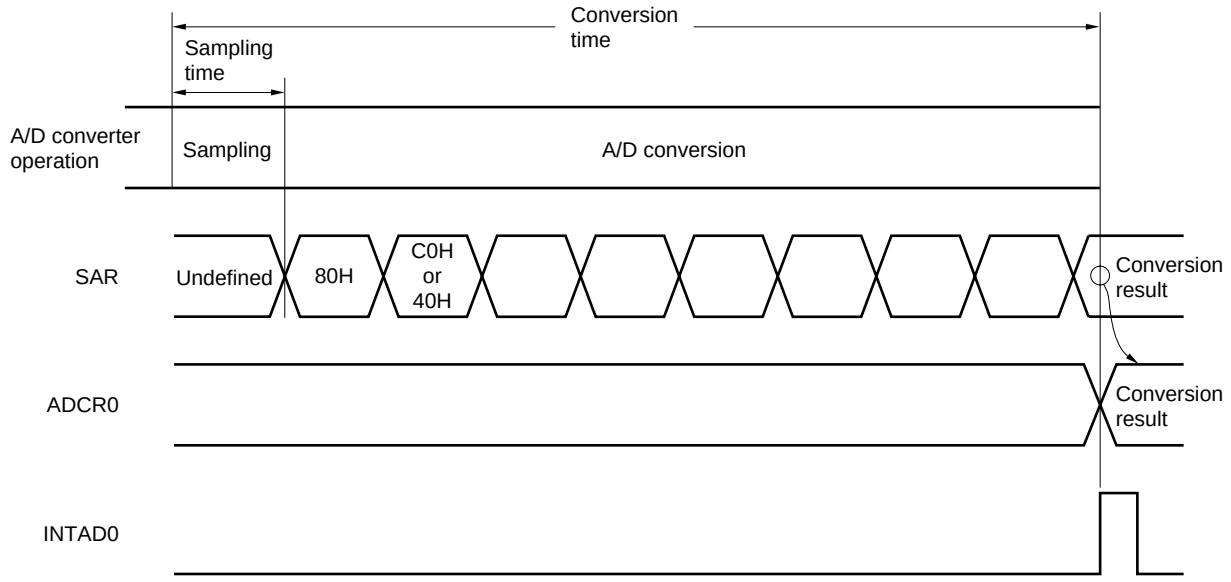
<7> Comparison is repeated until bit 0 of SAR is reached.

<8> When comparison is completed for all of the 8 bits, a significant digital result is left in SAR. This value is sent to and latched in A/D conversion result register 0 (ADCR0). At the same time, it is possible to generate an A/D conversion end interrupt request (INTAD0).

Cautions 1. The first A/D conversion value immediately after A/D conversion has been started may be undefined.

2. In standby mode, A/D converter operation is stopped.

Figure 10-4. Basic Operation of 8-Bit A/D Converter



A/D conversion continues until bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) is reset to 0 by software.

If an attempt is made to write to ADM0 or A/D input selection register 0 (ADS0) during A/D conversion, the ongoing A/D conversion is canceled. In this case, A/D conversion is restarted from the beginning, if ADCS0 is set to 1.

$\overline{\text{RESET}}$ input makes A/D conversion result register 0 (ADCR0) undefined.

10.4.2 Input voltage and conversion result

The relationships between the analog input voltage at the analog input pins (ANI0 to ANI3) and the A/D conversion result (A/D conversion result register 0 (ADCR0)) are represented by:

$$ADCR0 = INT\left(\frac{V_{IN}}{Vdd} \times 256 + 0.5\right)$$

or

$$(ADCR0 - 0.5) \times \frac{Vdd}{256} \leq V_{IN} < (ADCR0 + 0.5) \times \frac{Vdd}{256}$$

INT( ): Function that returns the integer part of a parenthesized value

V_{IN}: Analog input voltage

ADCR0: Value in A/D conversion result register 0 (ADCR0)

10.4.3 Operation mode of 8-bit A/D converter

The A/D converter is initially in select mode. In this mode, A/D input selection register 0 (ADS0) is used to select an analog input channel from ANI0 to ANI3 for A/D conversion.

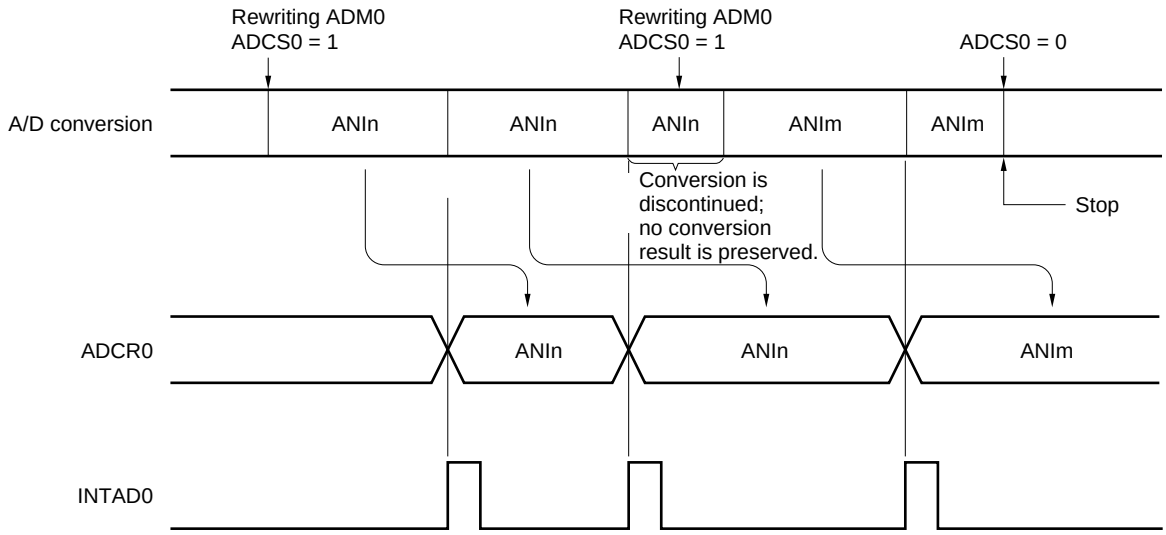
A/D conversion can be started only by software, that is, by setting A/D converter mode register 0 (ADM0).

The A/D conversion result is saved to A/D conversion result register 0 (ADCR0). At the same time, an interrupt request signal (INTAD0) is generated.

- **Software-started A/D conversion**

  Setting bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) to 1 triggers A/D conversion for a voltage applied to the analog input pin specified in A/D input selection register 0 (ADS0). Upon completion of A/D conversion, the conversion result is saved to A/D conversion result register 0 (ADCR0). At the same time, an interrupt request signal (INTAD0) is generated. Once A/D conversion is activated, and completed, another session of A/D conversion is started. A/D conversion is repeated until new data is written to ADM0. If data where ADCS0 is 1 is written to ADM0 again during A/D conversion, the ongoing session of A/D conversion is discontinued, and a new session of A/D conversion begins for the new data. If data where ADCS0 is 0 is written to ADM0 again during A/D conversion, A/D conversion is stopped immediately.

Figure 10-6. Software-Started A/D Conversion



Remarks 1. n = 0 to 3

2. m = 0 to 3

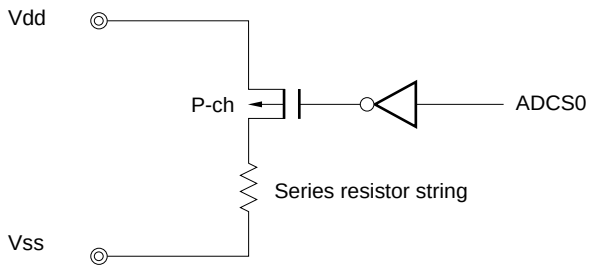
10.5 Cautions Related to 8-Bit A/D Converter

(1) Current drain in standby mode

In standby mode, the A/D converter stops its operation. Stopping conversion (bit 7 (ADCS0) of A/D converter mode register 0 (ADM0) = 0) can reduce the current drain.

Figure 12-7 shows how to reduce the current drain in standby mode.

Figure 10-7. How to Reduce Current Drain in Standby Mode



(2) Input range for pins ANI0 to ANI3

Be sure to keep the input voltage at ANI0 to ANI3 within its rating. If a voltage not lower than Vdd or not higher than V_{SS} (even within the absolute maximum rating) is input into a conversion channel, the conversion output of the channel becomes undefined. It may affect the conversion output of the other channels.

(3) Conflict

<1> Conflict between writing to A/D conversion result register 0 (ADCR0) at the end of conversion and reading from ADCR0 using instruction

Reading from ADCR0 takes precedence. After reading, the new conversion result is written to ADCR0.

<2> Conflict between writing to ADCR0 at the end of conversion and writing to A/D converter mode register 0 (ADM0) or A/D input selection register 0 (ADS0)

Writing to ADM0 or ADS0 takes precedence. ADCR0 is not written to. No A/D conversion end interrupt request signal (INTAD0) is generated.

(4) Conversion result immediately after start of A/D conversion

The first A/D conversion value immediately after A/D conversion has been started is undefined. Poll the A/D conversion end interrupt request (INTAD0) and drop the first conversion result.

(5) Timing of undefined A/D conversion result

The A/D conversion value may become undefined if the timing of the completion of A/D conversion and that to stop the A/D conversion operation conflict. Therefore, read the A/D conversion result while the A/D conversion operation is in progress. To read the A/D conversion result after the A/D conversion operation has been stopped, stop the A/D conversion operation before the next conversion operation is completed. Figures 10-8 and 10-9 show the timing at which the conversion result is read.

Figure 10-8. Conversion Result Read Timing (If Conversion Result is Undefined)

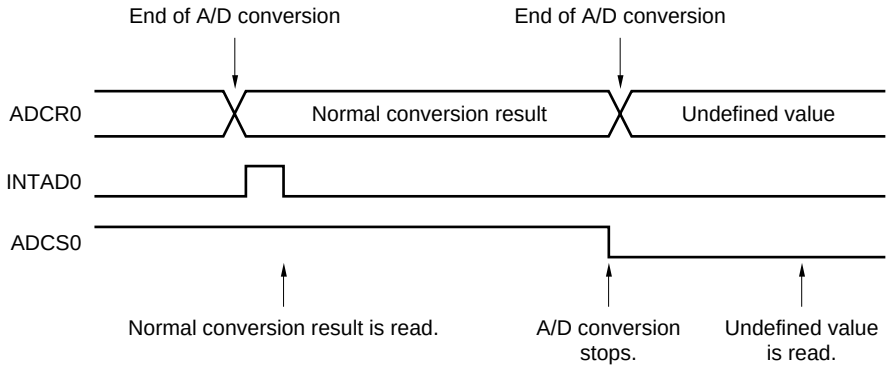
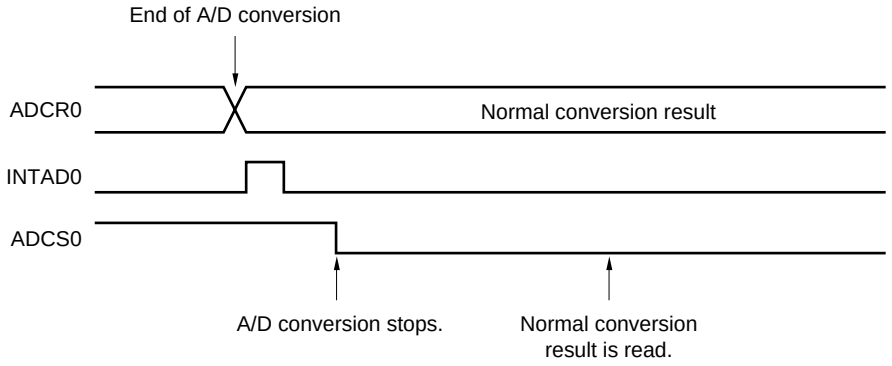


Figure 10-9. Conversion Result Read Timing (If Conversion Result is Normal)



(7) ANI0 to ANI3

The analog input pins (ANI0 to ANI3) are alternate-function pins. They are used also as port pins (P60 to P63).

If any of ANI0 to ANI3 has been selected for A/D conversion, do not execute input instructions for the ports. Otherwise, the conversion resolution may become lower.

If a digital pulse is applied to a pin adjacent to the analog input pins during A/D conversion, coupling noise may occur which prevents an A/D conversion result from being obtained as expected. Avoid applying a digital pulse to pins adjacent to the analog input pins during A/D conversion.

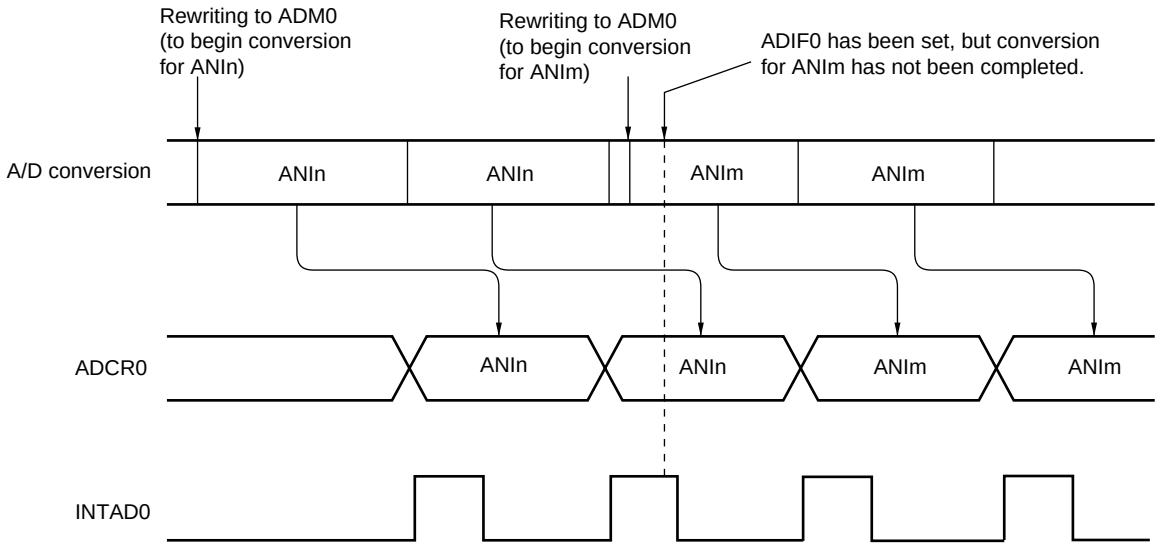
(9) Interrupt request flag (ADIF0)

Changing the content of A/D converter mode register 0 (ADM0) does not clear the interrupt request flag (ADIF0).

If the analog input pins are changed during A/D conversion, therefore, the A/D conversion result and the conversion end interrupt request flag may reflect the previous analog input immediately before writing to ADM0 occurs. In this case, ADIF0 may already be set if it is read-accessed immediately after ADM0 is write-accessed, even when A/D conversion has not been completed for the new analog input.

In addition, when A/D conversion is restarted, ADIF0 must be cleared beforehand.

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



Remarks 1. n = 0 to 3

2. m = 0 to 3

[MEMO]

CHAPTER 11 SERIAL INTERFACE 20

11.1 Serial Interface 20 Functions

Serial interface 20 has the following three modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode
- 3-wire serial I/O mode

(1) Operation stop mode

This mode is used when serial transfer is not performed. Power consumption is minimized in this mode.

(2) Asynchronous serial interface (UART) mode

This mode is used to send and receive the one byte of data that follows a start bit. It supports full-duplex communication.

Serial interface 20 contains an UART-dedicated baud rate generator, enabling communication over a wide range of baud rates. It is also possible to define baud rates by dividing the frequency of the clock input to the ASCK20 pin.

(3) 3-wire serial I/O mode (switchable between MSB-first and LSB-first transmission)

This mode is used to transmit 8-bit data, using three lines: a serial clock ($\overline{\text{SCK20}}$) line and two serial data lines (SI20 and SO20).

As it supports simultaneous transmission and reception, 3-wire serial I/O mode requires less processing time for data transmission than asynchronous serial interface mode.

Because, in 3-wire serial I/O mode, it is possible to select whether 8-bit data transmission begins with the MSB or LSB, serial interface 20 can be connected to any device regardless of whether that device is designed for MSB-first or LSB-first transmission.

3-wire serial I/O mode is useful for connecting peripheral I/O circuits and display controllers having conventional synchronous serial interfaces, such as those of the 75XL, 78K, and 17K Series devices.

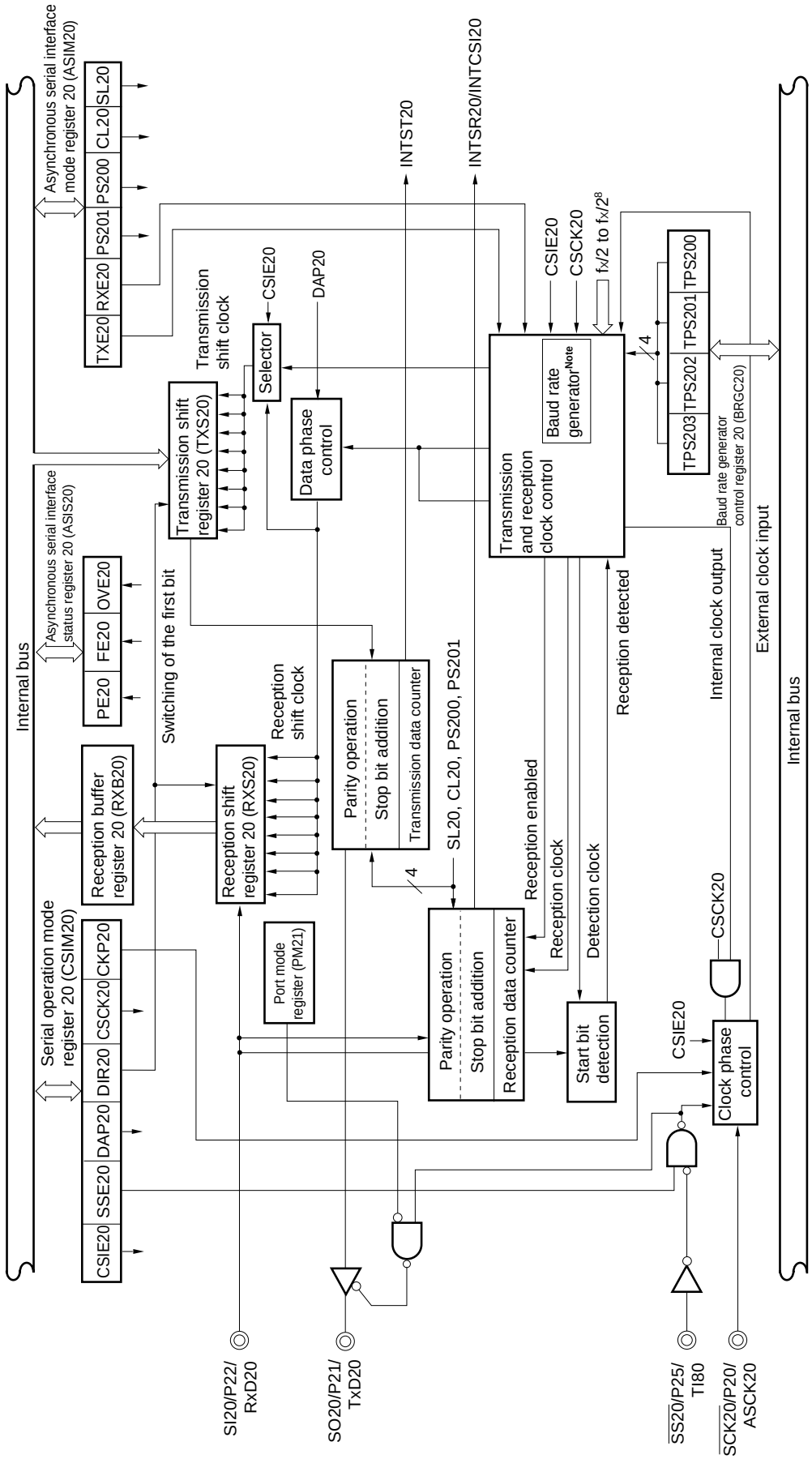
11.2 Serial Interface 20 Configuration

Serial interface 20 consists of the following hardware.

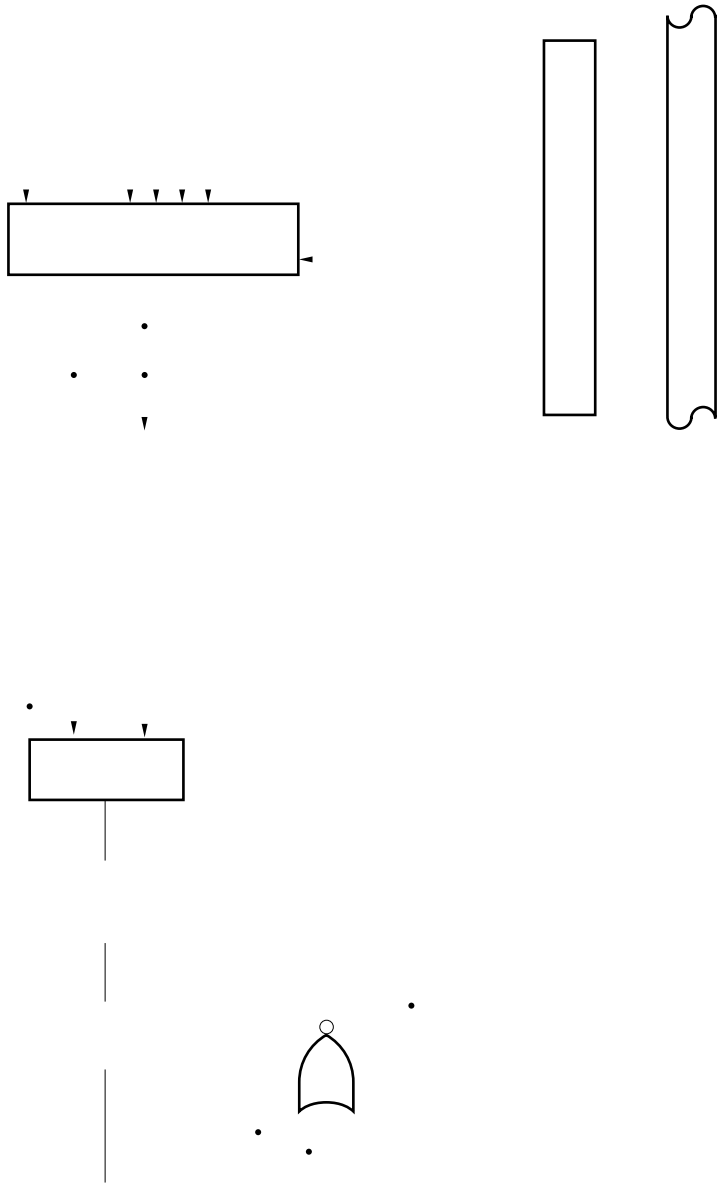
Table 11-1. Configuration of Serial Interface 20

Item	Configuration
Register	Transmission shift register 20 (TXS20) Reception shift register 20 (RXS20) Reception buffer register 20 (RXB20)
Control register	Serial operation mode register 20 (CSIM20) Asynchronous serial interface mode register 20 (ASIM20) Asynchronous serial interface status register 20 (ASIS20) Baud rate generator control register 20 (BRGC20)

Figure 14-1. Block Diagram of Serial Interface 20



Note See **Figure 14-2** for the configuration of the baud rate generator.



(1) Transmission shift register 20 (TXS20)

TXS20 is a register in which transmission data is prepared. The transmission data is output from TXS20 bit-serially.

When the data length is seven bits, bits 0 to 6 of the data in TXS20 will be transmission data. Writing data to TXS20 triggers transmission.

TXS20 can be written with an 8-bit memory manipulation instruction, but cannot be read.

$\overline{\text{RESET}}$ input sets TXS20 to FFH.

Caution Do not write to TXS20 during transmission.

TXS20 and reception buffer register 20 (RXB20) are mapped at the same address, such that any attempt to read from TXS20 results in a value being read from RXB20.

(2) Reception shift register 20 (RXS20)

RXS20 is a register in which serial data, received at the RxD20 pin, is converted to parallel data. Once one entire byte has been received, RXS20 feeds the reception data to reception buffer register 20 (RXB20).

RXS20 cannot be manipulated directly by a program.

(3) Reception buffer register 20 (RXB20)

RXB20 holds a reception data. A new reception data is transferred from reception shift register 20 (RXS20) every 1-byte data reception.

When the data length is seven bits, the reception data is sent to bits 0 to 6 of RXB20, in which the MSB is always fixed to 0.

RXB20 can be read with an 8-bit memory manipulation instruction, but cannot be written.

$\overline{\text{RESET}}$ input makes RXB20 undefined.

Caution RXB20 and transmission shift register 20 (TXS20) are mapped at the same address, such that any attempt to write to RXB20 results in a value being written to TXS20.

(4) Transmission control circuit

The transmission control circuit controls transmission. For example, it adds start, parity, and stop bits to the data in transmission shift register 20 (TXS20), according to the setting of asynchronous serial interface mode register 20 (ASIM20).

