NEC

Preliminary User's Manual

78K0S/KA1+

8-Bit Single-Chip Microcontrollers

μPD78F9221
μPD78F9222

Document No. U16898EJ1V0UD00 (1st edition)
Date Published November 2003 N CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

Caution: This product uses SuperFlash® technology licensed from Silicon Storage Technology, inc.

- **The information contained in this document is being issued in advance of the production cycle for the product. The parameters for the product may change before final production or NEC Electronics Corporation, at its own discretion, may withdraw the product prior to its production.**
- **Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics 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 a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific". The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics products depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

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

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

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

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)

(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.

(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M5D 02. 11-1

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC Electronics product in your application, please contact the NEC Electronics 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.

[GLOBAL SUPPORT]
http://www.necel.com/en/support/support.html

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

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 01

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87
- **Succursale Française**
  Vélizy-Villacoublay, France
  Tel: 01-30-67 58 00
- **Filiale Italiana**
  Milano, Italy
  Tel: 02-66 75 41
- **Branch The Netherlands**
  Eindhoven, The Netherlands
  Tel: 040-244 58 45
- **Tyskland Filial**
  Taeby, Sweden
  Tel: 08-63 80 820
- **United Kingdom Branch**
  Milton Keynes, UK
  Tel: 01908-691-133

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-558-3737

NEC Electronics Shanghai, Ltd.
Shanghai, P.R. China
Tel: 021-6841-1138

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

NEC Electronics Singapore Pte. Ltd.
Novena Square, Singapore
Tel: 6253-8311

J03.4

INTRODUCTION

Target Readers

This manual is intended for user engineers who wish to understand the functions of the 78K0S/KA1+ in order to design and develop its application systems and programs.

The target devices are the following subseries products.

- 78K0S/KA1+: μPD78F9221, 78F9222

Purpose

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

Organization

Two manuals are available for the 78K0S/KA1+: this manual and the Instruction Manual (common to the 78K/0S Series).

78K0S/KA1+ User's Manual	78K/0S Series Instructions User's Manual
• Pin functions • Internal block functions • Interrupts • Other internal peripheral functions • Electrical specifications (target)	• CPU function • Instruction set • Instruction description

How to Use This Manual

It is assumed that the readers of this manual have general knowledge of electrical engineering, logic circuits, and microcontrollers.

◊ To understand the overall functions of 78K0S/KA1+
  → Read this manual in the order of the **CONTENTS**.

◊ How to read register formats
  → The name of a bit whose number is enclosed with <> is reserved in the assembler and is defined in the C compiler by the header file sfrbit.h.

◊ To learn the detailed functions of a register whose register name is known
  → See **APPENDIX C REGISTER INDEX**.

◊ To learn the details of the instruction functions of the 78K/0S Series
  → Refer to **78K/0S Series Instructions User's Manual (U11047E)** separately available.

◊ To learn the electrical specifications (target) of the 78K0S/KA1+
  → See **CHAPTER 20 ELECTRICAL SPECIFICATIONS (TARGET VALUES)**.

Conventions

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

Related Documents

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

Documents Related to Devices

Document Name	Document No.
78K0S/KA1+ Subseries User's Manual	This manual
78K/0S Series Instructions User's Manual	U11047E

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

Document Name		Document No.
RA78K0S Assembler Package	Operation	U14876E
	Language	U14877E
	Structured Assembly Language	U11623E
CC78K0S C Compiler	Operation	U14871E
	Language	U14872E
ID78K Series Integrated Debugger Ver. 2.30 or Later	Operation (Windows™ Based)	U15185E
Project Manager Ver. 3.12 or Later (Windows Based)		U14610E

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

Document Name	Document No.
IE-78K0S-NS In-Circuit Emulator	U13549E
IE-78K0S-NS-A In-Circuit Emulator	U15207E

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

Documents Related to Flash Memory Writing

Document Name	Document No.
PG-FP4 Flash Memory Programmer User's Manual	U15260E

Other Related Documents

Document Name	Document No.
SEMICONDUCTOR SELECTION GUIDE - Products and Packages -	X13769X
Semiconductor Device Mount Manual	**Note**
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

Note See the "Semiconductor Device Mount Manual" website (http://www.necel.com/pkg/en/mount/index.html).