(5) Reception control circuit

The reception control circuit controls reception according to the setting of asynchronous serial interface mode register 20 (ASIM20). It also checks for errors, such as parity errors, during reception. If an error is detected, asynchronous serial interface status register 20 (ASIS20) is set according to the status of the error.

11.3 Serial Interface 20 Control Registers

Serial interface 20 is controlled by the following registers.

- Serial operation mode register 20 (CSIM20)
- Asynchronous serial interface mode register 20 (ASIM20)
- Asynchronous serial interface status register 20 (ASIS20)
- Baud rate generator control register 20 (BRGC20)

(1) Serial operation mode register 20 (CSIM20)

CSIM20 is used to make the settings related to 3-wire serial I/O mode.
CSIM20 is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears CSIM20 to 00H.

Figure 11-3. Format of Serial Operation Mode Register 20

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM20	CSIE20	SSE20	0	0	DAP20	DIR20	CSCK20	CKP20	FF72H	00H	R/W

CSIE20	3-wire serial I/O mode operation control
0	Operation disabled
1	Operation enabled

SSE20	$\overline{\text{SS20}}$-pin selection	Function of $\overline{\text{SS20}}$/P23 pin	Communication status
0	Not used	Port function	Communication enabled
1	Used	0	Communication enabled
		1	Communication disabled

DAP20	3-wire serial I/O mode data phase selection
0	Outputs at the falling edge of $\overline{\text{SCK20}}$.
1	Outputs at the rising edge of $\overline{\text{SCK20}}$.

DIR20	First-bit specification
0	MSB
1	LSB

CSCK20	3-wire serial I/O mode clock selection
0	External clock input to the $\overline{\text{SCK20}}$ pin
1	Output of the dedicated baud rate generator

CKP20	3-wire serial I/O mode clock phase selection
0	Clock is low active, and $\overline{\text{SCK20}}$ is at high level in the idle state.
1	Clock is high active, and $\overline{\text{SCK20}}$ is at low level in the idle state.

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

2. CSIM20 must be cleared to 00H, if UART mode is selected.

(2) Asynchronous serial interface mode register 20 (ASIM20)

ASIM20 is used to make the settings related to asynchronous serial interface mode.

ASIM20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIM20 to 00H.

Figure 11-4. Format of Asynchronous Serial Interface Mode Register 20

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM20	TXE20	RXE20	PS201	PS200	CL20	SL20	0	0	FF70H	00H	R/W

TXE20	Transmit operation control
0	Transmit operation stop
1	Transmit operation enable

RXE20	Receive operation control
0	Receive operation stop
1	Receive operation enable

PS201	PS200	Parity bit specification
0	0	No parity
0	1	Always add 0 parity at transmission. Parity check is not performed at reception (No parity error is generated).
1	0	Odd parity
1	1	Even parity

CL20	Transmit data character length specification
0	7 bits
1	8 bits

SL20	Transmit data stop bit length
0	1 bit
1	2 bits

Cautions 1. Bits 0 and 1 must all be set to 0.

2. If 3-wire serial I/O mode is selected, ASIM20 must be cleared to 00H.

3. Switch operating modes after halting serial transmit/receive operation.

Table 11-2. Serial Interface 20 Operating Mode Settings

(1) Operation stop mode

ASIM20		CSIM20			PM22	P22	PM21	P21	PM20	P20	First Bit	Shift Clock	P22/SI20/ RxD20 Pin Function	P21/SO20/ TxD20 Pin Function	P20/$\overline{SCK20}$/ ASCK20 Pin Function
TXE20	RXE20	CSIE20	DIR20	CSCK20											
0	0	0	×	×	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	–	–	P22	P21	P20
Other than above											Setting prohibited				

(2) 3-wire serial I/O mode

ASIM20		CSIM20			PM22	P22	PM21	P21	PM20	P20	First Bit	Shift Clock	P22/SI20/ RxD20 Pin Function	P21/SO20/ TxD20 Pin Function	P20/$\overline{SCK20}$/ ASCK20 Pin Function
TXE20	RXE20	CSIE20	DIR20	CSCK20											
0	0	1	0	0	×Note 2	×Note 2	0	1	1	×	MSB	External clock	SI20Note 2	SO20 (CMOS output)	$\overline{SCK20}$ input
				1					0	1		Internal clock			$\overline{SCK20}$ output
		1	1	0					1	×	LSB	External clock			$\overline{SCK20}$ input
				1					0	1		Internal clock			$\overline{SCK20}$ output
Other than above											Setting prohibited				

(3) Asynchronous serial interface mode

ASIM20		CSIM20			PM22	P22	PM21	P21	PM20	P20	First Bit	Shift Clock	P22/SI20/ RxD20 Pin Function	P21/SO20/ TxD20 Pin Function	P20/$\overline{SCK20}$/ ASCK20 Pin Function
TXE20	RXE20	CSIE20	DIR20	CSCK20											
1	0	0	0	0	×Note 1	×Note 1	0	1	1	×	LSB	External clock	P22	TxD20 (CMOS output)	ASCK20 input
									×Note 1	×Note 1		Internal clock			P20
0	1	0	0	0	1	×	×Note 1	×Note 1	1	×		External clock	RxD20	P21	ASCK20 input
									×Note 1	×Note 1		Internal clock			P20
1	1	0	0	0	1	×	0	1	1	×		External clock		TxD20 (CMOS output)	ASCK20 input
									×Note 1	×Note 1		Internal clock			P20
Other than above											Setting prohibited				

Notes 1. These pins can be used for port functions.

2. When only transmission is used, this pin can be used as P22 (CMOS input/output).

Remark ×: Don't care.

(3) Asynchronous serial interface status register 20 (ASIS20)

ASIS20 indicates the type of a reception error, if it occurs while asynchronous serial interface mode is set.

ASIS20 is set with a 1-bit or 8-bit memory manipulation instruction.

The contents of ASIS20 are undefined in 3-wire serial I/O mode.

$\overline{\text{RESET}}$ input clears ASIS20 to 00H.

Figure 11-5. Format of Asynchronous Serial Interface Status Register 20

PE20	Parity error flag
0	
1	

Notes 1. Even when the stop bit length is set to 2 bits by setting bit 2 (SL20) of asynchronous serial interface mode register 20 (ASIM20), the stop bit detection at reception is performed with 1 bit.

2. Be sure to read reception buffer register 20 (RXB20) when an overrun error occurs. If not, every time the data is received an overrun error is generated.

(4) Baud rate generator control register 20 (BRGC20)

BRGC20 is used to specify the serial clock for serial interface 20.

BRGC20 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears BRGC20 to 00H.

Figure 11-6. Format of Baud Rate Generator Control Register 20

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BRGC20	TPS203	TPS202	TPS201	TPS200	0	0	0	0	FF73H	00H	R/W

TPS203	TPS202	TPS201	TPS200	3-bit counter source clock selection	n
0	0	0	0	$f_X/2$ (2.5 MHz)	1
0	0	0	1	$f_X/2^2$ (1.25 MHz)	2
0	0	1	0	$f_X/2^3$ (625 kHz)	3
0	0	1	1	$f_X/2^4$ (313 kHz)	4
0	1	0	0	$f_X/2^5$ (156 kHz)	5
0	1	0	1	$f_X/2^6$ (78.1 kHz)	6
0	1	1	0	$f_X/2^7$ (39.1 kHz)	7
0	1	1	1	$f_X/2^8$ (19.5 kHz)	8
1	0	0	0	External clock input to the ASCK20 pin**Note**	–
Other than above				Setting prohibited	

Note An external clock can be used only in UART mode.

Cautions 1. When writing to BRGC00 is performed during a communication operation, the output of baud rate generator is disrupted and communications cannot be performed normally. Be sure not to write to BRGC00 during communication operations.

2. Be sure not to select n = 1 during operation at f_X = 5.0 MHz because the resulting baud rate exceeds the rated range.

3. When the external input clock is selected, set port mode register 2 (PM2) to input mode.

Remarks 1. f_X: Main system clock oscillation frequency

2. n: Values determined by the settings of TPS200 to TPS203 ($1 \leq n \leq 8$)

3. The parenthesized values apply to operation at f_X = 5.0 MHz.

The baud rate transmit/receive clock to be generated is either a signal scaled from the system clock, or a signal scaled from the clock input to the ASCK20 pin.

(a) Generation of baud rate transmit/receive clock form system clock

The transmit/receive clock is generated by scaling the system clock. The baud rate of a clock generated from the system clock is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_X}{2^{n+1} \times 8} \text{ [Hz]}$$

f_X: Main system clock oscillation frequency

n: Values in Figure 14-6, determined by the values of TPS200 to TPS203 ($2 \leq n \leq 8$)

Table 11-3. Example of Relationships between System Clock and Baud Rate

Baud Rate (bps)	n	BRGC20 Set Value	Error (%)	
			f_X = 5.0 MHz	f_X = 4.9152 MHz
1,200	8	70H	1.73	0
2,400	7	60H		
4,800	6	50H		
9,600	5	40H		
19,200	4	30H		
38,400	3	20H		
76,800	2	10H		

Caution Do not select n = 1 during operation at f_X = 5.0 MHz because the resulting baud rate exceeds the rated range.

(b) Generation of baud rate transmit/receive clock from external clock input to ASCK20 pin

The transmit/receive clock is generated by scaling the clock input from the ASCK20 pin. The baud rate of a clock generated from the clock input to the ASCK20 pin is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_{ASCK}}{16} \text{ [Hz]}$$

f_{ASCK}: Frequency of clock input to the ASCK20 pin

Table 11-4. Relationship between ASCK20 Pin Input Frequency and Baud Rate (When BRGC20 is Set to 80H)

Baud Rate (bps)	ASCK20 Pin Input Frequency (kHz)
75	1.2
150	2.4
300	4.8
600	9.6
1,200	19.2
2,400	38.4
4,800	76.8
9,600	153.6
19,200	307.2
31,250	500.0
38,400	614.4

11.4 Serial Interface 20 Operation

Serial interface 20 provides the following three types of modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode
- 3-wire serial I/O mode

11.4.1 Operation stop mode

In operation stop mode, serial transfer is not executed; therefore, the power consumption can be reduced. The P20/$\overline{\text{SCK20}}$/ASCK20, P21/SO20/TxD20, and P22/SI20/RxD20 pins can be used as normal I/O ports.

(1) Register setting

Operation stop mode is set by serial operation mode register 20 (CSIM20) and asynchronous serial interface mode register 20 (ASIM20).

(a) Serial operation mode register 20 (CSIM20)

CSIM20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM20 to 00H.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM20	CSIE20	SSE20	0	0	DAP20	DIR20	CSCK20	CKP20	FF72H	00H	R/W

CSIE20	Operation control in 3-wire serial I/O mode
0	Operation disable
1	Operation enable

Caution Bits 4 and 5 must all be set to 0.

(b) Asynchronous serial interface mode register 20 (ASIM20)

ASIM20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIM20 to 00H.

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM20	TXE20	RXE20	PS201	PS200	CL20	SL20	0	0	FF70H	00H	R/W

TXE20	Transmit operation control
0	Transmit operation stop
1	Transmit operation enable

RXE20	Receive operation control
0	Receive operation stop
1	Receive operation enable

Caution Bits 0 and 1 must all be set to 0.

11.4.2 Asynchronous serial interface (UART) mode

In this mode, the one-byte data following the start bit is transmitted/received and thus full-duplex communication is possible.

This device incorporates an UART-dedicated baud rate generator that enables communications at a desired baud rate from many options. In addition, the baud rate can also be defined by dividing the clock input to the ASCK20 pin.

The UART-dedicated baud rate generator also can output the 31.25-kbps baud rate that complies with the MIDI standard.

(1) Register setting

UART mode is set by serial operation mode register 20 (CSIM20), asynchronous serial interface mode register 20 (ASIM20), asynchronous serial interface status register 20 (ASIS20), and baud rate generator control register 20 (BRGC20).

(a) Serial operation mode register 20 (CSIM20)

CSIM20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM20 to 00H.

Set CSIM20 to 00H when UART mode is selected.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM20	CSIE20	SSE20	0	0	DAP20	DIR20	CSCK20	CKP20	FF72H	00H	R/W

CSIE20	3-wire serial I/O mode operation control
0	Operation disabled
1	Operation enabled

SSE20	$\overline{\text{SS20}}$-pin selection	Function of $\overline{\text{SS20}}$/P25 pin	Communication status
0	Not used	Port function	Communication enabled
1	Used	0	Communication enabled
		1	Communication disabled

DAP20	3-wire serial I/O mode data phase selection
0	Outputs at the falling edge of $\overline{\text{SCK20}}$.
1	Outputs at the rising edge of $\overline{\text{SCK20}}$.

DIR20	First-bit specification
0	MSB
1	LSB

CSCK20	3-wire serial I/O mode clock selection
0	External clock input to the $\overline{\text{SCK20}}$ pin
1	Output of the dedicated baud rate generator

CKP20	3-wire serial I/O mode clock phase selection
0	Clock is low active, and $\overline{\text{SCK20}}$ is high level in the idle state.
1	Clock is high active, and $\overline{\text{SCK20}}$ is low level in the idle state.

Caution Bits 4 and 5 must all be set to 0.

(b) Asynchronous serial interface mode register 20 (ASIM20)

ASIM20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIM20 to 00H.

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM20	TXE20	RXE20	PS201	PS200	CL20	SL20	0	0	FF70H	00H	R/W

TXE20	Transmit operation control
0	Transmit operation stop
1	Transmit operation enable

RXE20	Receive operation control
0	Receive operation stop
1	Receive operation enable

PS201	PS200	Parity bit specification
0	0	No parity
0	1	Always add 0 parity at transmission. Parity check is not performed at reception (No parity error is generated).
1	0	Odd parity
1	1	Even parity

CL20	Character length specification
0	7 bits
1	8 bits

SL20	Transmit data stop bit length specification
0	1 bit
1	2 bits

Cautions 1. Bits 0 and 1 must all be set to 0.

2. Switch operating modes after halting serial transmit/receive operation.

(c) Asynchronous serial interface status register 20 (ASIS20)

ASIS20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIS20 to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ASIS20	0	0	0	0	0	PE20	FE20	OVE20	FF71H	00H	R

PE20	Parity error flag
0	Parity error not generated
1	Parity error generated (when the parity of transmit data does not match)

FE20	Framing error flag
0	Framing error not generated
1	Framing error generated (when stop bit is not detected)Note 1

OVE20	Overrun error flag
0	Overrun error not generated
1	Overrun error generatedNote 2 (when the next receive operation is completed before data is read from reception buffer register 20)

Notes 1. Even when the stop bit length is set to 2 bits by setting bit 2 (SL20) of asynchronous serial interface mode register 20 (ASIM20), the stop bit detection at reception is performed with 1 bit.

2. Be sure to read reception buffer register 20 (RXB20) when an overrun error occurs. If not, every time the data is received an overrun error is generated.

(d) Baud rate generator control register 20 (BRGC20)

BRGC20 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears BRGC20 to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BRGC20	TPS203	TPS202	TPS201	TPS200	0	0	0	0	FF73H	00H	R/W

TPS203	TPS202	TPS201	TPS200	3-bit counter source clock selection	n
0	0	0	0	$f_X/2$ (2.5 MHz)	1
0	0	0	1	$f_X/2^2$ (1.25 MHz)	2
0	0	1	0	$f_X/2^3$ (625 kHz)	3
0	0	1	1	$f_X/2^4$ (313 kHz)	4
0	1	0	0	$f_X/2^5$ (156 kHz)	5
0	1	0	1	$f_X/2^6$ (78.1 kHz)	6
0	1	1	0	$f_X/2^7$ (39.1 kHz)	7
0	1	1	1	$f_X/2^8$ (19.5 kHz)	8
1	0	0	0	External clock input to ASCK20 pin	–
Other than above				Setting prohibited	

Cautions 1. When writing to BRGC20 is performed during a communication operation, the output of baud rate generator is disrupted and communications cannot be performed normally. Be sure not to write to BRGC20 during communication operation.

2. Be sure not to select n = 1 during an operation at f_X = 5.0 MHz because the resulting baud rate exceeds the rated range.

3. When the external input clock is selected, set port mode register 2 (PM2) to input mode.

Remarks 1. f_X: Main system clock oscillation frequency

2. n: Values determined by the settings of TPS200 to TPS203 ($1 \leq n \leq 8$)

3. The parenthesized values apply to operation at f_X = 5.0 MHz.

The baud rate transmit/receive clock to be generated is either a signal scaled from the system clock, or a signal scaled from the clock input to the ASCK20 pin.

(i) Generation of baud rate transmit/receive clock from system clock

The transmit/receive clock is generated by scaling the system clock. The baud rate of a clock generated from the system clock is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_X}{2^{n+1} \times 8} \text{ [Hz]}$$

f_X: Main system clock oscillation frequency

n: Values in the above table determined by the settings of TPS200 to TPS203 ($2 \leq n \leq 8$)

Table 11-5. Example of Relationships between System Clock and Baud Rate

Baud Rate (bps)	n	BRGC20 Set Value	Error (%)	
			f_X = 5.0 MHz	f_X = 4.9152 MHz
1,200	8	70H	1.73	0
2,400	7	60H		
4,800	6	50H		
9,600	5	40H		
19,200	4	30H		
38,400	3	20H		
76,800	2	10H		

Caution Do not select n = 1 during operation at f_X = 5.0 MHz because the resulting baud rate exceeds the rated range.

(ii) Generation of baud rate transmit/receive clock from external clock input to ASCK20 pin

The transmit/receive clock is generated by scaling the clock input from the ASCK20 pin. The baud rate of a clock generated from the clock input to the ASCK20 pin is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_{ASCK}}{16} \text{ [Hz]}$$

f_{ASCK}: Frequency of clock input to ASCK20 pin

Table 11-6. Relationship between ASCK20 Pin Input Frequency and Baud Rate (When BRGC20 is Set to 80H)

Baud Rate (bps)	ASCK20 Pin Input Frequency (kHz)
75	1.2
150	2.4
300	4.8
600	9.6
1,200	19.2
2,400	38.4
4,800	76.8
9,600	153.6
19,200	307.2
31,250	500.0
38,400	614.4

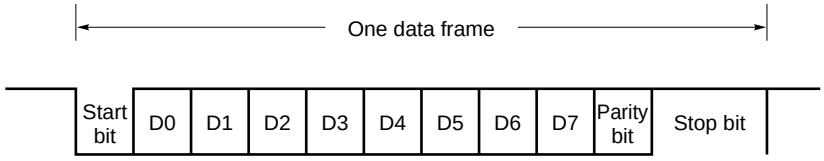
(2) Communication operation

(a) Data format

The transmit/receive data format is as shown in Figure 14-7. One data frame consists of a start bit, character bits, parity bit, and stop bit(s).

The specification of character bit length in one data frame, parity selection, and specification of stop bit length is carried out with asynchronous serial interface mode register 20 (ASIM20).

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



- Start bits ..... 1 bit
- Character bits ..... 7 bits/8 bits
- Parity bits ..... Even parity/odd parity/0 parity/no parity
- Stop bit(s) ..... 1 bit/2 bits

When 7 bits are selected as the number of character bits, only the lower 7 bits (bits 0 to 6) are valid; in transmission the most significant bit (bit 7) is ignored, and in reception the most significant bit (bit 7) is always "0".

The serial transfer rate is selected by ASIM20 and the baud rate generator control register 20 (BRGC20).

If a serial data receive error is generated, the receive error contents can be determined by reading the status of asynchronous serial interface status register 20 (ASIS20).

(b) Parity types and operation

The parity bit is used to detect a bit error in the communication data. Normally, the same kind of parity bit is used on the transmitting side and the receiving side. With even parity and odd parity, a one-bit (odd number) error can be detected. With 0 parity and no parity, an error cannot be detected.

(i) Even parity

- **At transmission**

  The parity bit is determined so that the number of bits with a value of "1" in the transmit data including the parity bit may be even. The parity bit value should be as follows.

  The number of bits with a value of "1" is an odd number in transmit data: 1
  The number of bits with a value of "1" is an even number in transmit data: 0

- **At reception**

  The number of bits with a value of "1" in the receive data including parity bit is counted, and if the number is odd, a parity error is generated.

(ii) Odd parity

- **At transmission**

  Conversely to the even parity, the parity bit is determined so that the number of bits with a value of "1" in the transmit data including parity bit may be odd. The parity bit value should be as follows.

  The number of bits with a value of "1" is an odd number in transmit data: 0
  The number of bits with a value of "1" is an even number in transmit data: 1

- **At reception**

  The number of bits with a value of "1" in the receive data including parity bit is counted, and if the number is even, a parity error is generated.

(iii) 0 parity

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

(iv) No parity

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

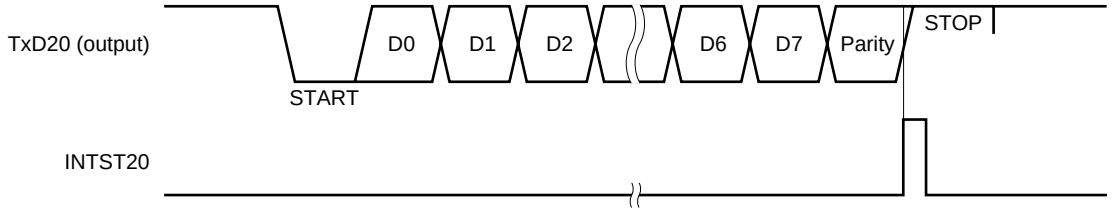
(c) Transmission

A transmit operation is started by writing transmit data to transmission shift register 20 (TXS20). The start bit, parity bit, and stop bit(s) are added automatically.

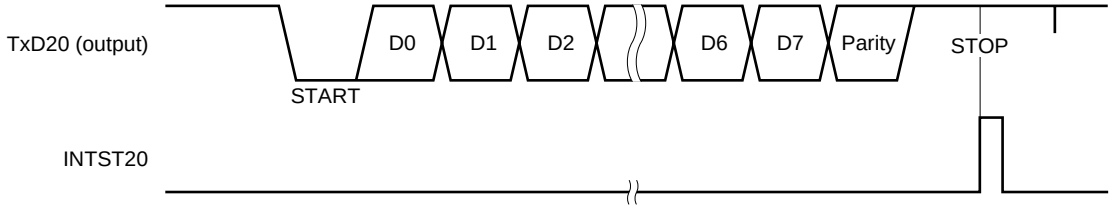
When the transmit operation starts, the data in TXS20 is shifted out, and when TXS20 is empty, a transmission completion interrupt (INTST20) is generated.

Figure 11-8. Asynchronous Serial Interface Transmission Completion Interrupt Timing

(a) Stop bit length: 1



(b) Stop bit length: 2



Caution Do not rewrite to asynchronous serial interface mode register 20 (ASIM20) during a transmit operation. If the ASIM20 register is rewritten to during transmission, subsequent transmission may not be performed (the normal state is restored by $\overline{\text{RESET}}$ input).

It is possible to determine whether transmission is in progress by software by using a transmission completion interrupt (INTST20) or the interrupt request flag (STIF20) set by INTST20.

(d) Reception

When bit 6 (RXE20) of asynchronous serial interface mode register 20 (ASIM20) is set to 1, a receive operation is enabled and sampling of the RxD20 pin input is performed.

RxD20 pin input sampling is performed using the serial clock specified by ASIM20.

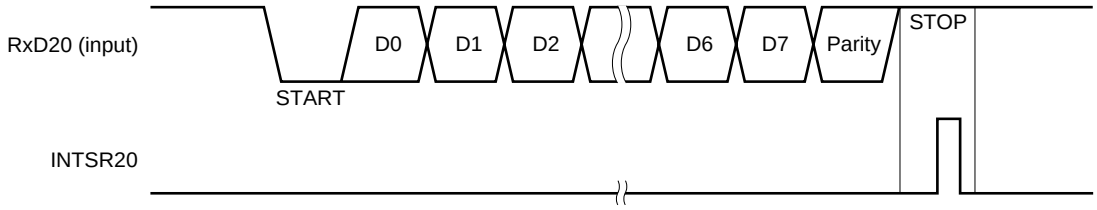
When the RxD20 pin input becomes low, the 3-bit counter starts counting, and at the time when half the time determined by the specified baud rate has passed, the data sampling start timing signal is output. If the RxD20 pin input sampled again as a result of this start timing signal is low, it is identified as a start bit, the 3-bit counter is initialized and starts counting, and data sampling is performed. When character data, a parity bit, and one stop bit are detected after the start bit, reception of one frame of data ends.

When one frame of data has been received, the receive data in the shift register is transferred to reception buffer register 20 (RXB20), and a reception completion interrupt (INTSR20) is generated.

If an error is generated, the receive data in which the error was generated is still transferred to RXB20, and INTSR20 is generated.

If the RXE20 bit is reset to 0 during the receive operation, the receive operation is stopped immediately. In this case, the contents of RXB20 and asynchronous serial interface status register 20 (ASIS20) are not changed, and INTSR20 is not generated.