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

CONTENTS

CHAPTER 1 OVERVIEW

1.1 Features

O Minimum instruction execution time selectable from high speed (0.2 μs) and low speed (3.2 μs) (with CPU clock of 10 MHz)

O General-purpose registers: 8 bits × 8 registers

O ROM and RAM capacities

Item / Part number	Program Memory (Flash Memory)	Memory (Internal High-Speed RAM)
μPD78F9221	2 KB	128 bytes
μPD78F9222	4 KB	256 bytes

O On-chip power-on clear (POC) circuit and low voltage detector (LVI)

O On-chip watchdog timer (operable on internal low-speed Ring-OSC clock)

O I/O ports: 17

O Timer: 4 channels

- 16-bit timer/event counter: 1 channel
- 8-bit timer: 2 channels
- Watchdog timer: 1 channel

O Serial interface: UART (LIN (Local Interconnect Network) bus supported) 1 channel

O 10-bit resolution A/D converter: 4 channels

O Supply voltage: V_{DD} = 2.0 to 5.5 V[Note]

O Operating temperature range: T_A = −40 to +85°C

Note Use this product in a voltage range of 2.2 to 5.5 V because the detection voltage (V_{POC}) of the power-on clear (POC) circuit is 2.1 V ±0.1 V.

1.2 Application Fields

O Automotive electronics

- System control of body instrumentation system (such as power windows and keyless entry reception)
- Sub-microcontroller of control system

O Household appliances

- Electric toothbrushes
- Electric shavers

O Toys

O Industrial equipment

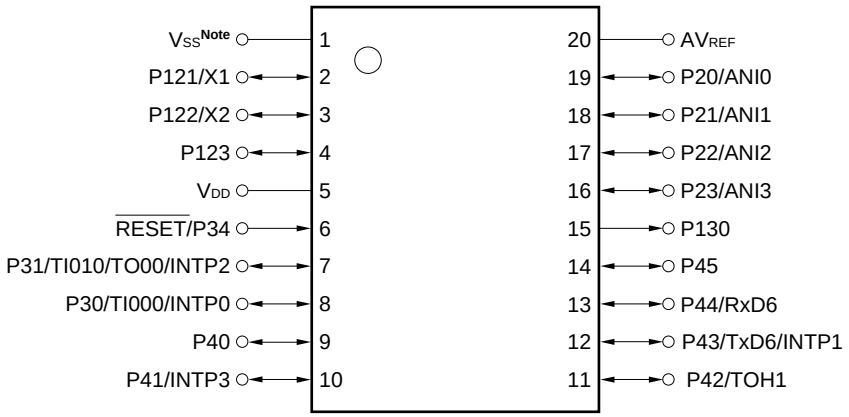
- Sensor and switch control
- Power tools

1.3 Ordering Information

Part Number	Package	Internal ROM
μPD78F9221MC-5A4	20-pin plastic SSOP (7.62 mm (300))	Flash memory
μPD78F9222MC-5A4	20-pin plastic SSOP (7.62 mm (300))	Flash memory

1.4 Pin Configuration (Top View)

20-pin plastic SSOP (7.62 mm (300))
μPD78F9221MC-5A4
μPD78F9222MC-5A4



Note VSS and AVSS are internally connected in the 78K0S/KA1+. Be sure to connect VSS to a stabilized GND in order to stabilize VSS via GND (= 0 V).