Figure 11-9. Asynchronous Serial Interface Reception Completion Interrupt Timing



Caution Be sure to read reception buffer register 20 (RXB20) even if a receive error occurs. If RXB20 is not read, an overrun error will be generated when the next data is received, and the receive error state will continue indefinitely.

(e) Receive errors

The following three errors may occur during a receive operation: a parity error, framing error, and overrun error. After data reception, an error flag is set in asynchronous serial interface status register 20 (ASIS20). Receive error causes are shown in Table 11-7.

It is possible to determine what kind of error was generated during reception by reading the contents of ASIS20 in the reception error interrupt servicing (see **Figures 11-9** and **11-10**).

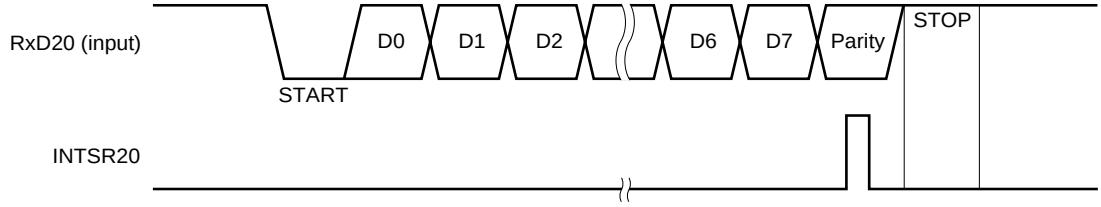
The contents of ASIS20 are reset to 0 by reading reception buffer register 20 (RXB20) or receiving the next data (if there is an error in the next data, the corresponding error flag is set).

Table 11-7. Receive Error Causes

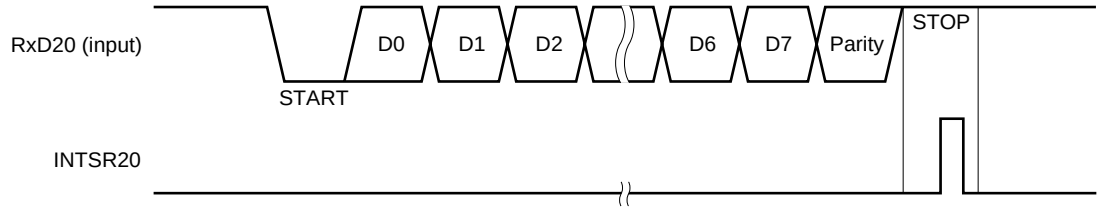
Receive Errors	Cause
Parity error	Transmission-time parity and reception data parity do not match
Framing error	Stop bit not detected
Overrun error	Reception of next data is completed before data is read from reception buffer register

Figure 14-10. Receive Error Timing

(a) Parity error generated



(b) Framing error or overrun error generated



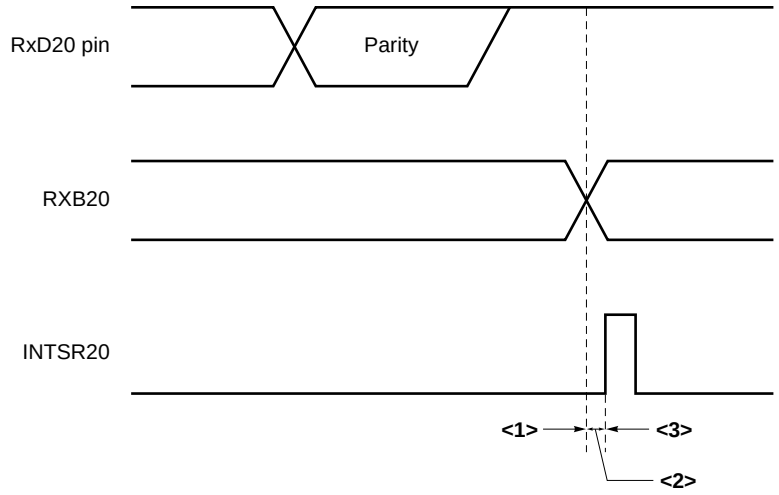
Cautions 1. The contents of the ASIS20 register are reset to 0 by reading reception buffer register 20 (RXB20) or receiving the next data. To ascertain the error contents, read ASIS20 before reading RXB20.

2. Be sure to read reception buffer register 20 (RXB20) even if a receive error is generated. If RXB20 is not read, an overrun error will be generated when the next data is received, and the receive error state will continue indefinitely.

(3) Cautions related to UART mode

(a) When bit 7 (TXE20) of asynchronous serial interface mode register 20 (ASIM20) is cleared during transmission, be sure to set transmission shift register 20 (TXS20) to FFH, then set TXE20 to 1 before executing the next transmission.

(b) When bit 6 (RXE20) of asynchronous serial interface mode register 20 (ASIM20) is cleared during reception, reception buffer register 20 (RXB20) and the receive completion interrupt (INTSR20) are as follows.



When RXE20 is set to 0 at a time indicated by **<1>**, RXB20 holds the previous data and INTSR20 is not generated.
When RXE20 is set to 0 at a time indicated by **<2>**, RXB20 renews the data and INTSR20 is not generated.
When RXE20 is set to 0 at a time indicated by **<3>**, RXB20 renews the data and INTSR20 is generated.

11.4.3 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connection of peripheral I/Os and display controllers, etc., which incorporate a conventional synchronous serial interface, such as the 75XL Series, 78K Series, 17K Series, etc.

Communication is performed using three lines: the serial clock ($\overline{SCK20}$), serial output (SO20), and serial input (SI20).

(1) Register setting

3-wire serial I/O mode settings are performed using serial operation mode register 20 (CSIM20), asynchronous serial interface mode register 20 (ASIM20), and baud rate generator control register 20 (BRGC20).

(a) Serial operation mode register 20 (CSIM20)

CSIM20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input clears CSIM20 to 00H.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM20	CSIE20	SSE20	0	0	DAP20	DIR20	CSCK20	CKP20	FF72H	00H	R/W

CSIE20	3-wire serial I/O mode operation control
0	Operation disabled
1	Operation enabled

SSE20	$\overline{SS20}$-pin selection	Function of $\overline{SS20}$/P25 pin	Communication status
0	Not used	Port function	Communication enabled
1	Used	0	Communication enabled
		1	Communication disabled

DAP20	3-wire serial I/O mode data phase selection
0	Outputs at the falling edge of $\overline{SCK20}$.
1	Outputs at the rising edge of $\overline{SCK20}$.

DIR20	First-bit specification
0	MSB
1	LSB

CSCK20	3-wire serial I/O mode clock selection
0	External clock input to the $\overline{SCK20}$ pin
1	Output of the dedicated baud rate generator

CKP20	3-wire serial I/O mode clock phase selection
0	Clock is low active, and $\overline{SCK20}$ is at high level in the idle state.
1	Clock is high active, and $\overline{SCK20}$ is at low level in the idle state.

Caution Bits 4 and 5 must all be set to 0.

(b) Asynchronous serial interface mode register 20 (ASIM20)

ASIM20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIM20 to 00H.

When 3-wire serial I/O mode is selected, ASIM20 must be set to 00H.

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM20	TXE20	RXE20	PS201	PS200	CL20	SL20	0	0	FF70H	00H	R/W

TXE20	Transmit operation control
0	Transmit operation stop
1	Transmit operation enable

RXE20	Receive operation control
0	Receive operation stop
1	Receive operation enable

PS201	PS200	Parity bit specification
0	0	No parity
0	1	Always add 0 parity at transmission. Parity check is not performed at reception (No parity error is generated).
1	0	Odd parity
1	1	Even parity

CL20	Character length specification
0	7 bits
1	8 bits

SL20	Transmit data stop bit length specification
0	1 bit
1	2 bits

Cautions 1. Bits 0 and 1 must all be set to 0.

2. Switch operating modes after halting serial transmit/receive operation.

(c) Baud rate generator control register 20 (BRGC20)

BRGC20 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears BRGC20 to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BRGC20	TPS203	TPS202	TPS201	TPS200	0	0	0	0	FF73H	00H	R/W

TPS203	TPS202	TPS201	TPS200	3-bit counter source clock selection	n
0	0	0	0	$f_X/2$ (2.5 MHz)	1
0	0	0	1	$f_X/2^2$ (1.25 MHz)	2
0	0	1	0	$f_X/2^3$ (625 kHz)	3
0	0	1	1	$f_X/2^4$ (313 kHz)	4
0	1	0	0	$f_X/2^5$ (156 kHz)	5
0	1	0	1	$f_X/2^6$ (78.1 kHz)	6
0	1	1	0	$f_X/2^7$ (39.1 kHz)	7
0	1	1	1	$f_X/2^8$ (19.5 kHz)	8
Other than above				Setting prohibited	

Cautions 1. When writing to BRGC20 is performed during a communication operation, the baud rate generator output is disrupted and communications cannot be performed normally. Be sure not to write to BRGC20 during communication operation.

2. Be sure not to select n = 1 during an operation at f_X = 5.0 MHz because the resulting baud rate exceeds the rated range.

3. When the external input clock is selected, set port mode register 2 (PM2) to input mode.

Remarks 1. f_X: Main system clock oscillation frequency

2. n: Values determined by the settings of TPS200 to TPS203 ($1 \leq n \leq 8$)

3. The parenthesized values apply to operation at f_X = 5.0 MHz.

If the internal clock is used as the serial clock for 3-wire serial I/O mode, set bits TPS200 to TPS203 to set the frequency of the serial clock. To obtain the frequency to be set, use the following expression. When an external clock is used, setting BRGC20 is not necessary.

$$\text{Serial clock frequency} = \frac{f_X}{2^{n+1}} \text{ [Hz]}$$

f_X: Main system clock oscillation frequency

n: Values in the above table determined by the settings of TPS200 to TPS203 ($1 \leq n \leq 8$)

(2) Communication operation

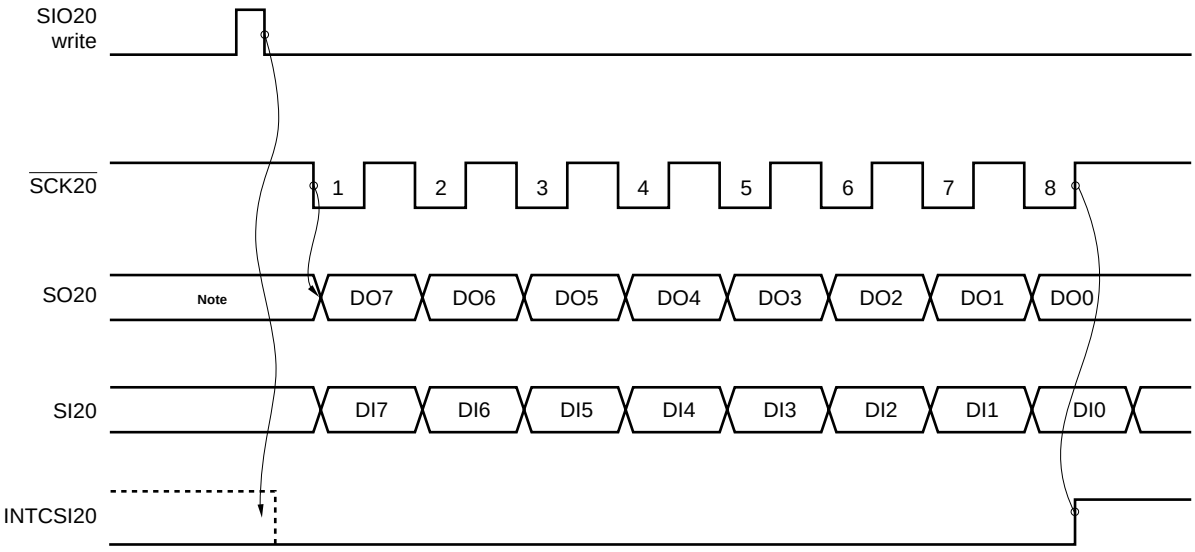
In 3-wire serial I/O mode, data transmission/reception is performed in 8-bit units. Data is transmitted/received bit by bit in synchronization with the serial clock.

Transmission shift register (TXS20/SIO20) and reception shift register (RXS20) shift operations are performed in synchronization with the fall of the serial clock ($\overline{SCK20}$). Then transmit data is held in the SO20 latch and output from the SO20 pin. Also, receive data input to the SI20 pin is latched in the reception buffer register (RXB20/SIO20) on the rise of $\overline{SCK20}$.

At the end of an 8-bit transfer, the operation of TXS20/SIO20 and RXS20 stops automatically, and the interrupt request signal (INTCSI20) is generated.

Figure 11-11. 3-Wire Serial I/O Mode Timing (1/7)

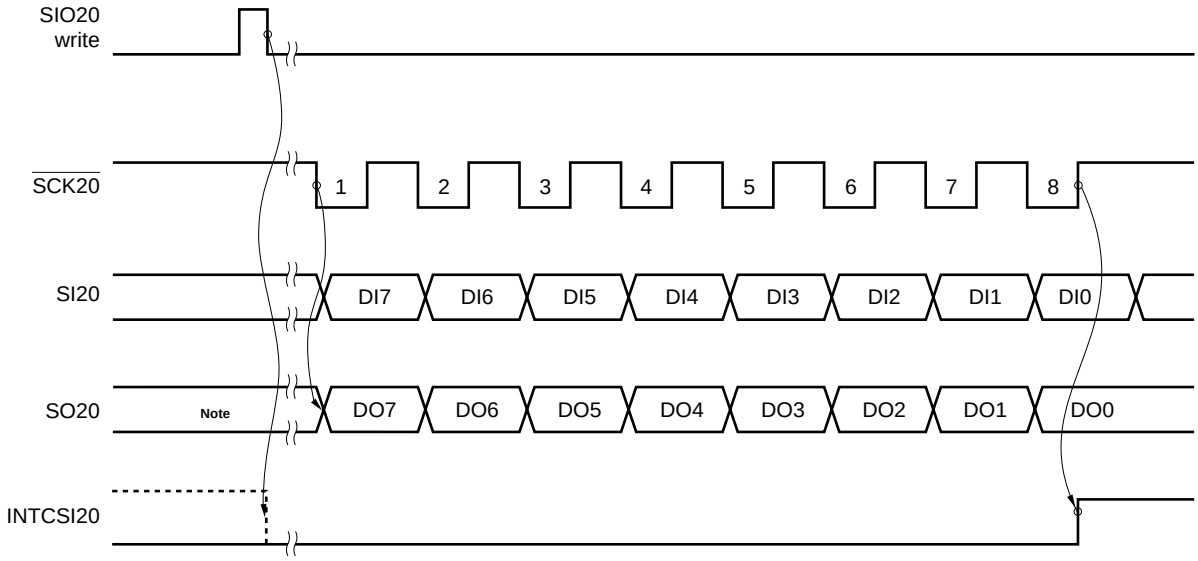
(i) Master operation timing (when DAP20 = 0, CKP20 = 0, SSE20 = 0)



Note The value of the last bit previously output is output.

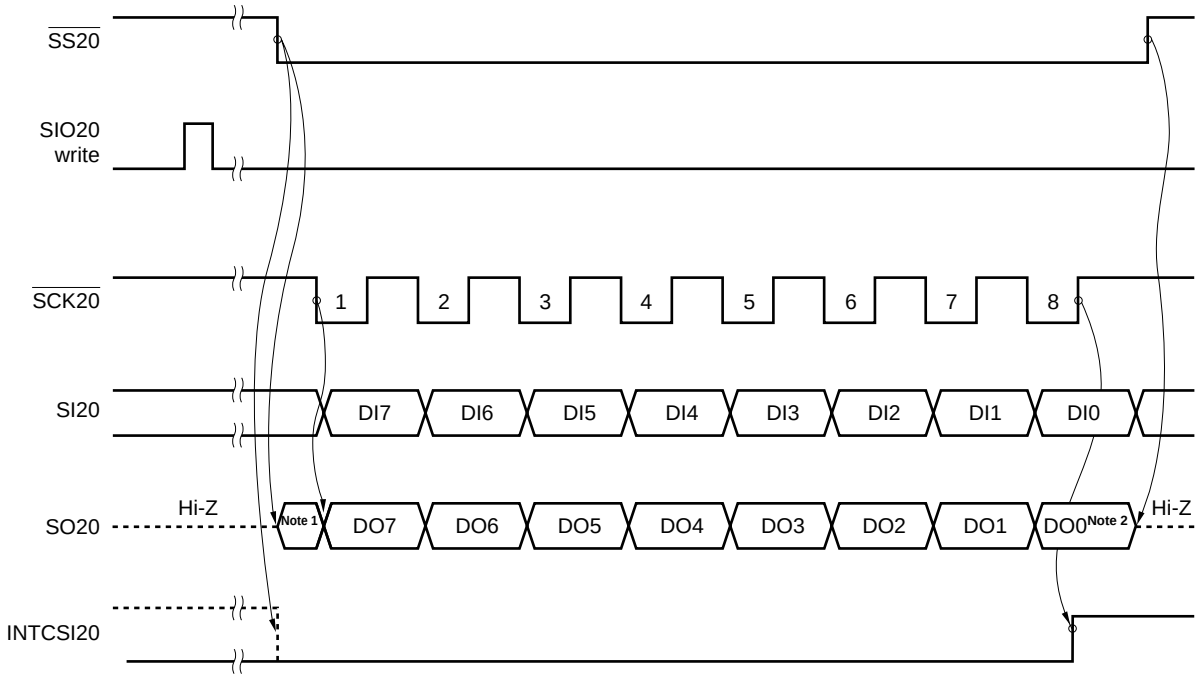
Figure 11-11. 3-Wire Serial I/O Mode Timing (2/7)

(ii) Slave operation timing (when DAP20 = 0, CKP20 = 0, SSE20 = 0)



Note The value of the last bit previously output is output.

(iii) Slave operation (when DAP20 = 0, CKP20 = 0, SSE20 = 1)

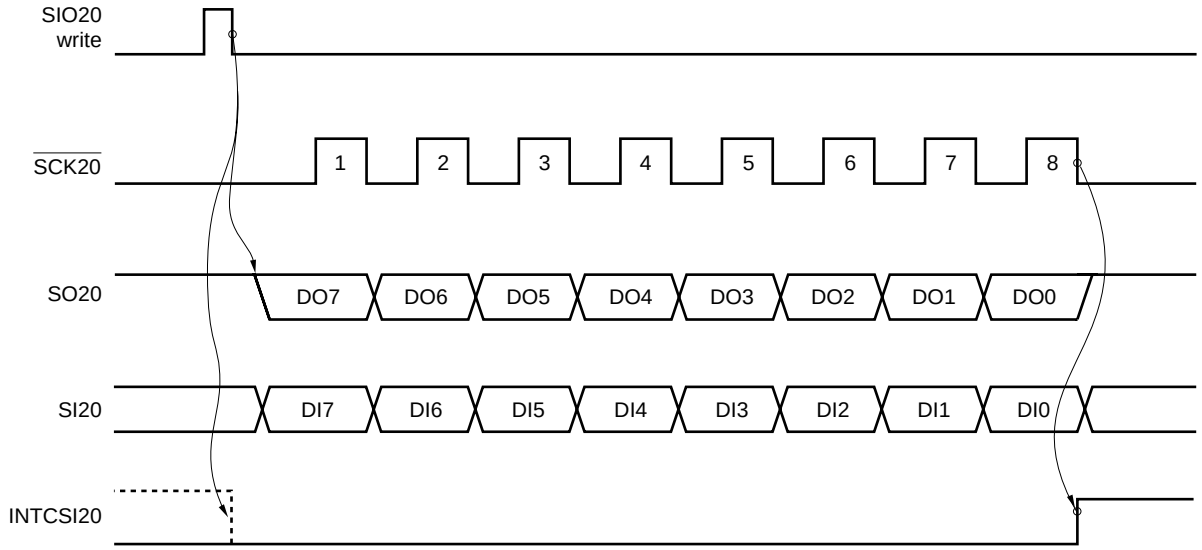


Notes 1. The value of the last bit previously output is output.

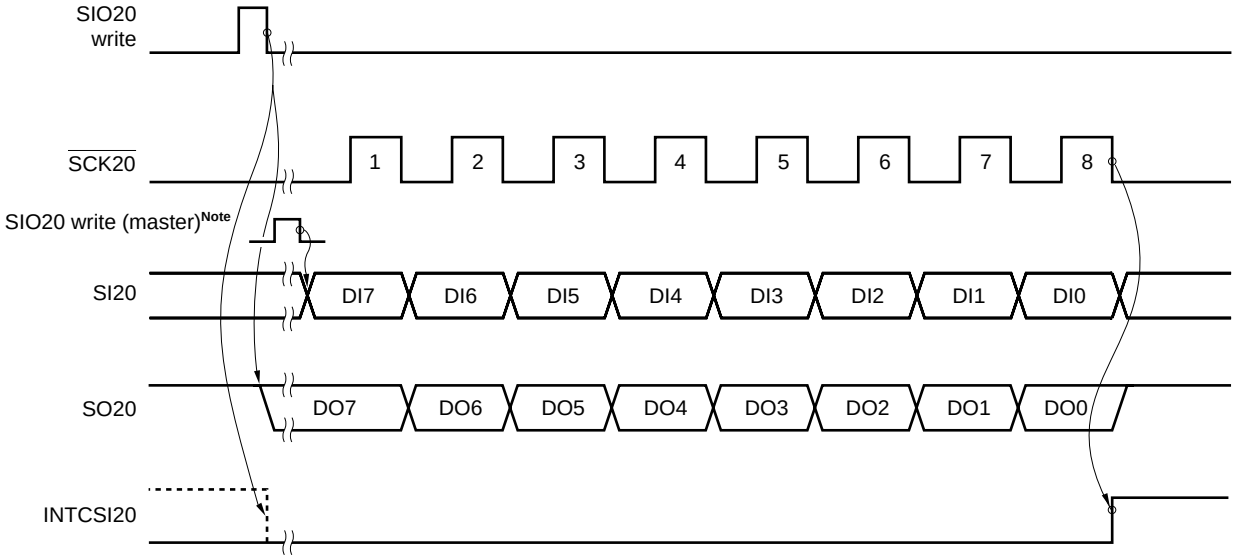
2. DO0 is output until $\overline{SS20}$ rises.
When $\overline{SS20}$ is high, SO20 is in a high-impedance state.

Figure 11-11. 3-Wire Serial I/O Mode Timing (3/7)

(iv) Master operation (when DAP20 = 0, CKP20 = 1, SSE20 = 0)



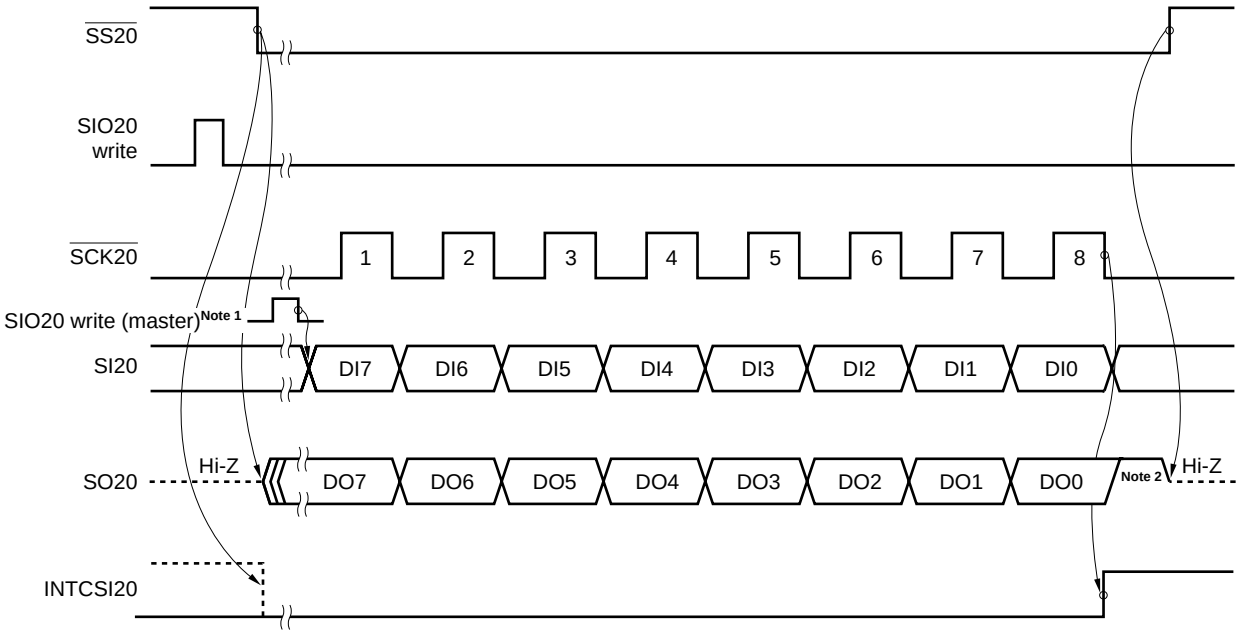
(v) Slave operation (when DAP20 = 0, CKP20 = 1, SSE20 = 0)



Note The data of SI20 is loaded at the first rising edge of $\overline{\text{SCK20}}$. Make sure that the master outputs the first bit before the first rising of $\overline{\text{SCK20}}$.