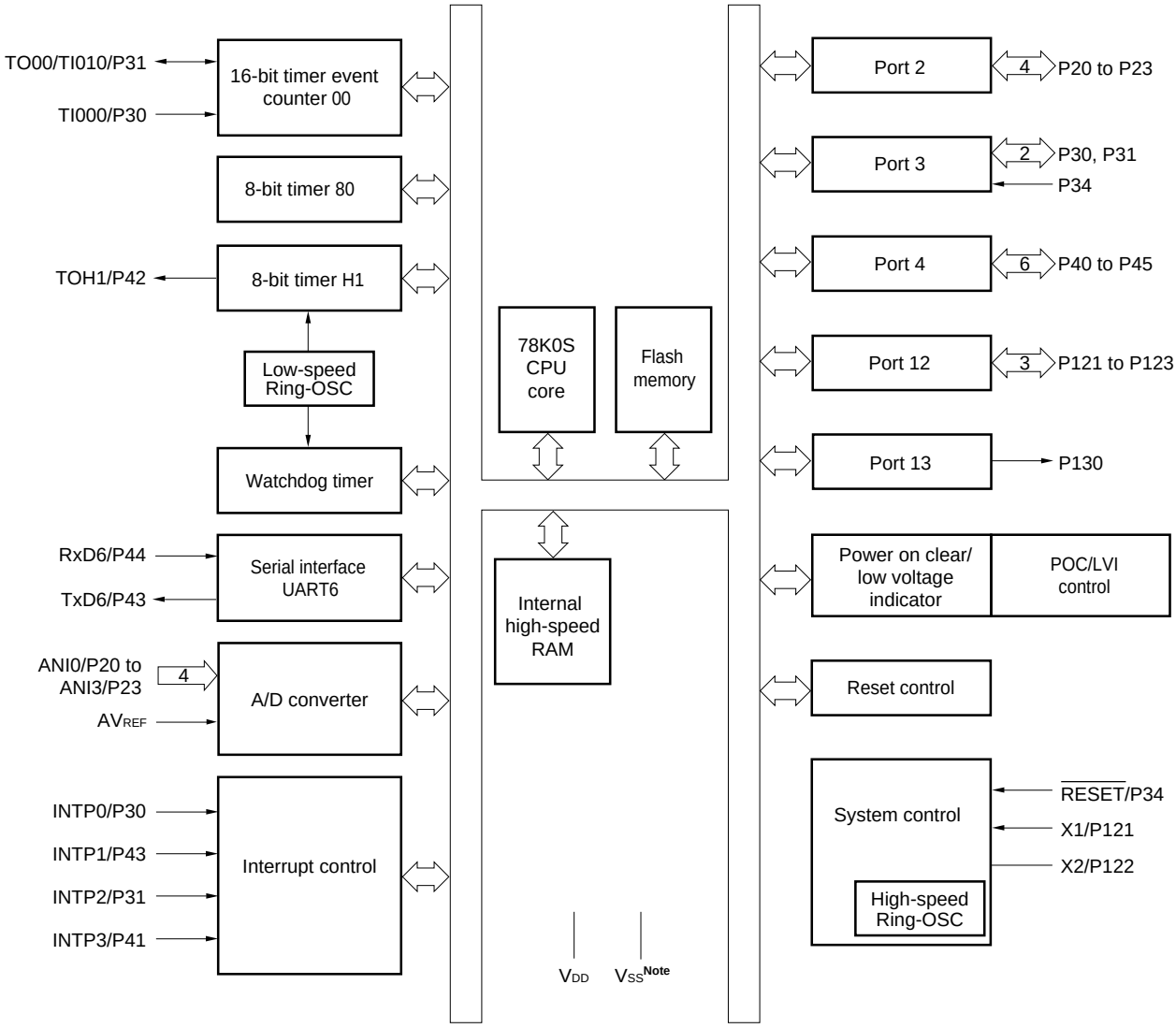
ANI0 to ANI3:	Analog input	RESET:	Reset
AVREF:	Analog reference voltage	RxD6:	Receive data
RxD6:	Receive data	TI000, TI010:	Timer input
INTP0 to INTP3:	External interrupt input	TO00, TOH1:	Timer output
P20 to P23:	Port 2	TxD6:	Transmit data
P30, P31, P34:	Port 3	VDD:	Power supply
P40 to P45:	Port 4	VSS:	Ground
P121 to P123:	Port 12	X1, X2:	Crystal oscillator (X1 input clock)
P130:	Port 13		

1.5 78K0S/Kx1+ Product Lineup

The following table shows the product lineup of the 78K0S/Kx1+.