Figure 11-11. 3-Wire Serial I/O Mode Timing (4/7)

(vi) Slave operation (when DAP20 = 0, CKP20 = 1, SSE20 = 1)



Notes 1. The data of SI20 is loaded at the first rising edge of $\overline{\text{SCK20}}$. Make sure that the master outputs the first bit before the first rising of $\overline{\text{SCK20}}$.

2. SO20 is high until $\overline{\text{SS20}}$ rises after completion of DO0 output. When $\overline{\text{SS20}}$ is high, SO20 is in a high-impedance state.

(vii) Master operation (when DAP20 = 1, CKP20 = 0, SSE20 = 0)

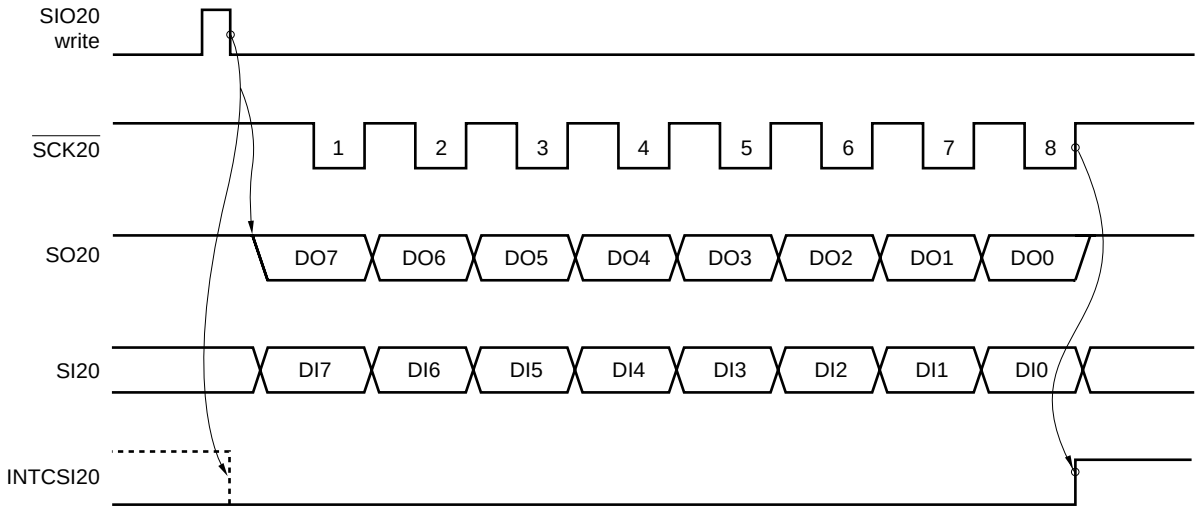
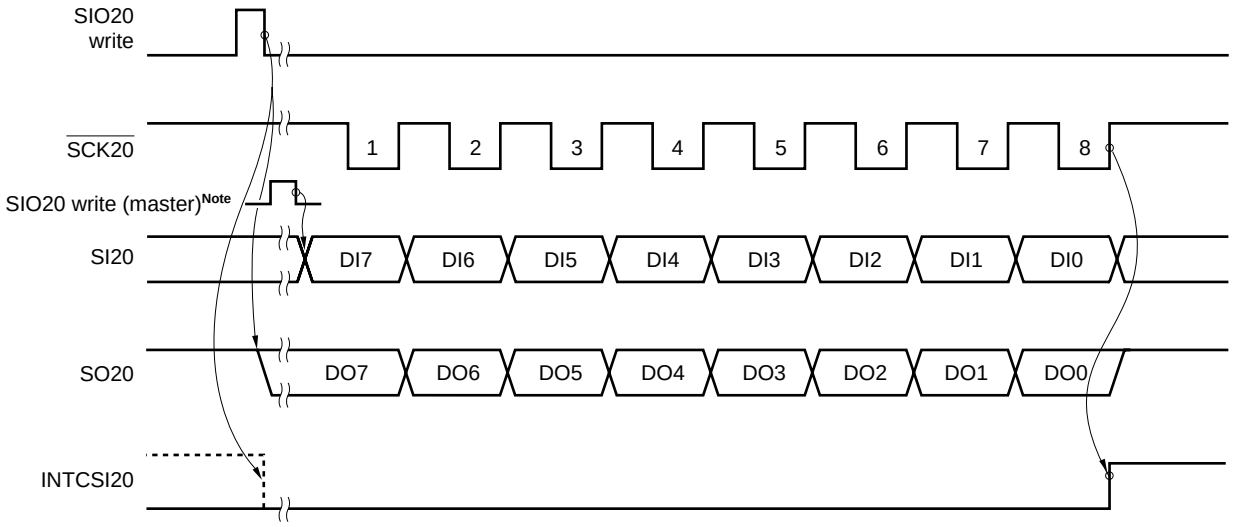


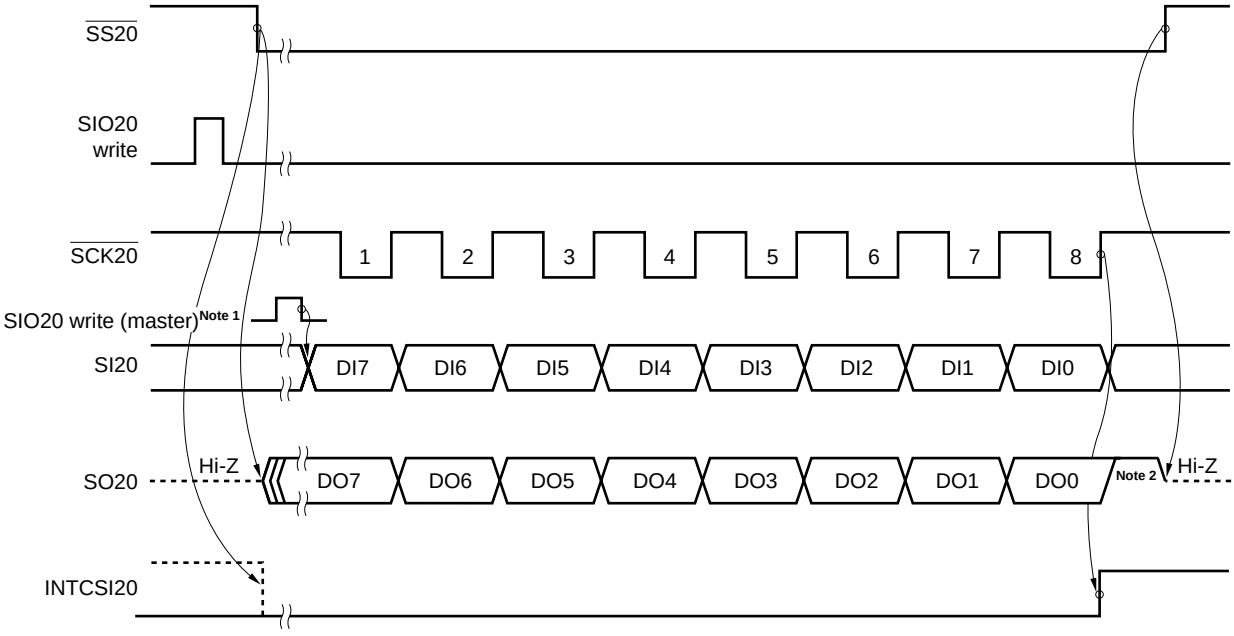
Figure 11-11. 3-Wire Serial I/O Mode Timing (5/7)

(viii) Slave operation (when DAP20 = 1, CKP20 = 0, SSE20 = 0)



Note The data of SI20 is loaded at the first falling edge of $\overline{\text{SCK20}}$. Make sure that the master outputs the first bit before the first falling of $\overline{\text{SCK20}}$.

(ix) Slave operation (when DAP20 = 1, CKP20 = 0, SSE20 = 1)

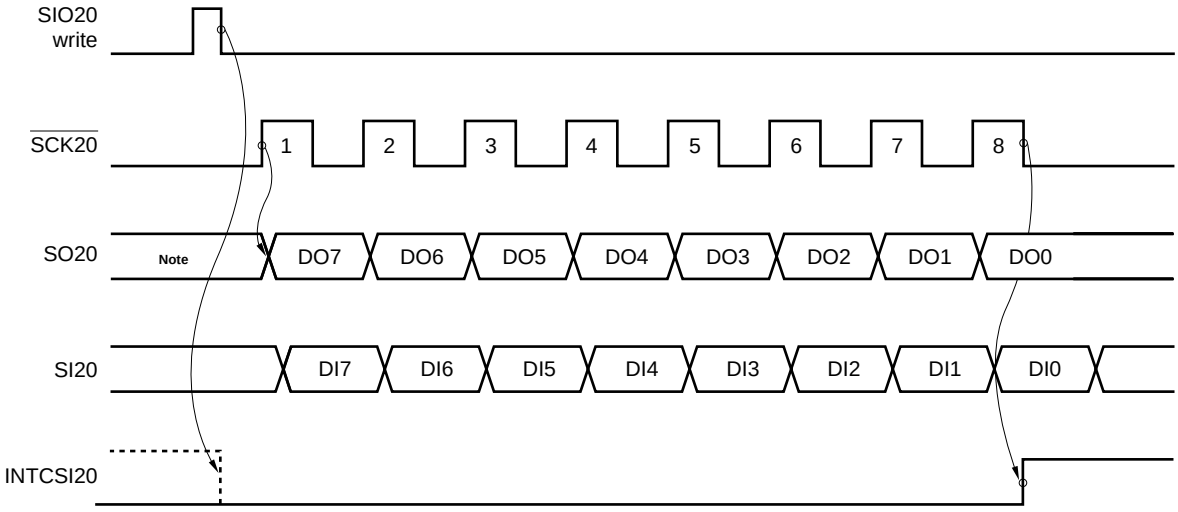


Notes 1. The data of SI20 is loaded at the first falling edge of $\overline{\text{SCK20}}$. Make sure that the master outputs the first bit before the first falling of $\overline{\text{SCK20}}$.

2. SO20 is high until $\overline{\text{SS20}}$ rises after completion of DO0 output. When $\overline{\text{SS20}}$ is high, SO20 is in a high-impedance state.

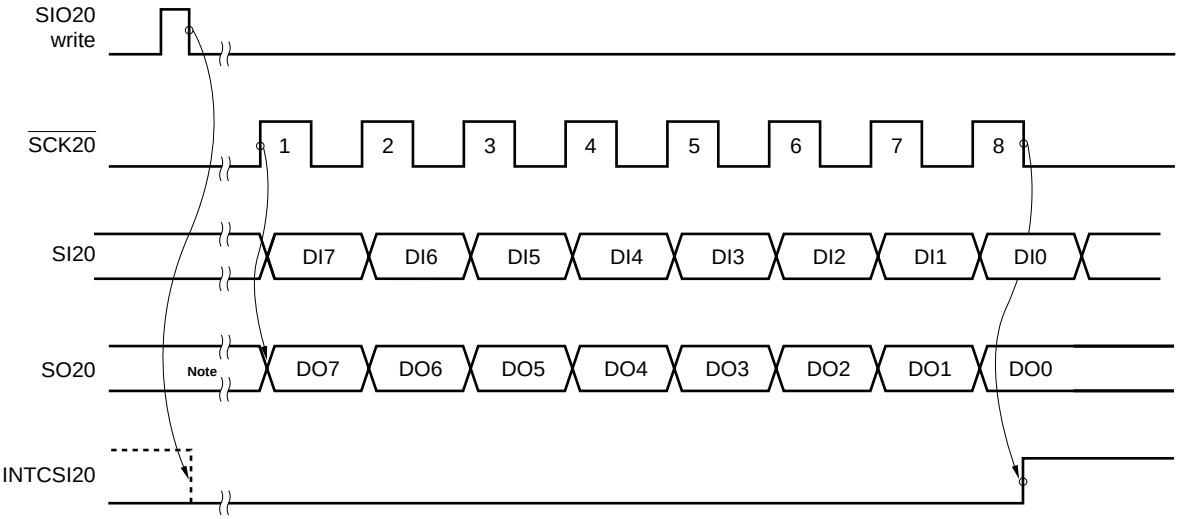
Figure 11-11. 3-Wire Serial I/O Mode Timing (6/7)

(x) Master operation (when DAP20 = 1, CKP20 = 1, SSE20 = 0)



Note The value of the last bit previously output is output.

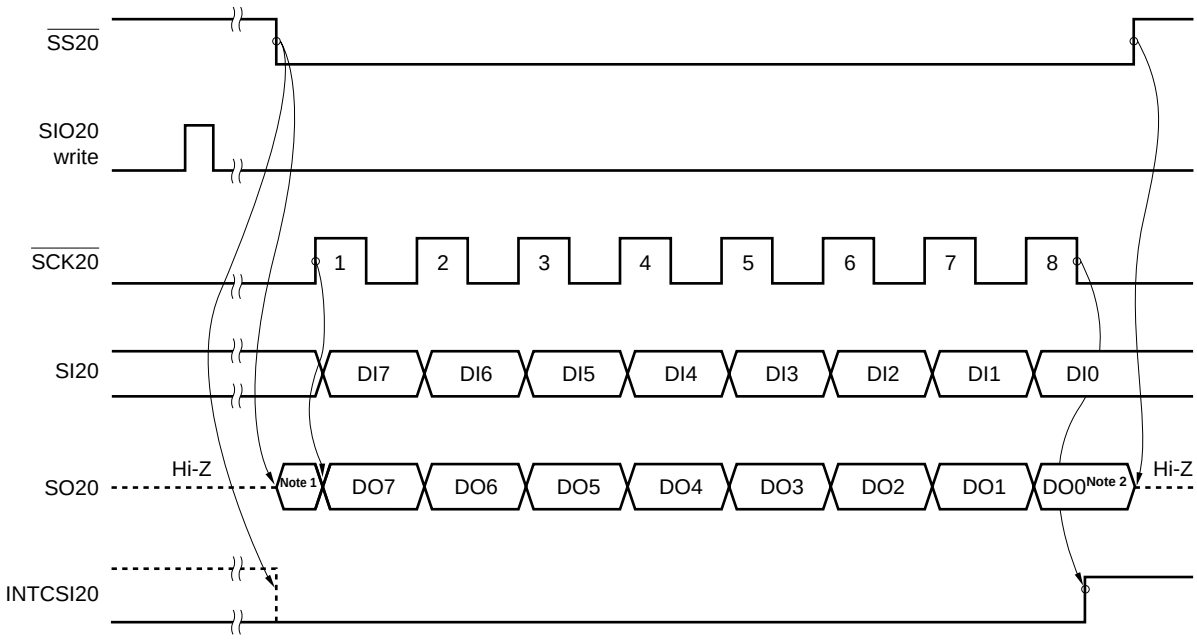
(xi) Slave operation (when DAP20 = 1, CKP20 = 1, SSE20 = 0)



Note The value of the last bit previously output is output.

Figure 11-11. 3-Wire Serial I/O Mode Timing (7/7)

(xii) Slave operation (when DAP20 = 1, CKP20 = 1, SSE20 = 1)



Notes 1. The value of the last bit previously output is output.

2. DO0 is output until $\overline{SS20}$ rises.
When $\overline{SS20}$ is high, SO20 is in a high-impedance state.

(3) Transfer start

Serial transfer is started by setting transfer data to the transmission shift register (TXS20/SIO20) when the following two conditions are satisfied.

- Serial operation mode register 20 (CSIM20) bit 7 (CSIE20) = 1
- Internal serial clock is stopped or $\overline{SCK20}$ is high after 8-bit serial transfer.

Caution If CSIE20 is set to "1" after data is written to TXS20/SIO20, transfer does not start.

A termination of 8-bit transfer stops the serial transfer automatically and generates the interrupt request signal (INTCSI20).

CHAPTER 12 MULTIPLIER

12.1 Multiplier Function

The multiplier has the following function:

- Calculation of 8 bits × 8 bits = 16 bits

12.2 Multiplier Configuration

(1) 16-bit multiplication result storage register 0 (MUL0)

This register stores the 16-bit result of multiplication.

This register holds the result of multiplication after the 16 CPU clocks have elapsed.

MUL0 is set with a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes this register undefined.

Caution MUL0 is designed to be manipulated with a 16-bit memory manipulation instruction. It can also be manipulated with 8-bit memory manipulation instructions, however. When an 8-bit memory manipulation instruction is used to manipulate MUL0, it must be accessed in direct addressing.

(2) Multiplication data registers A and B (MRA0 and MRB0)

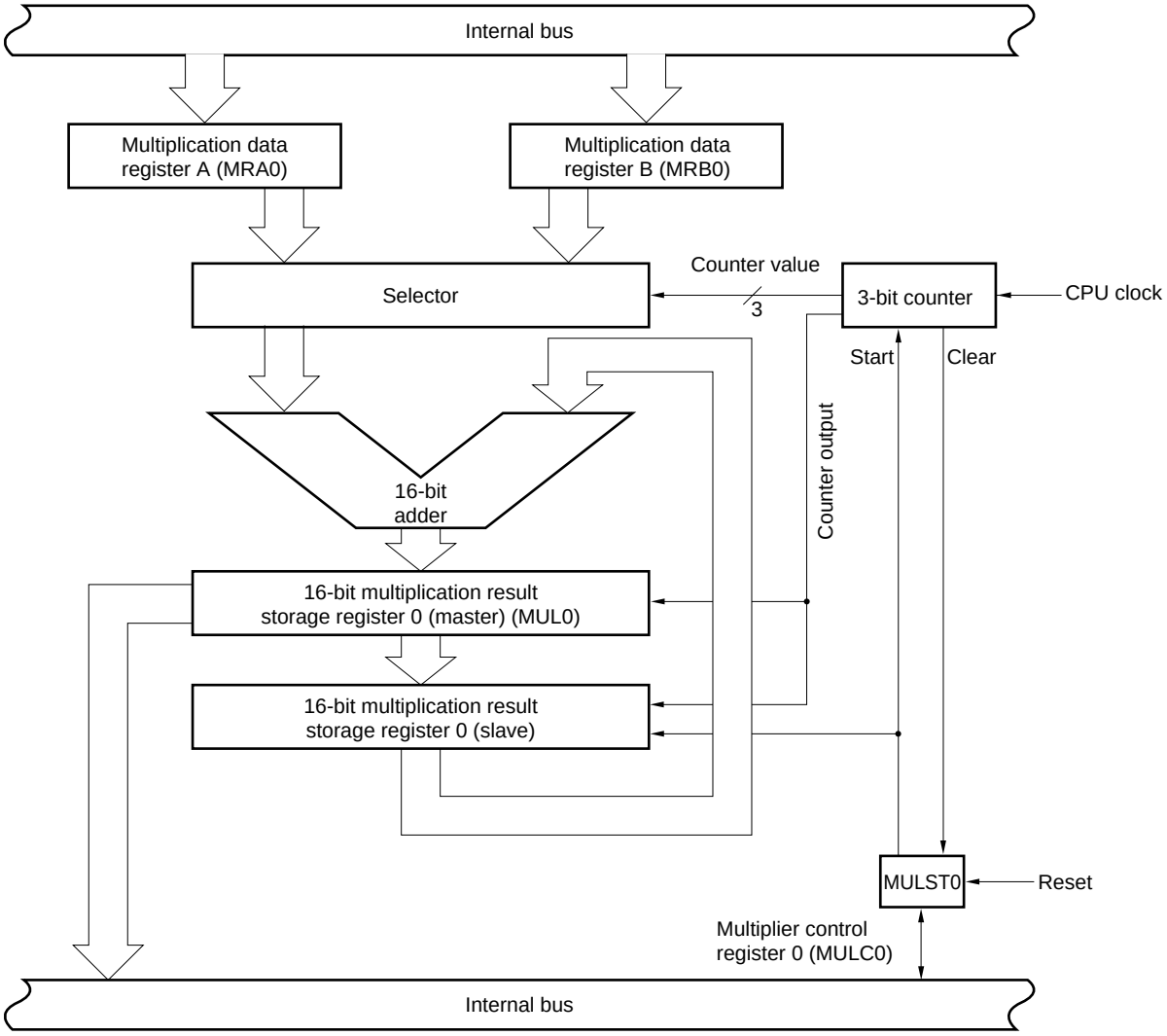
These are 8-bit multiplication data storage registers. The multiplier multiplies the values of MRA0 and MRB0.

MRA0 and MRB0 are set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes these registers undefined.

Figure 16-1 shows a block diagram of the multiplier.

Figure 12-1. Block Diagram of Multiplier



12.3 Multiplier Control Register

The multiplier is controlled by the following register:

- Multiplier control register 0 (MULC0)

MULC0 indicates the operating status of the multiplier, as well as controls the multiplier.
MULC0 is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 12-2. Format of Multiplier Control Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
MULC0	0	0	0	0	0	0	0	MULST0	FFD2H	00H	R/W

MULST0	Multiplier operation start control bit	Operating status of multiplier
0	Stops operation after resetting counter to 0.	Operation stops
1	Enables operation	Operation in progress

Caution Bits 1 to 7 must all be set to 0.

12.4 Multiplier Operation

The multiplier of the μPD789188 and 78F9189 Subseries can execute calculation of 8 bits $\times$ 8 bits = 16 bits. Figure 16-3 shows the operation timing of the multiplier where MRA0 is set to AAH and MRB0 is set to D3H.

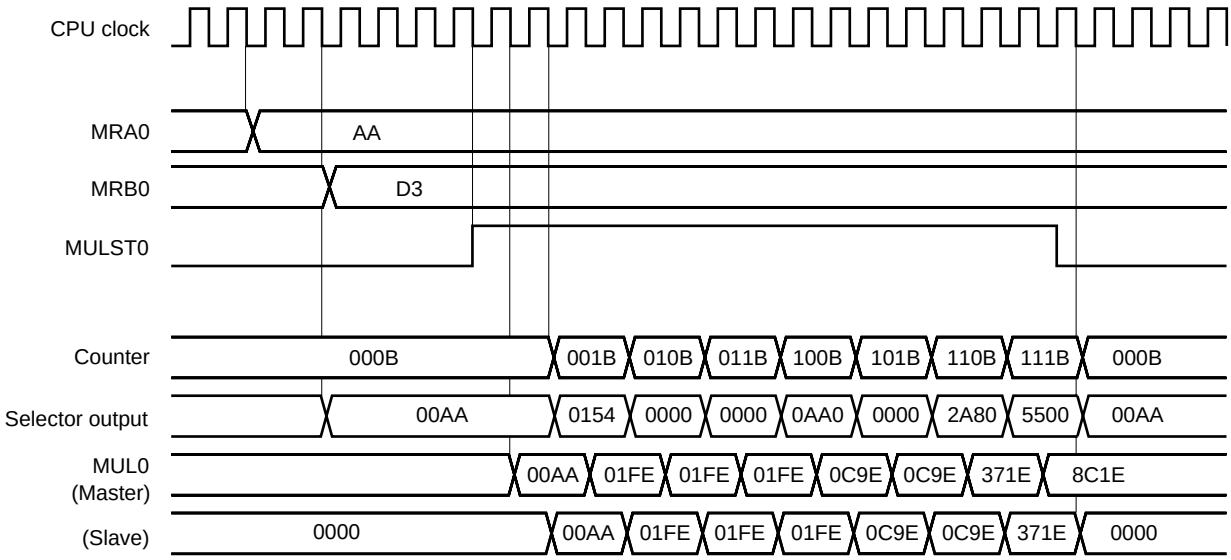
<1> Counting is started by setting MULST0.

<2> The data generated by the selector is added to the data of MUL0 at each CPU clock, and the counter value is incremented by one.

<3> If MULST0 is cleared when the counter value is 111B, the operation is stopped. At this time, MUL0 holds the data.

<4> While MULST0 is low, the counter and slave are cleared.

Figure 12-3. Multiplier Operation Timing (Example of AAH $\times$ D3H)



CHAPTER 13 INTERRUPT FUNCTIONS

13.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally. It does not undergo interrupt priority control and is given top priority over all other interrupt requests.

A standby release signal is generated.

The non-maskable interrupt has one source of interrupt from the watchdog timer.

(2) Maskable interrupt

These interrupts undergo mask control. If two or more interrupts are simultaneously generated, each interrupt has a predetermined priority (priority) as shown in Table 13-1.

A standby release signal is generated.

For the μPD789188 and 78F9189 Subseries, the maskable interrupt has four sources of external interrupts and ten sources of internal interrupts.

13.2 Interrupt Sources and Configuration

There are a total of 15 non-maskable and maskable interrupts in the interrupt sources for the μPD789188 and 78F9189 Subseries,.

Table 13-1. Interrupt Sources

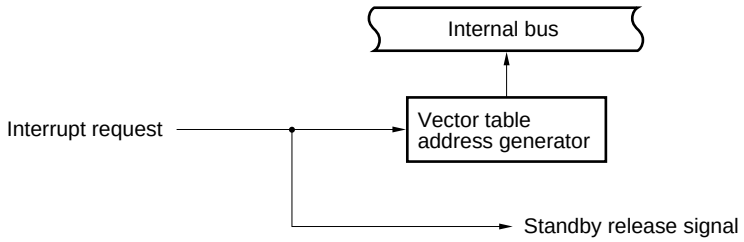
Interrupt Type	Priority[Note 1]	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Non-maskable interrupt	–	INTWDT	Watchdog timer overflow (when watchdog timer mode 1 is selected)	Internal	0004H	(A)
Maskable interrupt	0	INTWDT	Watchdog timer overflow (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	INTSR20	End of UART reception on serial interface 20	Internal	000EH	(B)
		INTCSI20	End of three-wire SIO transfer reception on serial interface 20			
	6	INTST20	End of UART transmission on serial interface 20		0010H	
	7	INTWT	Watch timer interrupt		0012H	
	8	INTWTI	Interval timer interrupt		0014H	
	9	INTTM80	Generation of match signal for 8-bit timer/event counter 80		0016H	
	10	INTTM81	Generation of match signal for 8-bit timer/event counter 81		0018H	
	11	INTTM82	Generation of match signal for 8-bit timer 82		001AH	
	12	INTTM90	Generation of match signal for 16-bit timer 90		001CH	
		-			001EH	
		-			0020H	
	13	INTAD0	A/D conversion completion signal		0022H	

Notes 1. The priority regulates which maskable interrupt is higher, when two or more maskable interrupts are requested simultaneously. Zero signifies the highest priority, while 13 is the lowest.

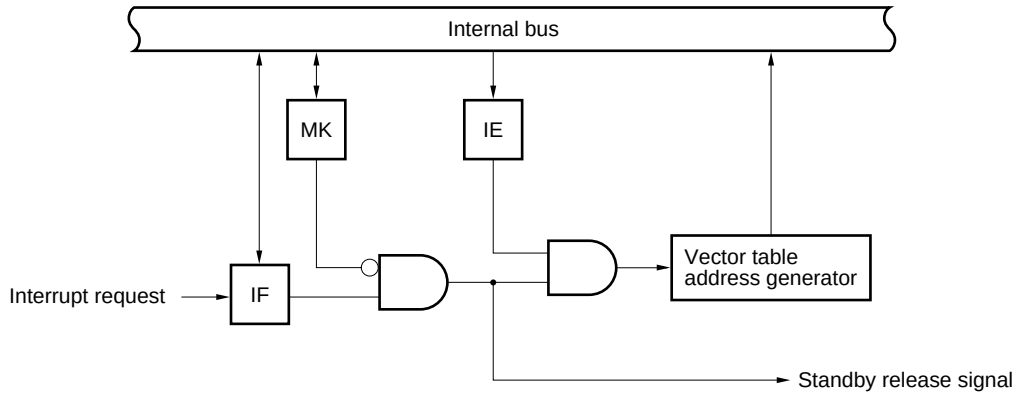
2. Basic configuration types (A), (B), and (C) correspond to (A), (B), and (C) in Figure 17-1, respectively.

Figure 13-1. Basic Configuration of Interrupt Function

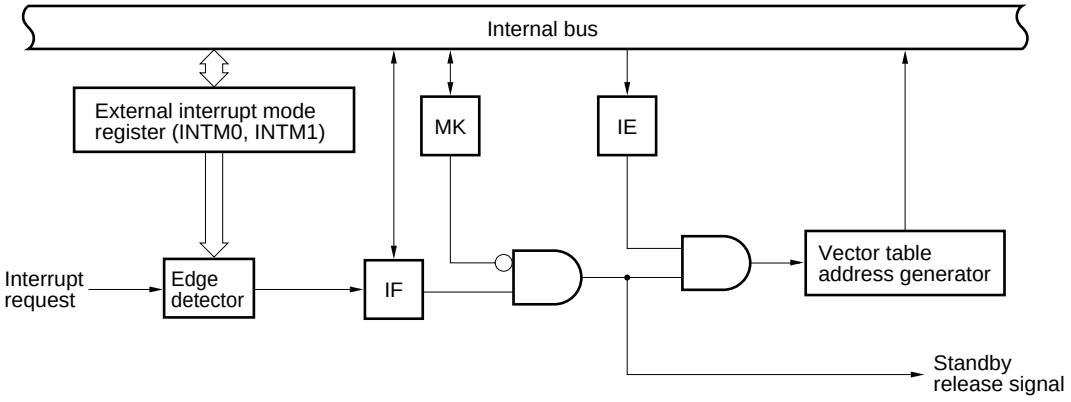
(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt



IF: Interrupt request flag

IE: Interrupt enable flag

MK: Interrupt mask flag

13.3 Interrupt Function Control Registers

The interrupt functions are controlled by the following registers:

- Interrupt request flag registers 0 and 1 (IF0 and IF1)
- Interrupt mask flag registers 0 and 1 (MK0 and MK1)
- External interrupt mode registers 0 and 1 (INTM0 and INTM1)
- Program status word (PSW)

Table 13-2 lists interrupt requests, the corresponding interrupt request flags, and interrupt mask flags.

Table 13-2. Interrupt Request Signals and Corresponding Flags

Interrupt Request Signal	Interrupt Request Flag	Interrupt Mask Flag
INTWDT	TMIF4	TMMK4
INTP0	PIF0	PMK0
INTP1	PIF1	PMK1
INTP2	PIF2	PMK2
INTP3	PIF3	PMK3
INTSR20/INTCSI20	SRIF20	SRMK20
INTST20	STIF20	STMK20
INTWT	WTIF	WTMK
INTWTI	WTIIF	WTIMK
INTTM80	TMIF80	TMMK80
INTTM81	TMIF81	TMMK81
INTTM82	TMIF82	TMMK82
INTTM90	TMIF90	TMMK90
INTAD0	ADIF0	ADMK0

(1) Interrupt request flag registers (IF0 and IF1)

An interrupt request flag is set to 1, when the corresponding interrupt request is issued, or when the related instruction is executed. It is cleared to 0, when the interrupt request is accepted, when a $\overline{\text{RESET}}$ signal is input, or when a related instruction is executed.

IF0 and IF1 are set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears IF0 and IF1 to 00H.

Figure 13-2. Format of Interrupt Request Flag Register

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
IF0	WTIF	STIF20	SRIF20	PIF3	PIF2	PIF1	PIF0	TMIF4	FFE0H	00H	R/W

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
IF1	ADIF0	0	0	TMIF90	TMIF82	TMF81	TMIF80	WTIIF	FFE1H	00H	R/W

xxIFx	Interrupt request flag
0	No interrupt request signal has been issued.
1	An interrupt request signal has been issued ;an interrupt request has been made.

Cautions 1. The TMIF4 flag can be read- and write-accessed only when the watchdog timer is being used as an interval timer. It must be cleared to 0 if the watchdog timer is used in watchdog timer mode 1 or 2.

2. When port 3 is being used as an output port, and its output level is changed, an interrupt request flag is set, because this port is also used as an external interrupt input. To use port 3 in output mode, therefore, the interrupt mask flag must be set to 1 in advance.

(2) Interrupt mask flag registers (MK0 and MK1)

The interrupt mask flags are used to enable and disable the corresponding maskable interrupts.

MK0 and MK1 are set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets MK0 and MK1 to FFH.

Figure 13-3. Format of Interrupt Mask Flag Register

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
MK0	WTMK	STMK2	SRMK	PMK3	PMK2	PMK1	PMK0	TMMK	FFE4H	FFH	R/W
		0	20					4			

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
MK1	ADMK	1	1	TMMK	TMMK	TMMK	TMMK	WTIMK	FFE5H	FFH	R/W
	0			90	82	81	80				

xxMK	Interrupt handling control
0	Enable interrupt handling.
1	disable interrupt handling.

Cautions 1. When the watchdog timer is being used in watchdog timer mode 1 or 2, any attempt to read TMMK4 flag results in an undefined value being detected.

2. When port 3 is being used as an output port, and its output level is changed, an interrupt request flag is set, because this port is also used as an external interrupt input. To use port 3 in output mode, therefore, the interrupt mask flag must be set to 1 in advance.

(3) External interrupt mode register 0 (INTM0)

INTM0 is used to specify the valid edge for INTP0 to INTP2.

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

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

Figure 13-4. Format of External Interrupt Mode Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTM0	ES21	ES20	ES11	ES10	ES01	ES00	0	0	FFECH	00H	R/W

ES21	ES20	INTP2 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES11	ES10	INTP1 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES01	ES00	INTP0 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Cautions 1. Bits 0 and 1 must all be set to 0.

2. Before setting INTM0, set the corresponding interrupt mask flag register (××MK×) to 1 to disable interrupts.

To enable interrupts, clear to 0 the corresponding interrupt request flag (××IF×), then the corresponding interrupt mask flag register (××MK×).

(4) External interrupt mode register 1 (INTM1)

INTM1 is used to specify the valid edge for INTP3 and INTLVI0.

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

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

Figure 13-5. Format of External Interrupt Mode Register 1

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTM1	0	0	0	0	ES41	ES40	ES31	ES30	FFEDH	00H	R/W

ES41	ES40	INTLVI0 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES31	ES30	INTP3 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Cautions 1. Bits 4 to 7 must all be set to 0.

2. Before setting INTM1, set the corresponding interrupt mask flag register (××MK×) to 1 to disable interrupts.

To enable interrupts, clear to 0 the corresponding interrupt request flag (××IF×), then the corresponding interrupt mask flag register (××MK×).

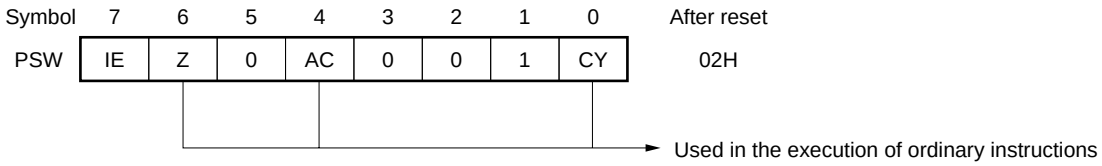
(5) Program status word (PSW)

The program status word is used to hold the instruction execution result and the current status of the interrupt requests. The IE flag, used to enable and disable maskable interrupts, is mapped to PSW.

PSW can be read- and write-accessed in 8-bit units, as well as using bit manipulation instructions and dedicated instructions (EI and DI). When a vector interrupt is accepted, PSW is automatically saved to a stack, and the IE flag is reset to 0.

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

Figure 13-6. Program Status Word Configuration



IE	Whether to enable/disable interrupt acceptance
0	Disable
1	Enable

13.4 Interrupt Processing Operation

13.4.1 Non-maskable interrupt request acceptance operation

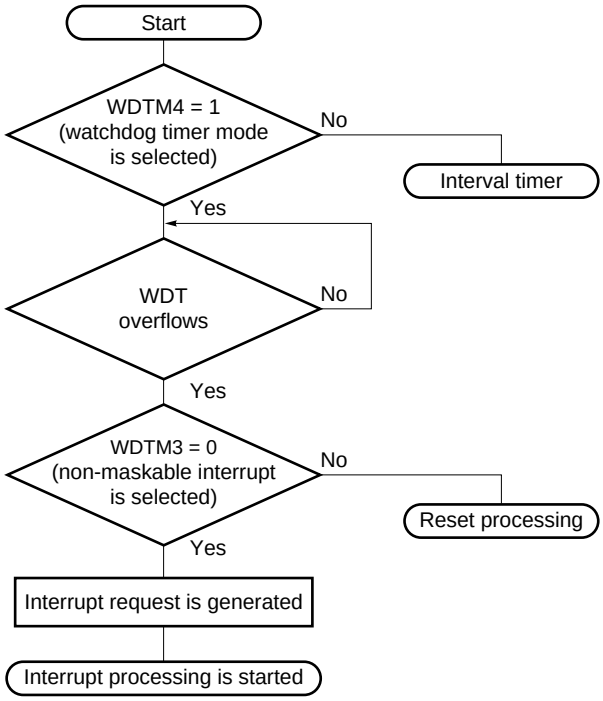
The non-maskable interrupt request is unconditionally accepted even when interrupts are disabled. It is not subject to interrupt priority control and takes precedence over all other interrupts.

When the non-maskable interrupt request is acknowledged, PSW and PC are saved to the stack in that order, the IE flag is reset to 0, the contents of the vector table are loaded to the PC, and then program execution branches.

Figure 13-7 shows the flowchart from non-maskable interrupt request generation to acceptance. Figure 13-8 shows the timing of non-maskable interrupt request acceptance. Figure 13-9 shows the acceptance operation if multiple non-maskable interrupts are generated.

Caution During a non-maskable interrupt service program execution, do not input another non-maskable interrupt request; if it is input, the service program will be interrupted and the new interrupt request will be acknowledged.

Figure 13-7. Flowchart from Non-Maskable Interrupt Request Generation to Acceptance



WDTM: Watchdog timer mode register
WDT: Watchdog timer

Figure 13-8. Timing of Non-Maskable Interrupt Request Acceptance

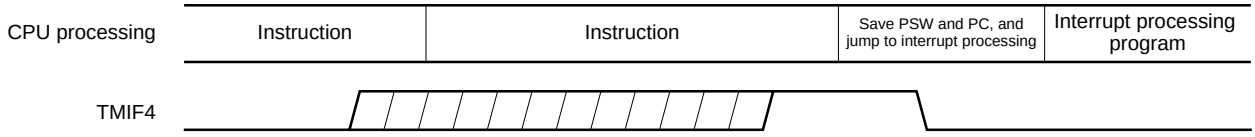
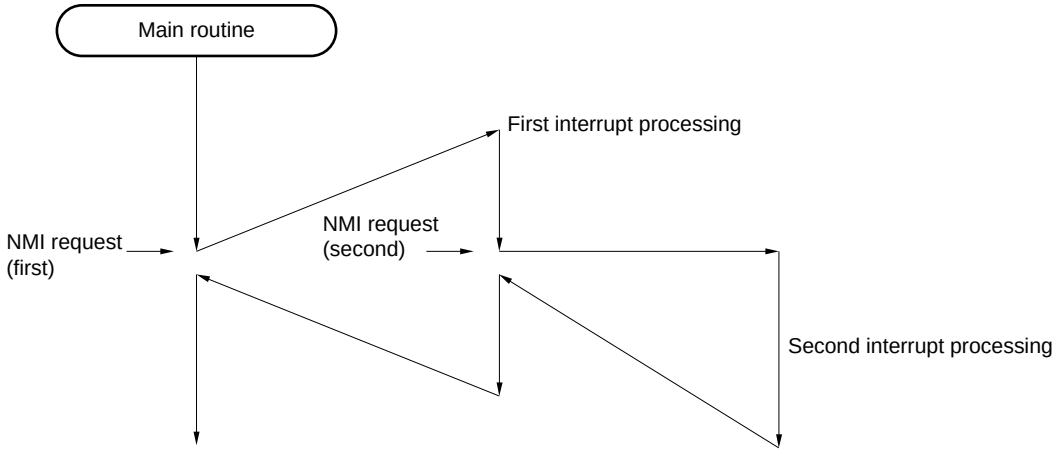


Figure 13-9. Accepting Non-Maskable Interrupt Request



13.4.2 Maskable interrupt request acceptance operation

A maskable interrupt request can be accepted when the interrupt request flag is set to 1 and the corresponding interrupt mask flag is cleared to 0. A vectored interrupt request is accepted in the interrupt enabled status (when the IE flag is set to 1).

The time required to start the interrupt processing after a maskable interrupt request has been generated is shown in Table 13-3.

See Figures 13-11 and 13-12 for the interrupt request acceptance timing.

Table 13-3. Time from Generation of Maskable Interrupt Request to Processing

Minimum Time	Maximum Time[Note]
9 clocks	19 clocks

Note The wait time is maximum when an interrupt request is generated immediately before BT and BF instruction.

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

When two or more maskable interrupt requests are generated at the same time, they are accepted starting from the interrupt request assigned the highest priority.

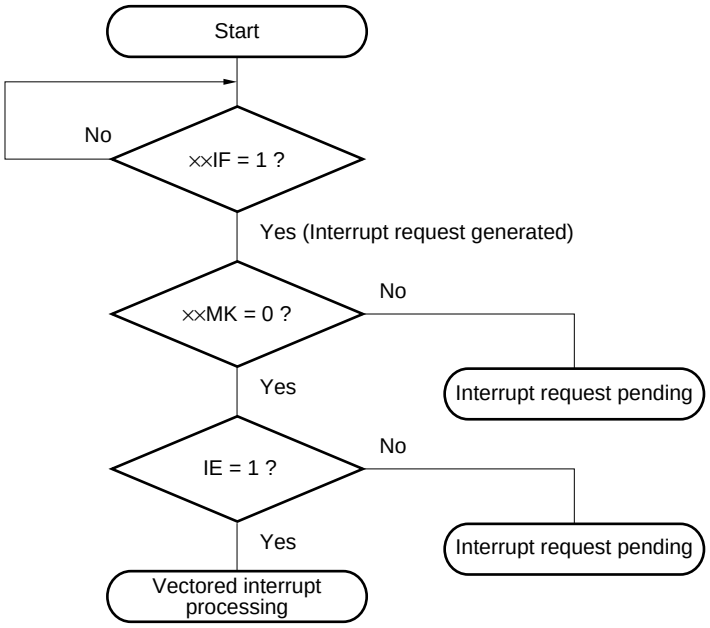
A pending interrupt is accepted when the status where it can be accepted is set.

Figure 17-10 shows the algorithm of accepting interrupt requests.

When a maskable interrupt request is accepted, the contents of PSW and PC are saved to the stack in that order, the IE flag is reset to 0, and the data in the vector table determined for each interrupt request is loaded to the PC, and execution branches.

To return from interrupt processing, use the RETI instruction.

Figure 13-10. Interrupt Request Acceptance Processing Algorithm

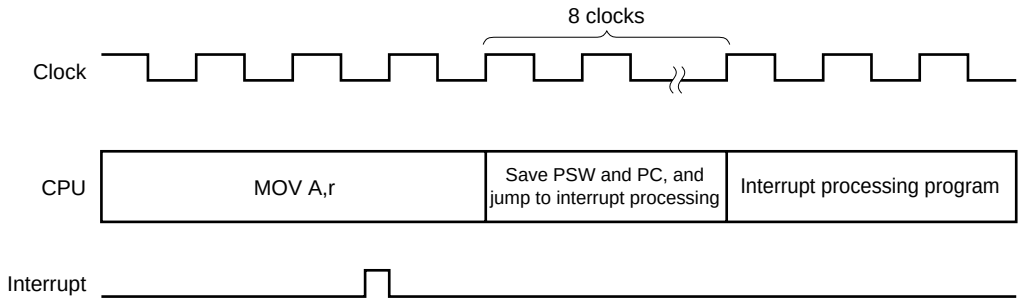


××IF: Interrupt request flag

××MK: Interrupt mask flag

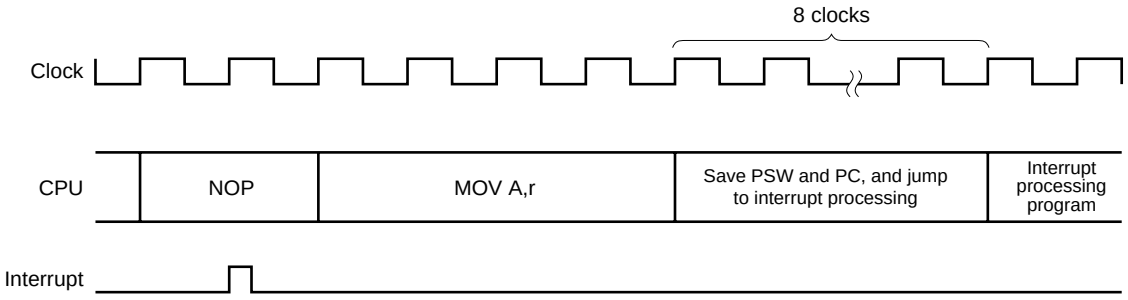
IE: Flag to control maskable interrupt request acceptance (1 = enable, 0 = disable)

Figure 13-11. Interrupt Request Acceptance Timing (Example of MOV A,r)



If an interrupt request flag (××IF) is set before an instruction clock n (n = 4 to 10) under execution becomes n – 1, the interrupt is accepted after the instruction under execution completes. Figure 13-11 shows an example of the interrupt request acceptance timing for an 8-bit data transfer instruction MOV A,r. Since this instruction is executed for 4 clocks, if an interrupt occurs for 3 clocks after the execution starts, the interrupt acceptance processing is performed after the MOV A,r instruction is completed.

Figure 13-12. Interrupt Request Acceptance Timing (When Interrupt Request Flag Generates at the Last Clock During Instruction Execution)



If an interrupt request flag (××IF) is set at the last clock of the instruction, the interrupt acceptance processing starts after the next instruction is executed. Figure 17-12 shows an example of the interrupt acceptance timing for an interrupt request flag that is set at the second clock of NOP (2-clock instruction). In this case, the MOV A,r instruction after the NOP instruction is executed, and then the interrupt acceptance processing is performed.

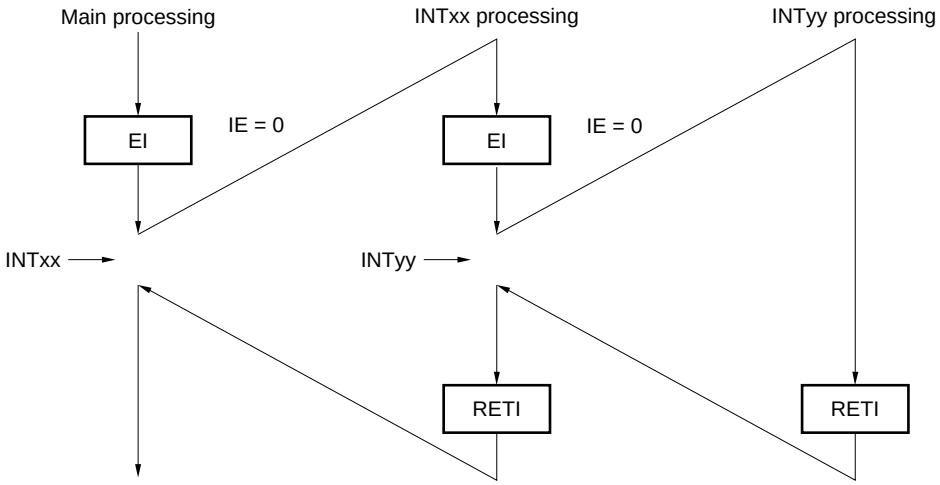
Caution Interrupt requests are reserved while interrupt request flag register 0 or 1 (IF0 or IF1) or the interrupt mask flag register 0 or 1 (MK0 or MK1) is being accessed.

13.4.3 Multiple interrupt processing

Multiple interrupt processing in which another interrupt is accepted while an interrupt is processed can be processed by priority. When two or more interrupts are generated at once, interrupt processing is performed according to the priority assigned to each interrupt request in advance (see **Table 13-1**).

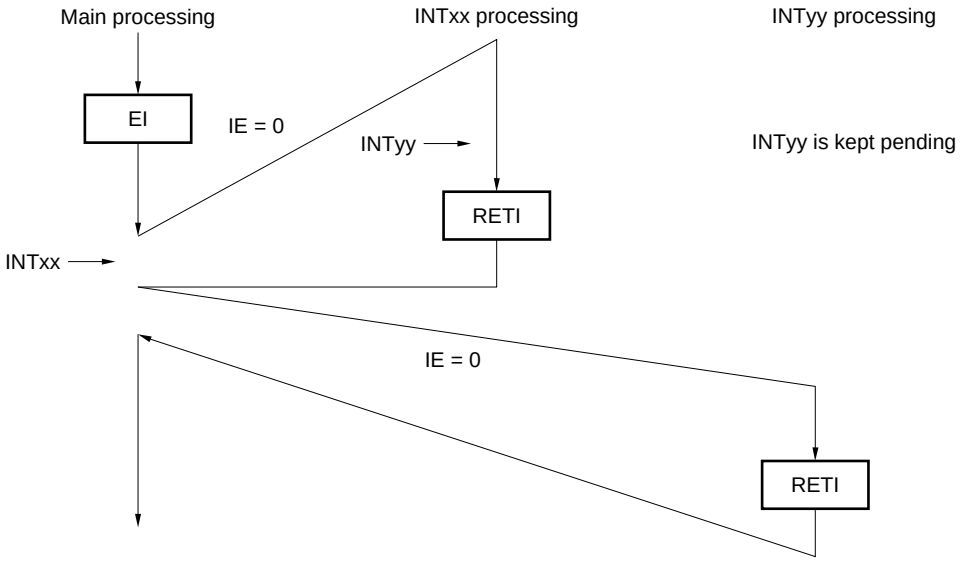
Figure 13-13. Example of Multiple Interrupt

Example 1. Multiple interrupt is accepted



During interrupt INTxx servicing, interrupt request INTyy is accepted, and a multiple interrupt is generated. An EI instruction is issued before each interrupt request acceptance, and the interrupt request acceptance enable state is set.