Item \ Part Number		78K0S/KU1+	78K0S/KY1+	78K0S/KA1+		78K0S/KB1+
Number of pins		8 pins	16 pins	20 pins		30 pins
Internal memory	Flash memory	1 KB, 2 KB, 4 KB	1 KB, 2 KB, 4 KB	2 KB	4 KB	4 KB, 8 KB
	RAM	128 bytes	128 bytes	128 bytes	256 bytes	256 bytes
Supply voltage		V_{DD} = 2.0 to 5.5 V				
Minimum instruction execution time		0.20 μs (10 MHz, V_{DD} = 4.0 to 5.5 V) 0.33 μs (6 MHz, V_{DD} = 3.0 to 5.5 V) 0.40 μs (5 MHz, V_{DD} = 2.7 to 5.5 V) 4.0 μs (500 kHz, V_{DD} = 2.0 to 5.5 V)				
System clock (oscillation frequency)		Internal high-speed Ring-OSC oscillation (8 MHz (TYP.)) Crystal/ceramic oscillation (1 to 10 MHz) X1 external clock input oscillation (1 to 10 MHz)				
Clock for TMH1 and WDT (oscillation frequency)		Internal low-speed Ring-OSC oscillation (240 kHz (TYP.))				
Port	CMOS I/O	5	13	15		24
	CMOS input	1	1	1		1
	CMOS output	–	–	1		1
Timer	16-bit (TM0)	1 ch				
	8-bit (TMH)	1 ch				
	8-bit (TM8)	–		1 ch		
	WDT	1 ch				
Serial interface		–		LIN-Bus-supporting UART: 1 ch		
A/D converter		8 bits: 4 ch (2.7 to 5.5V)		10 bits: 4 ch (2.7 to 5.5V)		
Interrupts	External	2		4		
	Internal	6		10		
Reset	$\overline{\text{RESET}}$ pin	Provided				
	POC	2.1 V ±0.1 V				
	LVI	Provided (selectable by software)				
	WDT	Provided				
Operating temperature range		–40 to +85°C				

1.6 Block Diagram



Note V_{SS} and AV_{SS} are internally connected in the 78K0S/KA1+. Be sure to connect V_{SS} to stabilized GND in order to stabilize V_{SS} via GND (= 0 V).

1.7 Functional Outline

Item		μPD78F9221	μPD78F9222
Internal memory	Flash memory	2 KB	4 KB
	High-speed RAM	128 bytes	256 bytes
Memory space		64 KB	
X1 input clock (oscillation frequency)		Crystal/ceramic/external clock input: 10 MHz (V_{DD} = 4.0 to 5.5 V), 6 MHz (V_{DD} = 3.0 to 5.5 V), 5 MHz (V_{DD} = 2.7 to 5.5 V), 500 kHz (V_{DD} = 2.0 to 5.5 V)	
Ring-OSC clock	High speed (oscillation frequency)	Internal Ring oscillation: 8 MHz (TYP.)	
	Low speed (for TMH1 and WDT)	Internal Ring oscillation: 240 kHz (TYP.)	
General-purpose registers		8 bits × 8 registers	
Minimum instruction execution time		0.2 μs/0.8 μs (X1 input clock: f_X = 10 MHz)	
Instruction set		• 16-bit operation • Bit manipulation (set, reset, test), etc.	
I/O port		Total: 17 pins CMOS I/O: 15 pins CMOS input: 1 pin CMOS output: 1 pin	
Timer		• 16-bit timer/event counter: 1 channel • 8-bit timer (timer H1): 1 channel • 8-bit timer (timer 80): 1 channel • Watchdog timer: 1 channel	
	Timer output	2 pins (PWM: 1 pin)	
A/D converter		10-bit resolution × 4 channels	
Serial interface		LIN-bus-supporting UART mode: 1 channel	
Vectored interrupt sources	External	4	
	Internal	10	
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on clear • Internal reset by low-voltage detector	
Supply voltage		V_{DD} = 2.0 to 5.5 V[Note]	
Operating temperature range		TA = –40 to +85°C	
Package		20-pin plastic SSOP (7.62 mm (300))	