Example 2. A multiple interrupt is not generated because interrupts are not enabled



Because interrupts are not enabled in interrupt INTxx servicing (an EI instruction is not issued), interrupt request INTyy is not accepted, and a multiple interrupt is not generated. The INTyy request is reserved and accepted after the INTxx processing is performed.

IE = 0: Interrupt request acceptance disabled

13.4.4 Interrupt request reserve

Some instructions may reserve the acceptance of an instruction request until the completion of the execution of the next instruction even if the interrupt request (maskable interrupt, non-maskable interrupt, and external interrupt) is generated during the execution. The following shows such instructions (interrupt request reserve instruction).

- Manipulation instruction for the interrupt request flag registers 0 and 1 (IF0 and IF1)
- Manipulation instruction for the interrupt mask flag registers 0 and 1 (MK0 and MK1)

CHAPTER 14 STANDBY FUNCTION

14.1 Standby Function and Configuration

14.1.1 Standby function

The standby function is to reduce the power dissipation of the system and can be effected in the following two modes:

(1) HALT mode

This mode is set when the HALT instruction is executed. HALT mode stops the operation clock of the CPU. The system clock oscillation circuit continues oscillating. This mode does not reduce the current drain as much as STOP mode, but is useful for resuming processing immediately when an interrupt request is generated, or for intermittent operations.

(2) STOP mode

This mode is set when the STOP instruction is executed. STOP mode stops the main system clock oscillation circuit and stops the entire system. The current drain of the CPU can be substantially reduced in this mode.

The low voltage (V_{DD} = 1.8 V min.) of the data memory can be retained. Therefore, this mode is useful for retaining the contents of the data memory at an extremely low current drain.

STOP mode can be released by an interrupt request, so that this mode can be used for intermittent operation. However, some time is required until the system clock oscillation circuit settles after STOP mode has been released. If processing must be resumed immediately by using an interrupt request, therefore, use HALT mode.

In both modes, the previous contents of the registers, flags, and data memory before setting standby mode are all retained. In addition, the statuses of the output latch of the I/O ports and output buffer are also retained.

Caution To set STOP mode, be sure to stop the operations of the peripheral hardware, and then execute the STOP instruction.

14.1.2 Standby function control register

The wait time after STOP mode is released upon interrupt request until the oscillation stabilizes is controlled with the oscillation stabilization time selection register (OSTS).

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

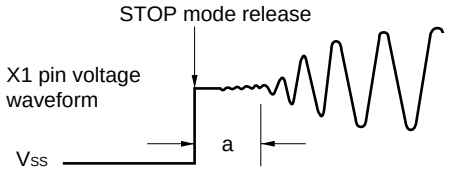
$\overline{\text{RESET}}$ input sets OSTS to 04H. However, the oscillation stabilization time after $\overline{\text{RESET}}$ input is $2^{15}/f_X$, instead of $2^{17}/f_X$.

Figure 14-1. Format of Oscillation Stabilization Time Selection Register

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	Oscillation stabilization time selection
0	0	0	$2^{12}/f_X$ (819 μs)
0	1	0	$2^{15}/f_X$ (6.55 ms)
1	0	0	$2^{17}/f_X$ (26.2 ms)
Other than above			Setting prohibited

Caution The wait time after STOP mode is released does not include the time from STOP mode release to clock oscillation start ("a" in the figure below), regardless of release by $\overline{\text{RESET}}$ input or by interrupt generation.



Remarks 1. f_X: System clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

14.2 Operation of Standby Function

14.2.1 HALT mode

(1) HALT mode

HALT mode is set by executing the HALT instruction.

The operation status in HALT mode is shown in the following table.

Table 14-1. Operation Statuses in HALT Mode

Item	HALT Mode Operation Status while the Main System Clock is Running
Main system clock generation circuit	Main system clock oscillation enabled
CPU	Operation disabled
Port (output latch)	Remains in the state existing before the selection of HALT mode
16-bit timer (TM90)	Operation enabled
8-bit timer/event counter (TM80)	Operation enabled
8-bit timer/event counter (TM81)	Operation enabled
8-bit timer (TM82)	Operation enabled
Watch timer	Operation enabled
Watchdog timer	Operation enabled
Serial interface 20	Operation enabled
A/D converter	Operation disabled
Multiplier	Operation disabled
External interrupt	Operation enabled[Note 1]

Notes 1. Maskable interrupt that is not masked

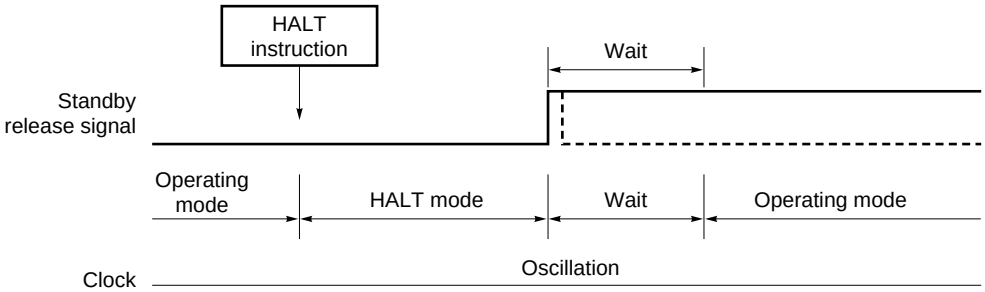
(2) Releasing HALT mode

HALT mode can be released by the following three types of sources:

(a) Releasing by unmasked interrupt request

HALT mode is released by an unmasked interrupt request. In this case, if the interrupt request is enabled to be accepted, vectored interrupt processing is performed. If the interrupt is disabled, the instruction at the next address is executed.

Figure 14-2. Releasing HALT Mode by Interrupt



Remarks 1. The broken line indicates the case where the interrupt request that has released standby mode is accepted.

2. The wait time is as follows:

- When vectored interrupt processing is performed: 9 to 10 clocks
- When vectored interrupt processing is not performed: 1 to 2 clocks

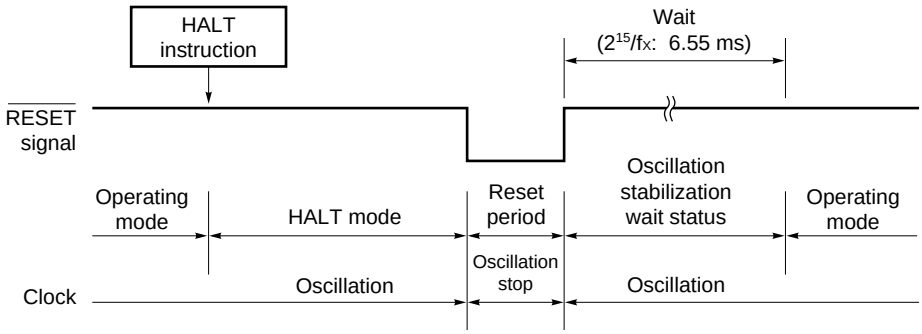
(b) Releasing by non-maskable interrupt request

HALT mode is released regardless of whether the interrupt is enabled or disabled, and vectored interrupt processing is performed.

(c) Releasing by $\overline{\text{RESET}}$ input

When HALT mode is released by the $\overline{\text{RESET}}$ signal, execution branches to the reset vector address in the same manner as the ordinary reset operation, and program execution is started.

Figure 14-3. Releasing HALT Mode by $\overline{\text{RESET}}$ Input



Remarks 1. fx: System clock oscillation frequency

2. The parenthesized values apply to operation at fx = 5.0 MHz.

Table 14-2. Operation after Release of HALT Mode

Releasing Source	MK××	IE	Operation
Maskable interrupt request	0	0	Executes next address instruction
	0	1	Executes interrupt processing
	1	×	Retains HALT mode
Non-maskable interrupt request	–	×	Executes interrupt processing
$\overline{\text{RESET}}$ input	–	–	Reset processing

×: Don't care

14.2.2 STOP mode

(1) Setting and operation status of STOP mode

STOP mode is set by executing the STOP instruction.

Caution Because standby mode can be released by an interrupt request signal, standby mode is released as soon as it is set if there is an interrupt source whose interrupt request flag is set and interrupt mask flag is reset. When STOP mode is set, therefore, HALT mode is set immediately after the STOP instruction has been executed, the wait time set by the oscillation stabilization time selection register (OSTS) elapses, and then operation mode is set.

The operation status in STOP mode is shown in the following table.

Table 14-3. Operation Statuses in STOP Mode

Item	STOP Mode Operation Status While the Main System Clock is Running
Clock generation circuit	Main system clock oscillation stopped
CPU	Operation disabled
Port (output latch)	Remains in the state existing before the selection of STOP mode

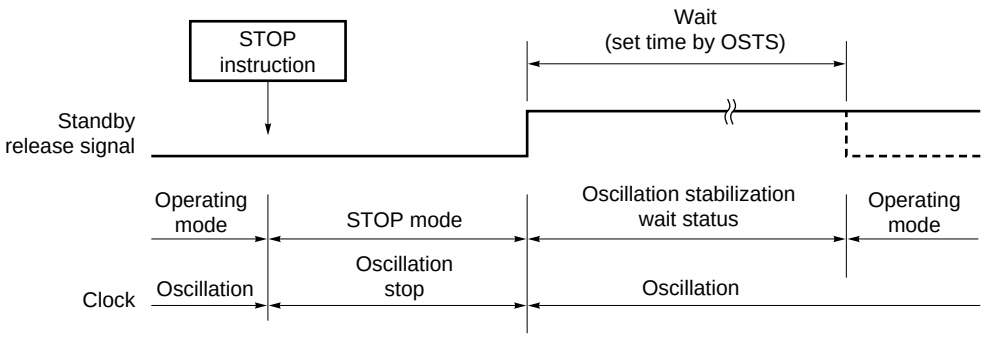
(2) Releasing STOP mode

STOP mode can be released by the following two types of sources:

(a) Releasing by unmasked interrupt request

STOP mode can be released by an unmasked interrupt request. In this case, if the interrupt is enabled to be accepted, vectored interrupt processing is performed, after the oscillation settling time has elapsed. If the interrupt acceptance is disabled, the instruction at the next address is executed.

Figure 14-4. Releasing STOP Mode by Interrupt

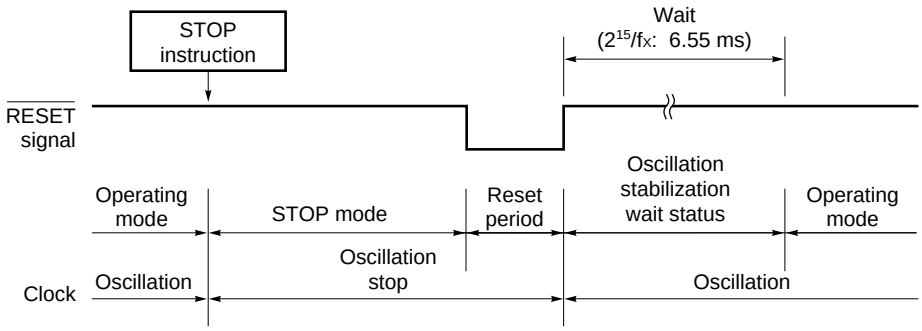


Remark The broken lines indicate the case where the interrupt request that has released standby mode is accepted.

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

When STOP mode is released by the $\overline{\text{RESET}}$ signal, the reset operation is performed after the oscillation settling time has elapsed.

Figure 14-5. Releasing STOP Mode by $\overline{\text{RESET}}$ Input



Remarks 1. fx: System clock oscillation frequency

2. The parenthesized values apply to operation at fx = 5.0 MHz.

Table 14-4. Operation after Release of STOP Mode

Releasing Source	MK××	IE	Operation
Maskable interrupt request	0	0	Executes next address instruction
	0	1	Executes interrupt processing
	1	×	Retains STOP mode
$\overline{\text{RESET}}$ input	–	–	Reset processing

×: Don't care

CHAPTER 15 RESET FUNCTION

The following two operations are available to generate reset signals.

(1) External reset input with $\overline{\text{RESET}}$ pin
(2) Internal reset by program run-away time detected with watchdog timer

External and internal reset have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H by reset signal input.

When a low level is input to the $\overline{\text{RESET}}$ pin or the watchdog timer overflows, a reset is applied and each hardware is set to the status shown in Table 19-1. Each pin has a high impedance during reset input or during oscillation settling time just after reset clear.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is cleared and program execution is started after the oscillation settling time has elapsed. The reset applied by the watchdog timer overflow is automatically cleared after reset, and program execution is started after the oscillation settling time has elapsed (see **Figures 15-2** through **15-4**).

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.

2. When STOP mode is cleared by reset, STOP mode contents are held during reset input. However, the port pins become high impedance.

Figure 151. Block Diagram of Reset Function

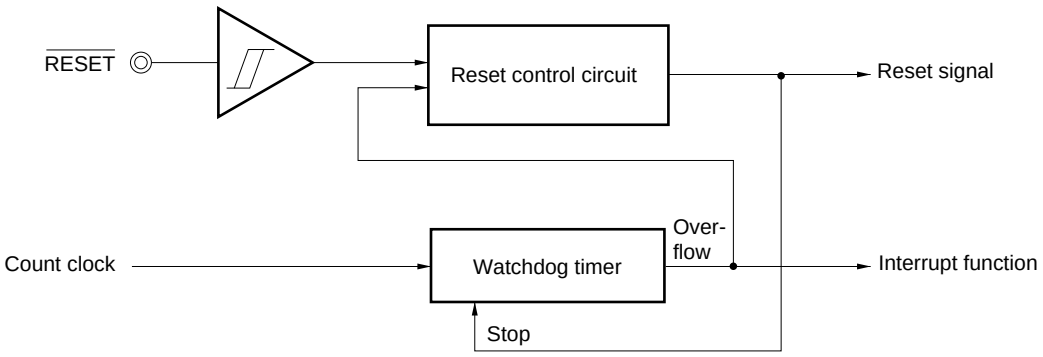


Figure 15-2. Reset Timing by $\overline{\text{RESET}}$ Input

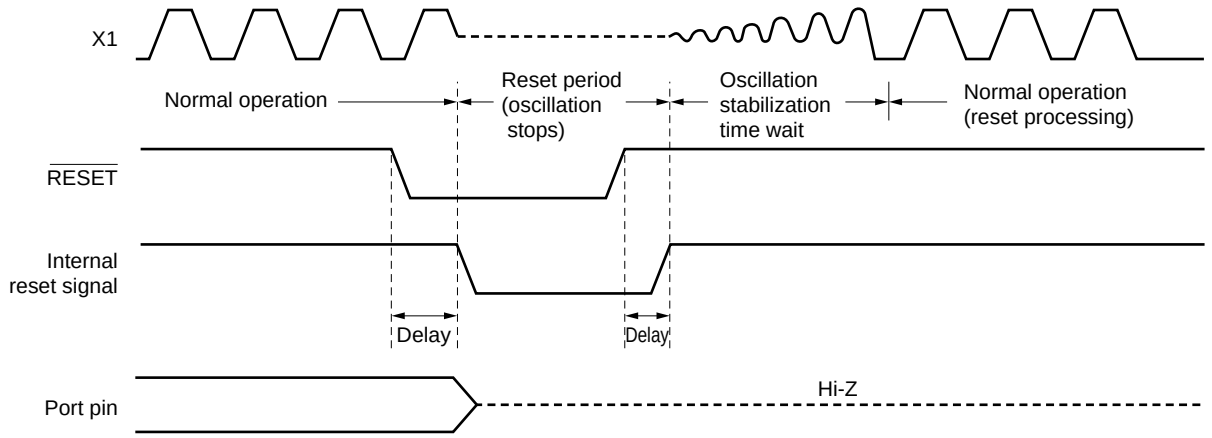


Figure 15-3. Reset Timing by Overflow in Watchdog Timer

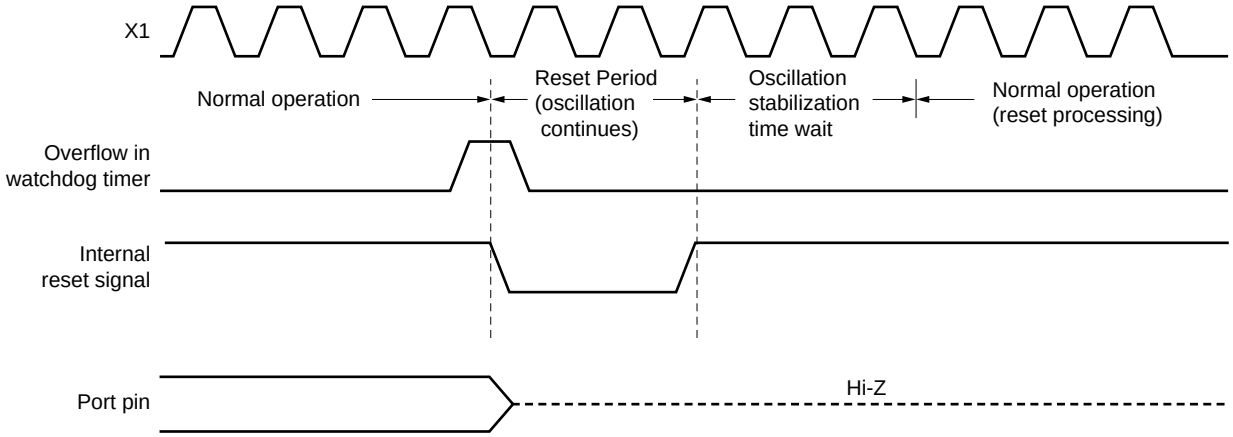


Figure 15-4. Reset Timing by $\overline{\text{RESET}}$ Input in STOP Mode

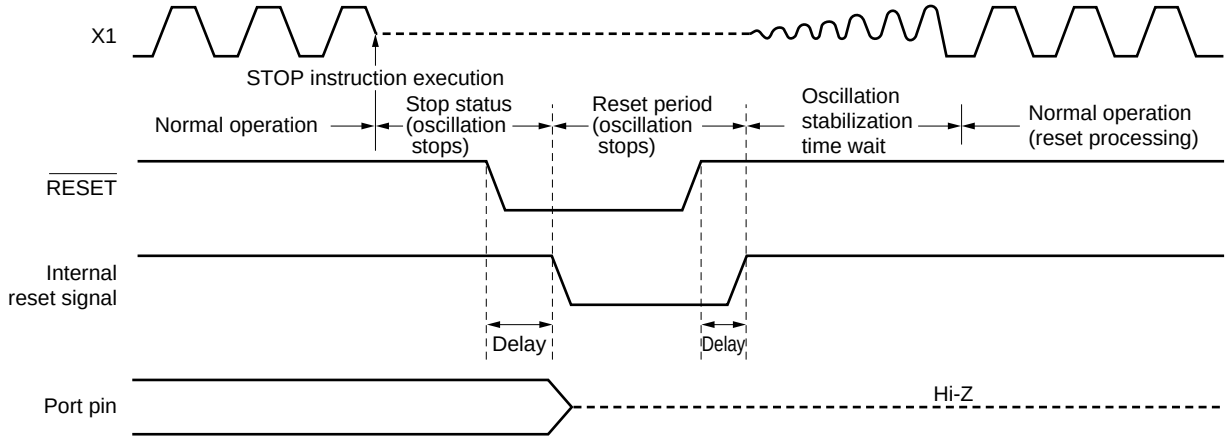


Table 15-1. State of the Hardware after a Reset (1/2)

Hardware		State after Reset
Program counter (PC)[Note 1]		Loaded with the contents of the reset vector table (0000H, 0001H)
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose register	Undefined[Note 2]
Ports (P0 to P3, P5, P6) (output latch)		00H
Port mode registers (PM0 to PM3, PM5)		FFH
Pull-up resistor option registers (PU0, PUB2, PUB3)		00H
Processor clock control register (PCC)		02H
Suboscillation mode register (SCKM)		00H
Subclock control register (CSS)		00H
Oscillation stabilization time selection register (OSTS)		04H
16-bit timer 90	Timer counter (TM90)	0000H
	Compare register (CR90)	FFFFH
	Capture register (TCP90)	Undefined
	Mode control register (TMC90)	00H
	Buzzer output control register (BZC90)	00H
8-bit timer/event counters 80 to 82	Timer counters (TM80 to TM82)	00H
	Compare registers (CR80 to CR82)	Undefined
	Mode control registers (TMC80 to TMC82)	00H
Watch timer	Mode control register (WTM)	00H
Watchdog timer	Timer clock selection register (TCL2)	00H
	Mode register (WDTM)	00H
A/D converter	Mode register (ADM0)	00H
	A/D input selection register (ADS0)	00H
	A/D conversion result register (ADCR0)	Undefined
Serial interface 20	Mode register (CSIM20)	00H
	Asynchronous serial interface mode register (ASIM20)	00H
	Asynchronous serial interface status register (ASIS20)	00H
	Baud rate generator control register (BRGC20)	00H
	Transmission shift register (TXS20)	FFH
	Reception buffer register (RXB20)	Undefined

Notes 1. While a reset signal is being input, and during the oscillation stabilization period, the contents of the PC will be undefined, while the remainder of the hardware will be the same as after the reset.

2. In standby mode, the RAM enters the hold state after a reset.

Table 15-1. State of the Hardware after a Reset (2/2)

Hardware		State after Reset
Multiplier	16-bit multiplication result storage register (MUL0)	Undefined
	Multiplication data registers (MRA0, MRB0)	Undefined
	Multiplier control register (MULC0)	00H
Interrupts	Request flag registers (IF0, IF1)	00H
	Mask flag registers (MK0, MK1)	FFH
	External interrupt mode registers (INTM0, INTM1)	00H

CHAPTER 16 μPD78F9189

The μPD78F9189 is flash memory versions of the μPD789188. The μPD78F9189 replaces the internal ROM of the μPD789188 with flash memory.

Table 16-1. Differences between μPD78F9189 and Mask ROM Versions

Item \ Product Name		Flash Memory Version	Mask ROM Version
		μPD78F9189	μPD789188
Internal memory	ROM	24 KB	16 KB
	High-speed RAM	512 bytes	
V_{PP} pin		Provided	Not provided
Pull-up resistor		17 (Software control)	21 (Software control: 17, mask option specification: 4)
A/D resolution		8 bits	8 bits
Electrical specifications		See the relevant data sheet	

16.1 Flash Memory Programming

The on-chip program memory in the μPD78F9189 is a flash memory.

The flash memory can be written with the μPD78F9189 mounted on the target system (on-board). Connect the dedicated flash writer (Flashpro III (part number: FL-PR3, PG-FP3)) to the host machine and target system to write the flash memory.

Remark FL-PR3 is made by Naito Densei Machida Mfg. Co., Ltd.

16.1.1 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 16-2. To select a communication mode, the format shown in Figure 16-2 is used. Each communication mode is selected by the number of V_{PP} pulses shown in Table 20-2.

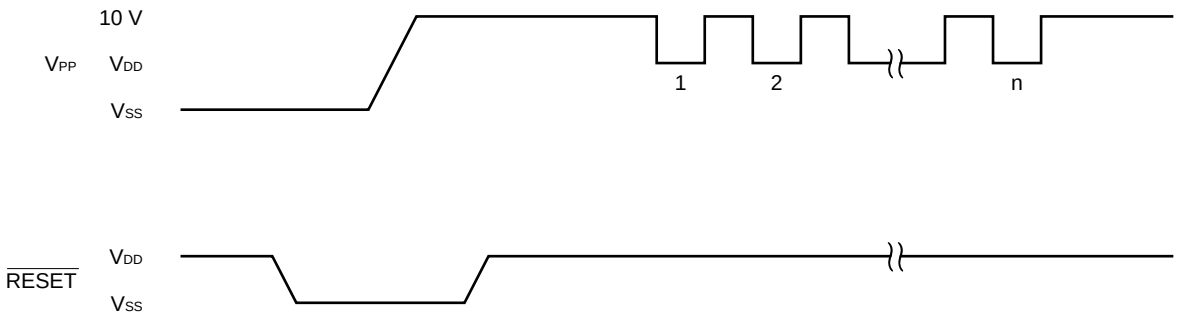
Table 16-2. Communication Mode

Communication Mode	Pins Used	Number of V_{PP} Pulses
3-wire serial I/O	$\overline{SCK20}$/ASCK20/P20 SO20/TxD20/P21 SI20/RxD20/P22	0
UART	TxD20/SO20/P21 RxD20/SI20/P22	8
Pseudo 3-wire mode[Note 2]	P00 (Serial clock input) P01 (Serial data input) P02 (Serial data output)	12

Notes 1. Serial transfer is performed by controlling a port by software.

Caution Be sure to select a communication mode depending on the V_{PP} pulse number shown in Table 16-2.

Figure 16-1. Format of Communication Mode Selection



16.1.2 Function of flash memory programming

By transmitting/receiving commands and data in the selected communication mode, operations such as writing to the flash memory are performed. Table 20-3 shows the major functions of flash memory programming.

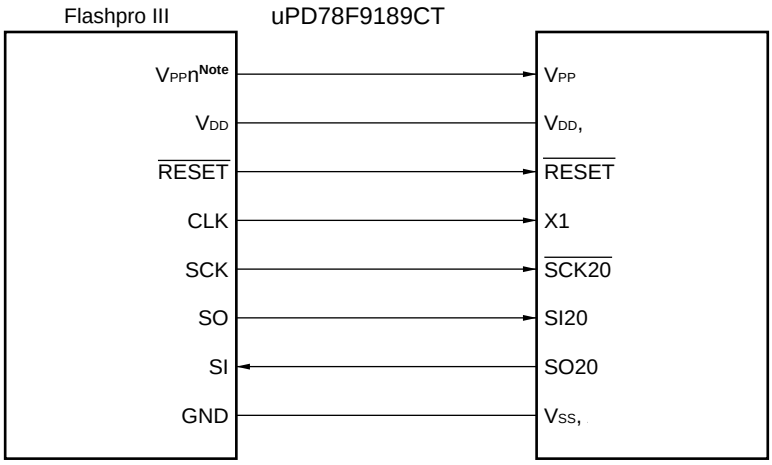
Table 16-3. Major Functions of Flash Memory Programming

Function	Description
Batch erase	Erases all contents of memory
Batch blank check	Checks erased state of entire memory
Data write	Write to flash memory based on write start address and number of data written (number of bytes)
Batch verify	Compares all contents of memory with input data

16.1.3 Flashpro III connection

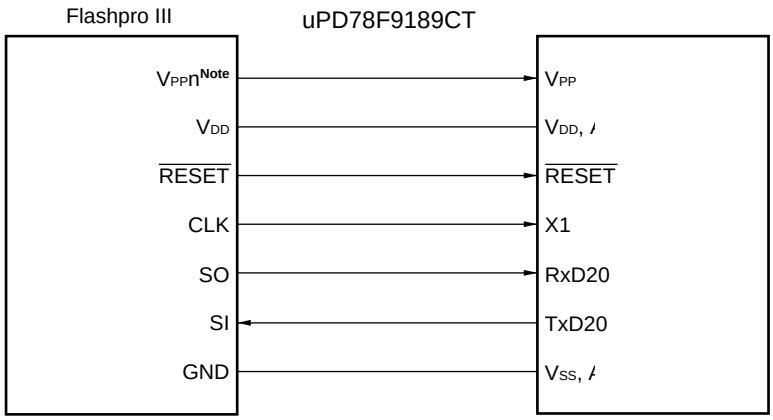
Connection between the Flashpro III and the μPD78F9189 differs depending on the communication mode (3-wire serial I/O, UART, or pseudo 3-wire mode). Figures 20-2 to 20-5 show the connection in the respective modes.

Figure 16-2. Flashpro III Connection in 3-Wire Serial I/O Mode



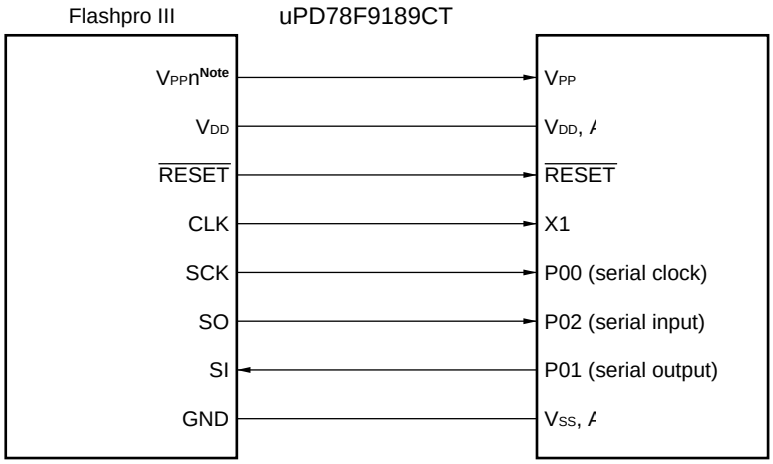
Note n = 1, 2

Figure 16-4. Flashpro III Connection in UART Mode



Note n = 1, 2

Figure 16-5. Flashpro III Connection in Pseudo 3-Wire Mode (When P0 is Used)



Note n = 1, 2

16.1.4 Setting with Flashpro III (PG-FP3)

When writing data to flash memory by using Flashpro III (PG-FP3), set the following.

<1> Load the parameter file.

<2> Use the type command to select a serial mode and a serial clock.

<3> An example of setting with PG-FP3 is shown below.

Table 16-4. Setting with PG-FP3

Communication Mode	Setting with PG-FP3		V_{PP} Pulse Count[Note 1]
3-wire serial I/O	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	
UART	COMM PORT	UART ch-0	8
	CPU CLK	On Target Board	
	On Target Board	4.1943 MHz	
	UART BPS	9,600 bps[Note 4]	
Pseudo 3-wire mode	COMM PORT	Port A	12
	CPU CLK	On Target Board	
		In Flashpro	
	On Target Board	4.1943 MHz	
	SIO CLK	1.0 kHz	
	In Flashpro	4.0 MHz	
	SIO CLK	1.0 kHz	

Notes 1. The number of V_{PP} pulses supplied from the Flashpro III when serial communication is initialized. These pulse counts determine the pins used for communication.

2. Select 4.0 MHz or 3.125 MHz.

3. Select 9,600 bps, 19,200 bps, 38,400 bps, or 76,800 bps.

Remark COMM PORT: Selects serial port.
SIO CLK: Selects serial clock frequency.
CPU CLK: Selects source of CPU clock to be input.

[MEMO]

CHAPTER 17 MASK OPTION

Table17-1. Selection of Mask Option for Pins

Pin	Mask Option
P50 to P53	Whether a pull-up resistor is to be incorporated can be specified in 1-bit units.

For P50 to P53 (port 5), whether a pull-up resistor is to be incorporated can be specified by a mask option. The mask option is specified in 1-bit units.

Caution The flash memory versions do not provide the pull-up resistor incorporation function by a mask option.

[MEMO]

CHAPTER 18 INSTRUCTION SET

This chapter lists the instruction set of the μPD789188 and 78F9189 Subseries. For details of the operation and machine language (instruction code) of each instruction, refer to **78K/0S Series User's Manual — Instruction (U11047E)**.

18.1 Operation

18.1.1 Operand identifiers and description methods

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

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

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

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

Table 18-1. Operand Identifiers and Description Methods

Identifier	Description Method
r rp sfr	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special-function register symbol
saddr saddrp	FE20H to FF1FH Immediate data or labels FE20H to FF1FH Immediate data or labels (even addresses only)
addr16 addr5	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions) 0040H to 007FH Immediate data or labels (even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label

Remark See **Table 3-3 Special Function Registers** for symbols of 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
IE:	Interrupt request enable flag
NMIS:	Flag indicating non-maskable interrupt servicing in progress
( ):	Memory contents indicated by address or register contents in parenthesis
×H, ×L:	Higher 8 bits and lower 8 bits of 16-bit register
∧:	Logical product (AND)
∨:	Logical sum (OR)
∀:	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" column

(Blank):	Unchanged
0:	Cleared to 0
1:	Set to 1
×:	Set/cleared according to the result
R:	Previously saved value is stored

18.2 Operation List

Mnemonic	Operands	Byte	Clock	Operation	Flag
					Z AC CY
MOV	r, #byte	3	6	r ← byte	
	saddr, #byte	3	6	(saddr) ← byte	
	sfr, #byte	3	6	sfr ← byte	
	A, r Note 1	2	4	A ← r	
	r, A Note 1	2	4	r ← A	
	A, saddr	2	4	A ← (saddr)	
	saddr, A	2	4	(saddr) ← A	
	A, sfr	2	4	A ← sfr	
	sfr, A	2	4	sfr ← A	
	A, !addr16	3	8	A ← (addr16)	
	!addr16, A	3	8	(addr16) ← A	
	PSW, #byte	3	6	PSW ← byte	× × ×
	A, PSW	2	4	A ← PSW	
	PSW, A	2	4	PSW ← A	× × ×
	A, [DE]	1	6	A ← (DE)	
	[DE], A	1	6	(DE) ← A	
	A, [HL]	1	6	A ← (HL)	
	[HL], A	1	6	(HL) ← A	
	A, [HL + byte]	2	6	A ← (HL + byte)	
	[HL + byte], A	2	6	(HL + byte) ← A	
XCH	A, X	1	4	A ↔ X	
	A, r Note 2	2	6	A ↔ r	
	A, saddr	2	6	A ↔ (saddr)	
	A, sfr	2	6	A ↔ sfr	
	A, [DE]	1	8	A ↔ (DE)	
	A, [HL]	1	8	A ↔ (HL)	
	A, [HL + byte]	2	8	A ↔ (HL + byte)	

Notes 1. Except r = A.

2. Except r = A, X.

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by processor clock control register (PCC).

Mnemonic	Operands	Byte	Clock	Operation	Flag		
					Z	AC	CY
MOVW	rp, #word	3	6	rp ← word			
	AX, saddrp	2	6	AX ← (saddrp)			
	saddrp, AX	2	8	(saddrp) ← AX			
	AX, rp Note	1	4	AX ← rp			
	rp, AX Note	1	4	rp ← AX			
XCHW	AX, rp Note	1	8	AX ↔ rp			
ADD	A, #byte	2	4	A, CY ← A + byte	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte	×	×	×
	A, r	2	4	A, CY ← A + r	×	×	×
	A, saddr	2	4	A, CY ← A + (saddr)	×	×	×
	A, !addr16	3	8	A, CY ← A + (addr16)	×	×	×
	A, [HL]	1	6	A, CY ← A + (HL)	×	×	×
	A, [HL + byte]	2	6	A, CY ← A + (HL + byte)	×	×	×
ADDC	A, #byte	2	4	A, CY ← A + byte + CY	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte + CY	×	×	×
	A, r	2	4	A, CY ← A + r + CY	×	×	×
	A, saddr	2	4	A, CY ← A + (saddr) + CY	×	×	×
	A, !addr16	3	8	A, CY ← A + (addr16) + CY	×	×	×
	A, [HL]	1	6	A, CY ← A + (HL) + CY	×	×	×
	A, [HL + byte]	2	6	A, CY ← A + (HL + byte) + CY	×	×	×
SUB	A, #byte	2	4	A, CY ← A – byte	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte	×	×	×
	A, r	2	4	A, CY ← A – r	×	×	×
	A, saddr	2	4	A, CY ← A – (saddr)	×	×	×
	A, !addr16	3	8	A, CY ← A – (addr16)	×	×	×
	A, [HL]	1	6	A, CY ← A – (HL)	×	×	×
	A, [HL + byte]	2	6	A, CY ← A – (HL + byte)	×	×	×

Note Only when rp = BC, DE, or HL.

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by processor clock control register (PCC).

Mnemonic	Operands	Byte	Clock	Operation	Flag		
					Z	AC	CY
SUBC	A, #byte	2	4	A, CY ← A – byte – CY	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte – CY	×	×	×
	A, r	2	4	A, CY ← A – r – CY	×	×	×
	A, saddr	2	4	A, CY ← A – (saddr) – CY	×	×	×
	A, !addr16	3	8	A, CY ← A – (addr16) – CY	×	×	×
	A, [HL]	1	6	A, CY ← A – (HL) – CY	×	×	×
	A, [HL + byte]	2	6	A, CY ← A – (HL + byte) – CY	×	×	×
AND	A, #byte	2	4	A ← A ∧ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ∧ byte	×		
	A, r	2	4	A ← A ∧ r	×		
	A, saddr	2	4	A ← A ∧ (saddr)	×		
	A, !addr16	3	8	A ← A ∧ (addr16)	×		
	A, [HL]	1	6	A ← A ∧ (HL)	×		
	A, [HL + byte]	2	6	A ← A ∧ (HL + byte)	×		
OR	A, #byte	2	4	A ← A ∨ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ∨ byte	×		
	A, r	2	4	A ← A ∨ r	×		
	A, saddr	2	4	A ← A ∨ (saddr)	×		
	A, !addr16	3	8	A ← A ∨ (addr16)	×		
	A, [HL]	1	6	A ← A ∨ (HL)	×		
	A, [HL + byte]	2	6	A ← A ∨ (HL + byte)	×		
XOR	A, #byte	2	4	A ← A ▽ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ▽ byte	×		
	A, r	2	4	A ← A ▽ r	×		
	A, saddr	2	4	A ← A ▽ (saddr)	×		
	A, !addr16	3	8	A ← A ▽ (addr16)	×		
	A, [HL]	1	6	A ← A ▽ (HL)	×		
	A, [HL + byte]	2	6	A ← A ▽ (HL + byte)	×		

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by processor clock control register (PCC).

Mnemonic	Operands	Byte	Clock	Operation	Flag		
					Z	AC	CY
CMP	A, #byte	2	4	A – byte	×	×	×
	saddr, #byte	3	6	(saddr) – byte	×	×	×
	A, r	2	4	A – r	×	×	×
	A, saddr	2	4	A – (saddr)	×	×	×
	A, !addr16	3	8	A – (addr16)	×	×	×
	A, [HL]	1	6	A – (HL)	×	×	×
	A, [HL + byte]	2	6	A – (HL + byte)	×	×	×
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	×	×	×
INC	r	2	4	r ← r + 1	×	×	
	saddr	2	4	(saddr) ← (saddr) + 1	×	×	
DEC	r	2	4	r ← r + 1	×	×	
	saddr	2	4	(saddr) ← (saddr) – 1	×	×	
INCW	rp	1	4	rp ← rp + 1			
DECW	rp	1	4	rp ← rp – 1			
ROR	A, 1	1	2	(CY, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$) × 1			×
ROL	A, 1	1	2	(CY, $A_0 \leftarrow A_7$, $A_{m+1} \leftarrow A_m$) × 1			×
RORC	A, 1	1	2	(CY ← A_0, $A_7 \leftarrow$ CY, $A_{m-1} \leftarrow A_m$) × 1			×
ROLC	A, 1	1	2	(CY ← A_7, $A_0 \leftarrow$ CY, $A_{m+1} \leftarrow A_m$) × 1			×
SET1	saddr.bit	3	6	(saddr.bit) ← 1			
	sfr.bit	3	6	sfr.bit ← 1			
	A.bit	2	4	A.bit ← 1			
	PSW.bit	3	6	PSW.bit ← 1	×	×	×
	[HL].bit	2	10	(HL).bit ← 1			
CLR1	saddr.bit	3	6	(saddr.bit) ← 0			
	sfr.bit	3	6	sfr.bit ← 0			
	A.bit	2	4	A.bit ← 0			
	PSW.bit	3	6	PSW.bit ← 0	×	×	×
	[HL].bit	2	10	(HL).bit ← 0			
SET1	CY	1	2	CY ← 1			1
CLR1	CY	1	2	CY ← 0			0
NOT1	CY	1	2	CY ← $\overline{CY}$			×

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by processor clock control register (PCC).

Mnemonic	Operands	Byte	Clock	Operation	Flag		
					Z	AC	CY
CALL	!addr16	3	6	$(SP - 1) \leftarrow (PC + 3)_H$, $(SP - 2) \leftarrow (PC + 3)_L$, $PC \leftarrow$ addr16, $SP \leftarrow SP - 2$			
CALLT	[addr5]	1	8	$(SP - 1) \leftarrow (PC + 1)_H$, $(SP - 2) \leftarrow (PC + 1)_L$, $PC_H \leftarrow$ (00000000, addr5 + 1), $PC_L \leftarrow$ (00000000, addr5), $SP \leftarrow SP - 2$			
RET		1	6	$PC_H \leftarrow (SP + 1)$, $PC_L \leftarrow (SP)$, $SP \leftarrow SP + 2$			
RETI		1	8	$PC_H \leftarrow (SP + 1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP + 2)$, $SP \leftarrow SP + 3$, NMIS $\leftarrow 0$	R	R	R
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	4	$PSW \leftarrow (SP)$, $SP \leftarrow SP + 1$	R	R	R
	rp	1	6	$rp_H \leftarrow (SP + 1)$, $rp_L \leftarrow (SP)$, $SP \leftarrow SP + 2$			
MOVW	SP, AX	2	8	$SP \leftarrow AX$			
	AX, SP	2	6	$AX \leftarrow SP$			
BR	!addr16	3	6	$PC \leftarrow$ addr16			
	$addr16	2	6	$PC \leftarrow PC + 2 +$ jdisp8			
	AX	1	6	$PC_H \leftarrow A$, $PC_L \leftarrow X$			
BC	$saddr16	2	6	$PC \leftarrow PC + 2 +$ jdisp8 if CY = 1			
BNC	$saddr16	2	6	$PC \leftarrow PC + 2 +$ jdisp8 if CY = 0			
BZ	$saddr16	2	6	$PC \leftarrow PC + 2 +$ jdisp8 if Z = 1			
BNZ	$saddr16	2	6	$PC \leftarrow PC + 2 +$ jdisp8 if Z = 0			
BT	saddr.bit, $addr16	4	10	$PC \leftarrow PC + 4 +$ jdisp8 if (saddr.bit) = 1			
	sfr.bit, $addr16	4	10	$PC \leftarrow PC + 4 +$ jdisp8 if sfr.bit = 1			
	A.bit, $addr16	3	8	$PC \leftarrow PC + 3 +$ jdisp8 if A.bit = 1			
	PSW.bit, $addr16	4	10	$PC \leftarrow PC + 4 +$ jdisp8 if PSW.bit = 1			
BF	saddr.bit, $addr16	4	10	$PC \leftarrow PC + 4 +$ jdisp8 if (saddr.bit) = 0			
	sfr.bit, $addr16	4	10	$PC \leftarrow PC + 4 +$ jdisp8 if sfr.bit = 0			
	A.bit, $addr16	3	8	$PC \leftarrow PC + 3 +$ jdisp8 if A.bit = 0			
	PSW.bit, $addr16	4	10	$PC \leftarrow PC + 4 +$ jdisp8 if PSW.bit = 0			
DBNZ	B, $addr16	2	6	$B \leftarrow B - 1$, then $PC \leftarrow PC + 2 +$ jdisp8 if $B \neq 0$			
	C, $addr16	2	6	$C \leftarrow C - 1$, then $PC \leftarrow PC + 2 +$ jdisp8 if $C \neq 0$			
	saddr, $addr16	3	8	$(saddr) \leftarrow (saddr) - 1$, then $PC \leftarrow PC + 3 +$ jdisp8 if $(saddr) \neq 0$			
NOP		1	2	No Operation			
EI		3	6	IE $\leftarrow$ 1 (Enable Interrupt)			
DI		3	6	IE $\leftarrow$ 0 (Disable Interrupt)			
HALT		1	2	Set HALT Mode			
STOP		1	2	Set STOP Mode			

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by processor clock control register (PCC).

18.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

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

1st Operand \ 2nd Operand	#byte	A	r	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte]

(2) 16-bit instructions

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

1st Operand \ 2nd Operand	#word	AX	rp[Note]	saddrp	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	
rp	MOVW	MOVW[Note]				INCW DECW PUSH POP
saddrp		MOVW				
sp		MOVW				

Note Only when rp = BC, DE, or HL.

(3) Bit manipulation instructions

SET1, CLR1, NOT1, BT, BF

1st Operand \ 2nd Operand	$addr16	None
A.bit	BT BF	SET1 CLR1
sfr.bit	BT BF	SET1 CLR1
saddr.bit	BT BF	SET1 CLR1
PSW.bit	BT BF	SET1 CLR1
[HL].bit		SET1 CLR1
CY		SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLT, BR, BC, BNC, BZ, BNZ, DBNZ

2nd Operand / 1st Operand	AX	!addr16	[addr5]	$addr16
Basic instructions	BR	CALL BR	CALLT	BR BC BNC BZ BNZ
Compound instructions				DBNZ

(5) Other instructions

RET, RETI, NOP, EI, DI, HALT, STOP

CHAPTER 19 ELECTRICAL SPECIFICATIONS

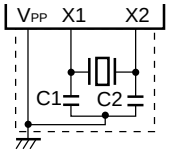
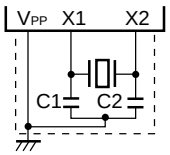
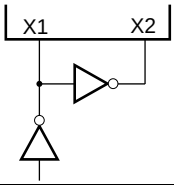
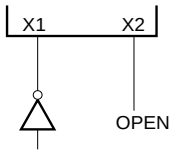
Absolute Maximum Ratings (T_A = 25°C)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.3 to +6.5	V
	V_{PP}		–0.3 to +10.5	V
Input voltage	V_{I1}	Pins other than P50 to P53, P23, P24	–0.3 to V_{DD} + 0.3	V
	V_{I2}	P23, P24	–0.3 to +5.5	V
	V_{I3}	P50 to P53	–0.3 to +13	V
Output voltage	V_O		–0.3 to V_{DD} + 0.3	V
Output current, high	I_{OH}	Per pin	–10	mA
		Total for all pins	–30	mA
Output current, low	I_{OL}	Per pin	30	mA
		Total for all pins	160	mA
Operating ambient temperature	T_A	In normal operation mode	–40 to +85	°C
		During flash memory programming	+10 to +40	°C
Storage temperature	T_{stg}		–40 to +125	°C

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 otherwise specified, the characteristics of alternate-function pins are the same as those of port pins.

Main System Clock Oscillator Characteristics (T_A = –40 to +85 °C, V_{DD} = 4.0 to 5.5 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		Oscillation frequency (f_X) **Note 1**	V_{DD} = oscillation voltage range	1.0		5.0	MHz
		Oscillation stabilization time **Note 2**	After V_{DD} reaches oscillation voltage range MIN.			4	ms
Crystal resonator		Oscillation frequency (f_X) **Note 1**		1.0		5.0	MHz
		Oscillation stabilization time **Note 2**				10	ms
External clock		X1 input frequency (f_X) **Note 1**		1.0		5.0	MHz
		X1 input high-/low-level width (t_{XH}, t_{XL})		85		500	ns
		X1 input frequency (f_X) **Note 1**		1.0		5.0	MHz
		X1 input high-/low-level width (t_{XH}, t_{XL})		85		500	ns

Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

2. Time required to stabilize oscillation after reset or STOP mode release. Use a resonator that stabilizes oscillation within the oscillation wait time.

Cautions 1. When using the main system clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the 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 V_{SS0}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

2. When the main system clock is stopped and the device is operating on the subsystem clock, wait until the oscillation stabilization time has been secured by the program before switching back to the main system clock.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high	I_{OH}	Per pin				–1	mA
		Total for all pins				–15	mA
Output current, low	I_{OL}	Per pin				10	mA
		Total for all pins				80	mA
Input voltage, high	V_{IH1}	P00 to P04, P10, P11,P60 to P63		0.7 V_{DD}		V_{DD}	V
	V_{IH2}	P50 to P53		0.7 V_{DD}		12	V
	V_{IH3}	RESET, P20 to P26, P30 to P33		0.8 V_{DD}		V_{DD}	V
	V_{IH4}	X1, X2		V_{DD} – 0.5		V_{DD}	V
Input voltage, low	V_{IL1}	P00 to P04, P10, P11,P60 to P63		0		0.3 V_{DD}	V
	V_{IL2}	P50 to P53		0		0.3 V_{DD}	V
	V_{IL3}	RESET,P20 to P26, P30 to P33		0		0.2 V_{DD}	V
	V_{IL4}	X1, X2		0		0.4	V
Output voltage, high	V_{OH}	Pins other than P23, P24, P50 to P53	I_{OH} = –1 mA	V_{DD} – 1.0			V
			I_{OH} = –100 μA	V_{DD} – 0.5			V
Output voltage, low	V_{OL1}	Pins other than P50 to P53	I_{OL} = 10 Ma			1.0	V
			I_{OL} = 400 μA			0.5	V
	V_{OL2}	P50 to P53	I_{OL} = 10 mA			1.0	V
			I_{OL} = 1.6 mA			0.4	V
Input leakage current, high	I_{LIH1}	$V_I = V_{DD}$	Pins other than P50 to P53 (N-ch open-drain) X1, X2			3	μA
	I_{LIH2}		X1, X2			20	μA
	I_{LIH3}	V_I = 12 V	P50 to P53 (N-ch open drain)			20	μA
Input leakage current, low	I_{LIL1}	V_I = 0 V	Pins other than P50 to P53 (N-ch open-drain) X1, X2			–3	μA
	I_{LIL2}		X1, X2			–20	μA
	I_{LIL3}		P50 to P53 (N-ch open drain)			–3 **Note**	μA
Output leakage current, high	I_{LOH}	$V_O = V_{DD}$				3	μA
Output leakage current, low	I_{LOL}	V_O = 0 V				–3	μA
Software pull-up resistor	R_1	V_I = 0 V, for pins other than P23, P24, and P50 to P53		50	100	200	kΩ

Note A low-level input leakage current of -60 μA(MAX.) flows only during the 1-cycle time after a read instruction is executed to P50 to P53 and P50 to P53 are set to input mode. At times other than this, -3 μA (MAX.) current flows.