Note Use this product in a voltage range of 2.2 to 5.5 V because the detection voltage (V_{POC}) of the power-on-clear (POC) circuit is 2.1 V ±0.1 V.

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

(1) Port pins

Pin Name	I/O	Function		After Reset	Alternate-Function Pin
P20 to P23	I/O	Port 2. 4-bit I/O port. Can be set to input or output mode in 1-bit units. An on-chip pull-up resistor can be connected by setting software.		Input	ANI0 to ANI3
P30	I/O	Port 3	Can be set to input or output mode in 1-bit units. An on-chip pull-up resistor can be connected by setting software.	Input	TI000/INTP0
P31					TI010/TO00/INTP2
P34	Input		Input only	Input	$\overline{\text{RESET}}$
P40	I/O	Port 4. 6-bit I/O port. Can be set to input or output mode in 1-bit units. An on-chip pull-up resistor can be connected by setting software.		Input	–
P41					INTP3
P42					TOH1
P43					TxD6/INTP1
P44					RxD6
P45					–
P121	I/O	Port 12. 3-bit I/O port. Can be set to input or output mode in 1-bit units. An on-chip pull-up resistor can be connected only to P123 by setting software.		Input	X1
P122					X2
P123					–
P130	Output	Port 13. 1-bit output-only port		Output	–

(2) Non-port pins

Pin Name	I/O	Function	After Reset	Alternate-Function Pin
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/TI000
INTP1				P43/TxD6
INTP2				P31/TI010/TO00
INTP3				P41
RxD6	Input	Serial data input for asynchronous serial interface	Input	P44
TxD6	Output	Serial data output for asynchronous serial interface	Input	P43/INTP1
TI000	Input	External count clock input to 16-bit timer/event counter 00. Capture trigger input to capture registers (CR000 and CR010) of 16-bit timer/event counter 00	Input	P30/INTP0
TI010		Capture trigger input to capture register (CR000) of 16-bit timer/event counter 00		P31/TO00/INTP2
TO00	Output	16-bit timer/event counter 00 output	Input	P31/TI010/INTP2
TOH1	Output	8-bit timer H1 output	Input	P42
ANI0 to ANI3	Input	Analog input of A/D converter	Input	P20 to P23
AV_{REF}	–	Reference voltage of A/D converter	–	–
$\overline{RESET}$	Input	System reset input	–	–
X1	Input	Connection of crystal/ceramic oscillator for system clock oscillation. External clock input	–	P121
X2	–	Connection of crystal/ceramic oscillator for system clock oscillation.	–	P122
V_{DD}	–	Positive power supply	–	–
V_{SS}	–	Ground potential	–	–

2.2 Pin Functions

2.2.1 P20 to P23 (Port 2)

P20 to P23 constitute a 4-bit I/O port, port 2. In addition to I/O port pins, these pins also have a function to input analog signals to the A/D converter. These pins can be set to the following operation modes in 1-bit units.

(1) Port mode

P20 to P23 function as a 4-bit I/O port. Each bit of this port can be set to the input or output mode by using port mode register 2 (PM2). In addition, an on-chip pull-up resistor can be connected to the port by using pull-up resistor option register 2 (PU2).

(2) Control mode

P20 to P23 function as the analog input pins (ANI0 to ANI3) of the A/D converter. When using these pins as analog input pins, refer to **10.6 Cautions for A/D converter (5) ANI0/P20 to ANI3/P23**.

2.2.2 P30, P31, and P34 (Port 3)

P30, P31 and P34 constitute a 2-bit I/O port, port 3. In addition to I/O port pins, these pins also have functions to input/output a timer signal, and input an external interrupt request signal.

P34 is a 1-bit input-only port. This pin is also used as a $\overline{\text{RESET}}$ pin.

P30 and P31 can be set to the following operation modes in 1-bit units.

(1) Port mode

P30 and P31 function as a 2-bit I/O port. Each bit of this port can be set to the input or output mode by using port mode register 3 (PM3). In addition, an on-chip pull-up resistor can be connected to the port by using pull-up resistor option register 3 (PU3).

P34 functions as a 1-bit input-only port.

(2) Control mode

P30, P31, and P34 function to input/output signals to/from internal timers, and to input an external interrupt request signal.

(a) INTP0 and INTP2

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

(b) TI000

This pin inputs an external count clock to 16-bit timer/event counter 00, or a capture trigger signal to the capture registers (CR000 and CR010) of 16-bit timer/event counter 00.

(c) TI010

This pin inputs a capture trigger signal to the capture register (CR000) of 16-bit timer/event counter 00.

(d) TO00

This pin outputs a signal from 16-bit timer/event counter 00.

2.2.3 P40 to P45 (Port 4)

P40 to P45 constitute a 6-bit I/O port, port 4. In addition to I/O port pins, these pins also have functions to output a timer signal, input external interrupt request signals, and input/output the data of the serial interface.

These pins can be set to the following operation modes in 1-bit units.

(1) Port mode

P40 and P45 function as a 6-bit I/O port. Each bit of this port can be set to the input or output mode by using port mode register 4 (PM4). In addition, an on-chip pull-up resistor can be connected to the port by using pull-up resistor option register 4 (PU4).

(2) Control mode

P40 and 45 function to output a signal from an internal timer, input external interrupt request signals, and input/output data of the serial interface.

(a) INTP1 and INTP3

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

(b) TOH1

This is the output pin of 8-bit timer H1.

(c) TxD6

This pin outputs serial data from the asynchronous serial interface.

(d) RxD6

This pin inputs serial data to the asynchronous serial interface.

2.2.4 P121 to P123 (Port 12)

P121 to P123 constitute a 3-bit I/O port, port 12.

Each bit of this port can be set to the input or output mode by using port mode register 12 (PM12). An on-chip pull-up resistor can be connected to P123 by using pull-up resistor option register 12 (PU12).

P121 and P122 also function as the X1 and X2 pins, respectively.

2.2.5 P130 (Port 13)

This is a 1-bit output-only port.

2.2.6 $\overline{\text{RESET}}$

This pin inputs an active-low system reset signal.

2.2.7 X1 and X2

These pins connect an oscillator to oscillate the X1 input clock.

Supply an external clock to X1.

2.2.8 AV_{REF}

This pin inputs a reference voltage to the internal A/D converter. When the A/D converter is not used, connect this pin to V_{DD}.

2.2.9 V_{DD}

This is the positive power supply pin.

2.2.10 V_{SS}

This is the ground pin.