Remark Unless otherwise specified, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (T_A = –40 to +85 °C, V_{DD} = 4.0 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Power supply current	I_{DD1} **Note 1**	5.0-MHz crystal oscillation operating mode (C1 = C2 = 22pF)	V_{DD} = **4.0 to 5.5** **Note 4**		5.0	15.0	mA
	I_{DD2} **Note 1**	5.0-MHz crystal oscillation HALT mode (C1 = C2 = 22pF)	V_{DD} = **4.0 to 5.5** **Note 4**		2.0	6.0	mA
	I_{DD6} **Note 2**	5.0-MHz crystal oscillation A/D operating mode (C1 = C2 = 22pF)	V_{DD} = **4.0 to 5.5** **Note 4**		6.0	17.0	mA

Notes 1. The ADCS0 (bit 7 of ADM0; A/D converter mode register 0) = 1, V_{DD}, and the port current (including the current flowing through the internal pull-up resistors) are not included.

2. The ADCS0 =1 and port current (including the current flowing through the internal pull-up resistors) are not included. Refer to the A/D converter characteristics for the current flowing through Vdd.

3. When the main system clock is stopped.

4. During high-speed mode operation (when the processor clock control register (PCC) is set to 00H.)

5. During low-speed mode operation (when PCC is set to 02H)

Remark Unless otherwise specified, the characteristics of alternate-function pins are the same as those of port pins.

AC Characteristics

(1) Basic operation (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Cycle time (minimum instruction execution time)	T_{CY}	Operation based on the main system clock	0.4		8	μs
TI80 and TI81 input frequency	f_{TI}		0		4	MHz
TI80 and TI81 input high-/low-level width	t_{TIH}, t_{TIL}		0.1			μs
Interrupt input high-/low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP3	10			μs
$\overline{\text{RESET}}$ input low-level width	t_{RSL}		10			μs
CPT90 input high-/low-level width	t_{CPH}, t_{CPL}		10			μs

(2) Serial interface (T_A = –40 to +85 °C, V_{DD} = 4.0 to 5.5 V)

(a) 3-wire serial I/O mode ($\overline{SCK20}$...Internal clock)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK20}$ cycle time	t_{KCY1}		800			ns
$\overline{SCK20}$ high-/low-level width	t_{KH1}, t_{KL1}		t_{KCY1}/2–50			ns
SI20 setup time (to $\overline{SCK20}$ ↑)	t_{SIK1}		150			ns
SI20 hold time (from $\overline{SCK20}$ ↑)	t_{KSI1}		400			ns
SO20 output delay time from $\overline{SCK20}$↓	t_{KSO1}	R = 1 kΩ, C = 100 pF **Note**	0		250	ns

Note R and C are the load resistance and load capacitance of the SO20 output line.

(b) 3-wire serial I/O mode ($\overline{SCK20}$...External clock)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK20}$ cycle time	t_{KCY2}		900			ns
$\overline{SCK20}$ high-/low-level width	t_{KH2}, t_{KL2}		400			ns
SI20 setup time (to $\overline{SCK20}$ ↑)	t_{SIK2}		100			ns
SI20 hold time (from $\overline{SCK20}$ ↑)	t_{KSI2}		400			ns
SO20 output delay time from $\overline{SCK20}$ ↓	t_{KSO2}	R = 1 kΩ, C = 100 pF **Note**	0		300	ns
SO20 setup time (when using $\overline{SS20}$, to $\overline{SS20}$ ↓)	t_{KAS2}				120	ns
SO20 disable time (when using $\overline{SS20}$, from $\overline{SS20}$ ↑)	t_{KDS2}				240	ns

Note R and C are the load resistance and load capacitance of the SO20 output line.

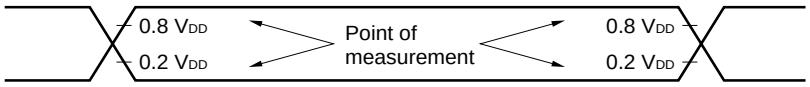
(c) UART mode (dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					78125	bps

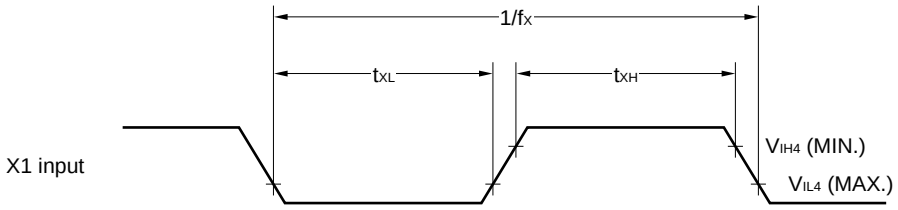
(d) UART mode (external clock input)

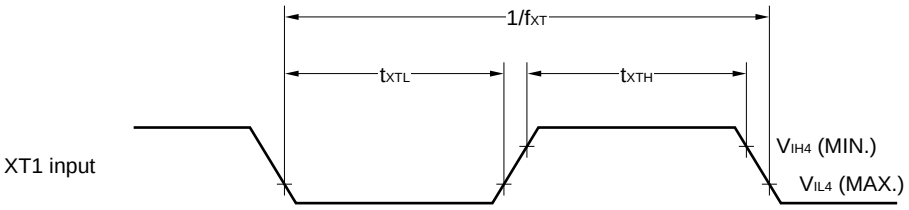
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
ASCK20 cycle time	t_{KCY3}		900			Ns
ASCK20 high-/low-level width	t_{KH3}, t_{KL3}		400			Ns
Transfer rate					39063	Bps
ASCK20 rise time, fall time	t_R, t_F				1	μs

AC Timing Measurement Points (excluding the X1 and XT1 inputs)

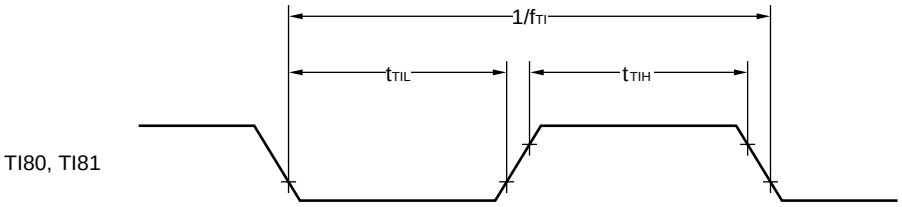


Clock Timing





TI Timing



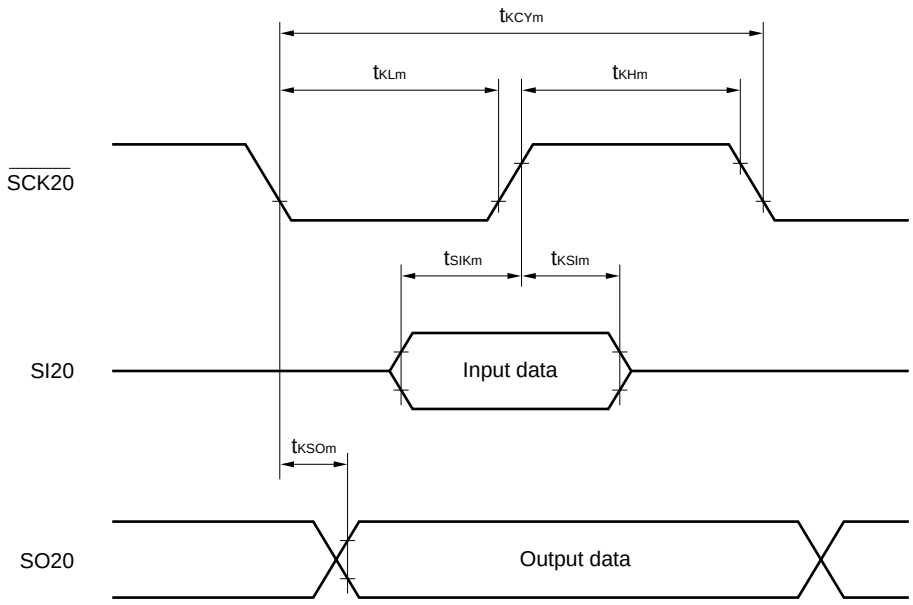
Interrupt Input Timing

$\overline{\text{RESET}}$ Input Timing

CPT90 Input Timing

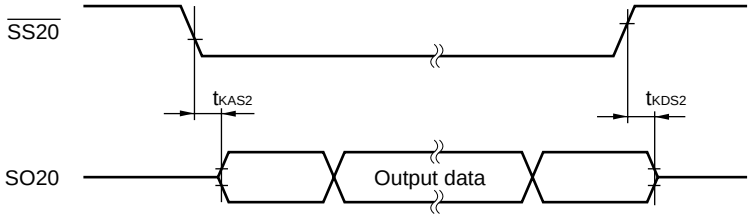
Serial Transfer Timing

3-wire serial I/O mode:

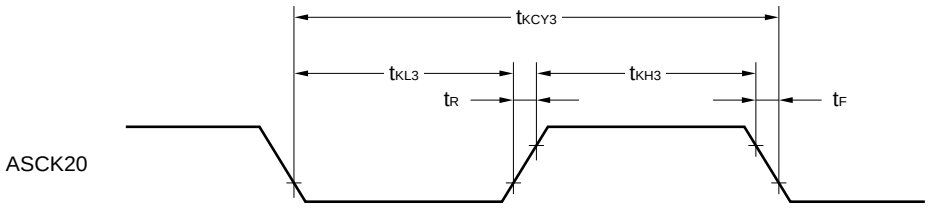


Remark m = 1, 2

3-wire serial I/O mode (when using SS20):



UART mode (external clock input):



8-Bit A/D Converter Characteristics (T_A = –40 to +85 °C, 4.0 ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error **Note**				±0.2	±0.4	%FSR
Conversion time	t_{CONV}		14		100	μs
Zero-scale error **Note**					±0.4	%FSR
Full-scale error **Note**					±0.4	%FSR
Integral linearity error **Note**	INL				±2.5	LSB
Differential linearity error **Note**	DNL				±1.5	LSB
Analog input voltage	V_{IAN}		0		Vdd	V
Reference voltage	AV_{REF}		4.5	Vdd	5.5	V
Resistance between Vdd and V_{SS}	R_{AIREF}		20	40		kΩ

Note Excludes quantization error (±0.05%FSR).

Remark FSR: Full scale range

FLASH MEMORY WRITE/DELETE CHARACTERISTICS (T_A = 10 to 40 °C, V_{DD} = 4.0 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Write current (V_{DD} pin) **Note**	I_{DDW}	When V_{PP} supply voltage = V_{PP1} (5.0-MHz crystal oscillation operation mode)			18	mA
Write current (V_{PP} pin) **Note**	I_{PPW}	When V_{PP} supply voltage = V_{PP1}			7.5	mA
Delete current (V_{DD} pin) **Note**	I_{DDE}	When V_{PP} supply voltage = V_{PP1} (5.0-MHz crystal oscillation operation mode)			18	mA
Delete current (V_{PP} pin) **Note**	I_{PPE}	When V_{PP} supply voltage = V_{PP1}			100	mA
Unit delete time	t_{er}		0.5	1	1	s
Total delete time	t_{era}				20	s
Write count		Delete/write are regarded as 1 cycle			20	Times
V_{PP} supply voltage	V_{PP0}	In normal operation	0		$0.2V_{DD}$	V
	V_{PP1}	During flash memory programming	9.7	10.0	10.3	V

Note The current flowing to the ports (including the current flowing through an on-chip pull-up resistor) and AV_{DD} current are not included.

Data Memory Stop Mode Low Power Supply Voltage Data Retention Characteristics (T_A = –40 to +85 °C)

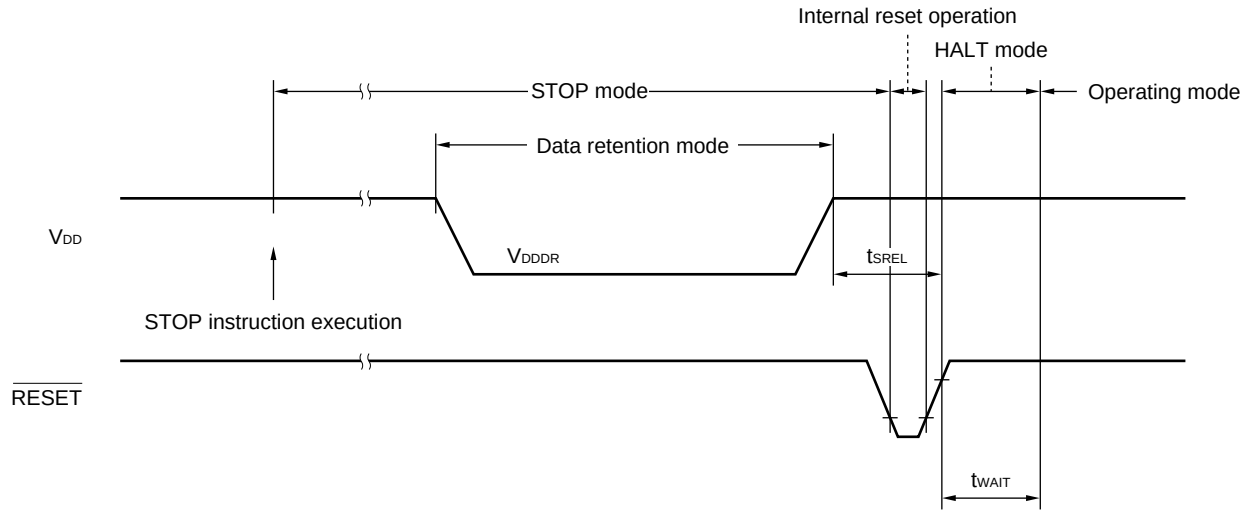
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention power supply voltage	V_{DDDR}		4.0		5.5	V
Release signal set time	t_{SREL}		0			μs
Oscillation stabilization wait time **Note 1**	t_{WAIT}	Release by $\overline{\text{RESET}}$		$2^{15}/f_X$		s
		Release by interrupt request		**Note 2**		s

Notes 1. The oscillation stabilization time is the time the CPU operation is stopped to prevent unstable operation when oscillation starts.

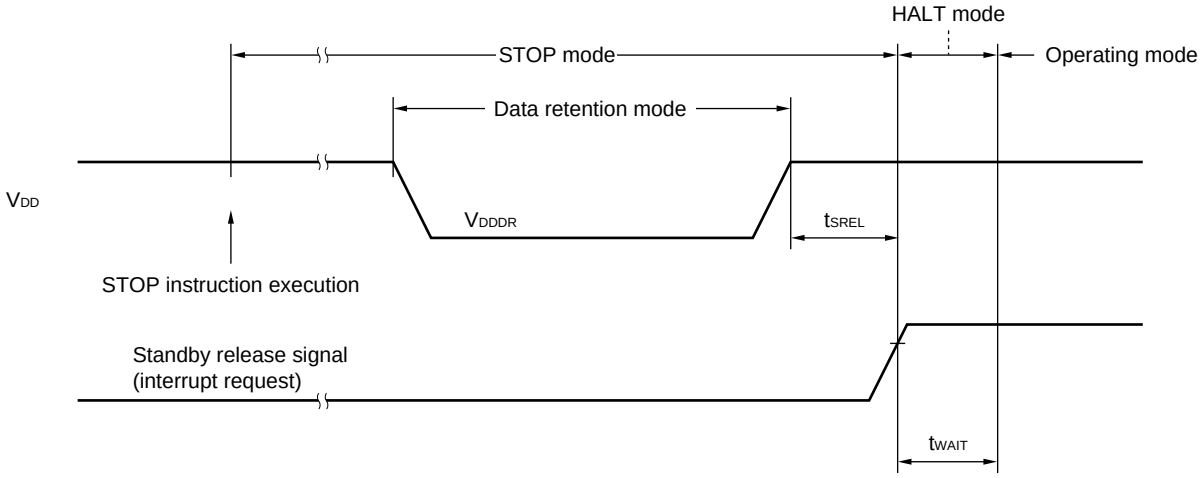
2. By using bits 0 to 2 (OSTS0 to OSTS2) of the oscillation stabilization time selection register (OSTS), $2^{12}/f_X$, $2^{15}/f_X$, or $2^{17}/f_X$ can be selected.

Remark f_X: Main system clock oscillation frequency

Data Retention Timing ($\overline{\text{STOP}}$ Mode Release by RESET)

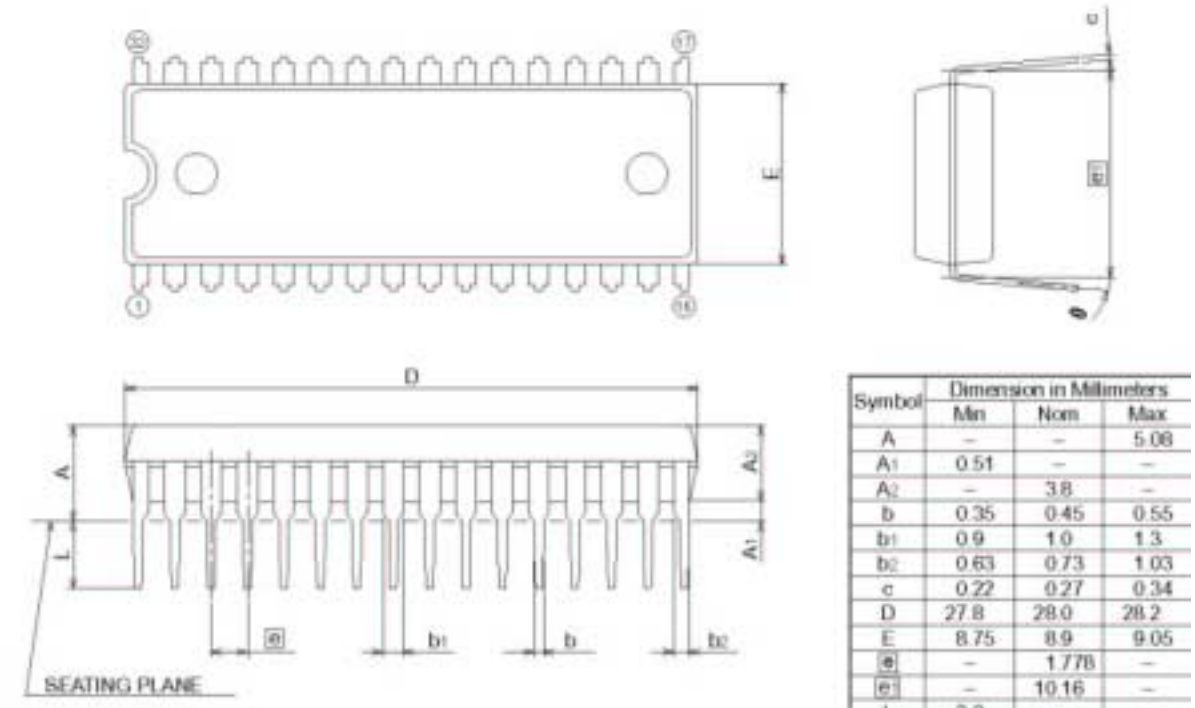


Data Retention Timing (Standby Release Signal: STOP Mode Release by Interrupt Signal)



CHAPTER 20 PACKAGE DRAWING

32 pin Shrink DIP(400mil)



Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	5.08
A_1	0.51	–	–
A_2	–	3.8	–
b	0.35	0.45	0.55
b_1	0.9	1.0	1.3
b_2	0.63	0.73	1.03
c	0.22	0.27	0.34
D	27.8	28.0	28.2
E	8.75	8.9	9.05
e	–	1.778	–
e_1	–	10.16	–
L	3.0	–	–
θ	0°	–	15°

CHAPTER 21 RECOMMENDED SOLDERING CONDITIONS

The μPD78F9189CT/μPD789188 should be soldered and mounted under the following recommended conditions.

For the 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 your NEC sales representative.

Table 21-1. Surface Mounting Type Soldering Conditions

μPD78F9189CT/μPD789188CT: 32-pin plastic shrink DIP (400mil)

Soldering Method	Soldering Conditions
Wave soldering	Solder bath temperature: 260 °C max., Time: 10 seconds max.
Partial heating	Pin temperature: 300 °C max., Time: 3 seconds max. (per each pin)

Caution Do not use different soldering methods together (except for partial heating).

APPENDIX A. DEVELOPMENT TOOLS

The following development tools are available for developing systems using the μPD78F9189 and μPD789188.

Language Processing Software

RA78K0S **Notes 1, 2, 3**	Assembler package common to 78K/0S Series
CC78K0S **Notes 1, 2, 3**	C compiler package common to 78K/0S Series
DF789189 **Notes 1, 2, 3**	Device file for μPD789188,and 78F9189Y Subseries
CC78K0S-L **Notes 1, 2, 3**	C compiler library source file common to 78K/0S Series

Flash Memory Writing Tools

Flashpro III (Part No. FL-PR3 **Note 4** , PG-FP3)	Flash programmer dedicated for on-chip flash memory microcontrollers

Debugging Tools(1/2)

IE-78K0S-NS In-circuit emulator	In-circuit emulator used to debug hardware or software when application systems which use the 78K/0S Series are developed. The IE-78K0S-NS supports an integrated debugger (ID78K0S-NS). The IE-78K0S-NS is used in combination with an interface adapter for connection to an AC adapter, emulation probe, or host machine.
IE-70000-MC-PS-B AC adapter	Adapter used to supply power from a 100- to 240-V AC outlet
IE-70000-98-IF-C Interface adapter	Adapter required when using the PC-9800 series (excluding notebook PCs) as the host machine for the IE-78K0S-NS (C bus supported)
IE-70000-CD-IF-A PC card/interface	PC card and interface cable required when using a notebook PC as the host machine for the IE-78K0S-NS (PCMCIA socket supported)
IE-70000-PC-IF-C Interface adapter	Adapter required when using an IBM PC/AT™ or compatible as the host machine for the IE-78K0S-NS (ISA bus supported)
IE-70000-PCI-IF Interface adapter	Adapter required when using a PC equipped with a PCI bus as the host machine for the IE-78K0S-NS
IE-789177-NS-EM1 Emulation board	Emulation board used to emulate the peripheral hardware specific to the device. This is used in combination with the in-circuit emulator.
NP-789188CT Emulation probe	Board to connect an in-circuit emulator to the target system.

Debugging Tools(2/2)

NP-789188CT Emulation probe	Board to connect an in-circuit emulator to the target system.
SM78K0S **Notes 1, 2**	System simulator common to 78K/0S Series
ID78K0S-NS **Notes 1, 2**	Integrated debugger common to 78K/0S Series
DF789189 **Notes 1, 2**	Device file for μPD789188 and 78F9189 Subseries

Real-Time OS

MX78K0S **Notes 1, 2**	OS for 78K/0S Series

Notes 1. Based on the PC-9800 series (Japanese Windows™)

2. Based on IBM PC/AT and compatibles (Japanese Windows/English Windows)

3. Based on the HP9000 series 700™ (HP-UX™), SPARCstation™ (SunOS™, Soraris™), and NEWS™ (NEWS-OS™)

4. Product made by and available from Naito Densei Machida Mfg. Co., Ltd. (+81-44-822-3813).

5. Product made by TOKYO ELETECH CORPORATION.
Refer to: Daimaru Kogyo, Ltd.
Tokyo Electronic Division (+81-3-3820-7112)
Osaka Electronic Division (+81-6-6244-6672)

Remark The RA78K0S, CC78K0S, and SM78K0S can be used in combination with the DF789177.

APPENDIX B. RELATED DOCUMENTS

Documents Related to Devices

Document Name	Document No.	
	Japanese	English
μPD789188, 78F9189 Subseries User's Manual	-	-
78K/0S Series Instruction User's Manual	U11047J	U11047E

Document Related to Development Tools (User's Manuals)

Document Name		Document No.	
		Japanese	English
RA78K0S Assembler Package	Operation	U11622J	U11622E
	Assembly Language	U11599J	U11599E
	Structured Assembly Language	U11623J	U11623E
CC78K0S C Compiler	Operation	U11816J	U11816E
	Language	U11817J	U11817E
SM78K0S System Simulator Windows based	Reference	U11489J	U11489E
SM78K Series System Simulator	External Parts User Open Interface Specifications	U10092J	U10092E
ID78K0S-NS Windows based	Reference	U12901J	U12901E
IE-78K0S-NS In-circuit Emulator		U13549J	U13549E
IE-789177-NS-EM1 Emulation Board		U14621J	U14621E

Documents Related to Embedded Software (User's Manuals)

Document Name		Document No.	
		Japanese	English
OS for 78K/0S Series MX78K0S	Fundamental	U12938J	U12938E

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

Other Documents

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

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

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

[MEMO]

NEC

Facsimile Message

From:

Name

Company

Tel. FAX

Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: 1-800-729-9288
1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Technical Documentation Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6465-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

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

Taiwan
NEC Electronics Taiwan Ltd.
Fax: 02-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: 044-548-7900

I would like to report the following error/make the following suggestion:

Document title: _____

Document number: _____ Page number: _____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	❑	❑	❑	❑
Technical Accuracy	❑	❑	❑	❑
Organization	❑	❑	❑	❑

CS 99.1