2.3 Pin I/O Circuits and Connection of Unused Pins

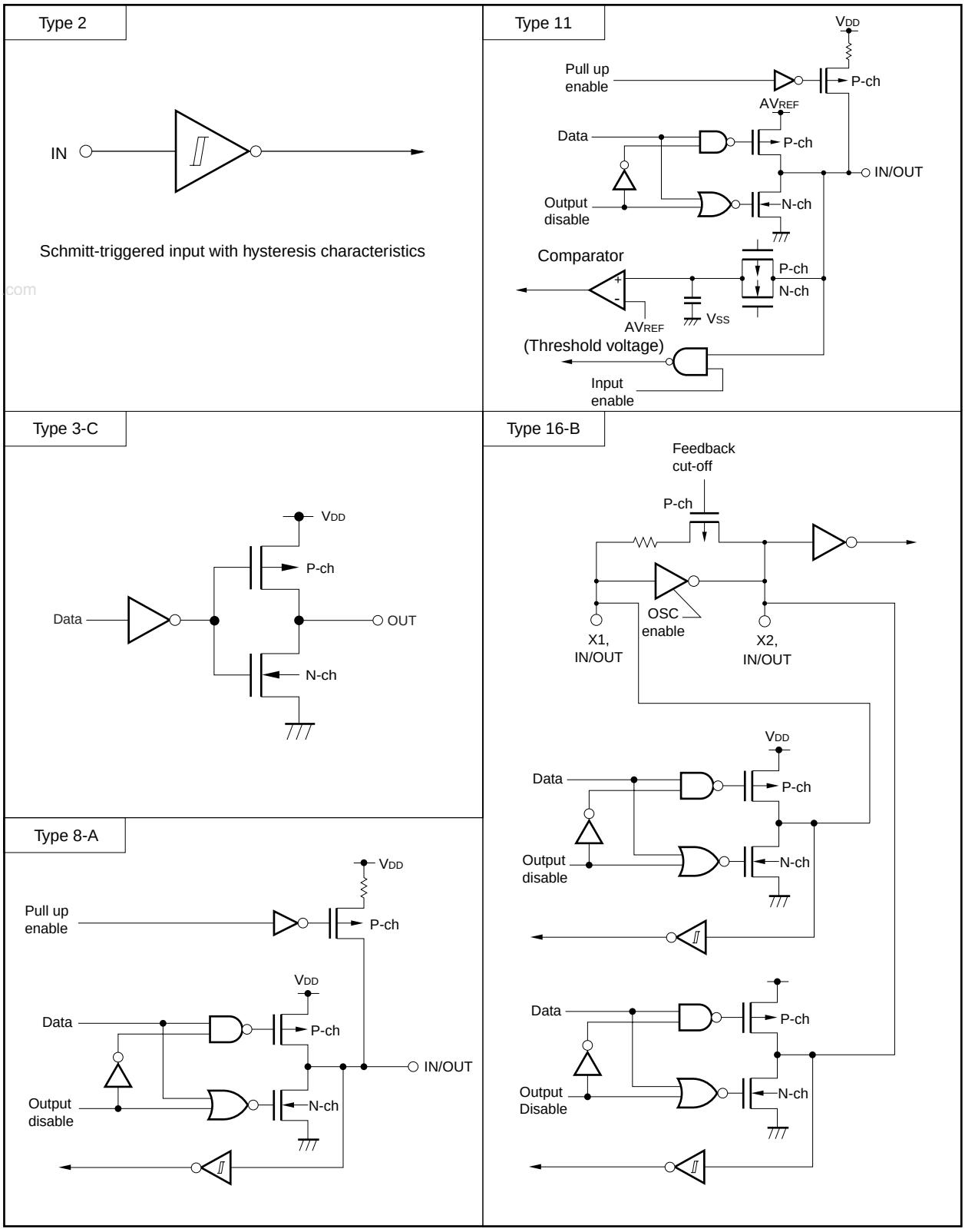
Table 2-1 shows I/O circuit type of each pin and the connections of unused pins.

For the configuration of the I/O circuit of each type, refer to **Figure 2-1**.

Table 2-1. Types of Pin I/O Circuits and Connection of Unused Pins

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pin
P20/ANI0 to P23/ANI3	11	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P30/TI000/INTP0	8-A		
P31/TI010/TO00/INTP2			
P34/$\overline{\text{RESET}}$	2	Input	Directly connect to V_{DD} or V_{SS}.
P40	8-A	I/O	Input: Individually connect to V_{DD} or V_{SS} via resistor. Output: Leave open.
P41/INTP3			
P42/TOH1			
P43/TxD6/INTP1			
P44/RxD6			
P45			
P121/X1	16-B		
P122/X2			
P123	8-A		
P130	3-C	Output	Leave open.
AV_{REF}	–	Input	Directly connect to V_{DD}.

Figure 2-1. Pin I/O Circuits

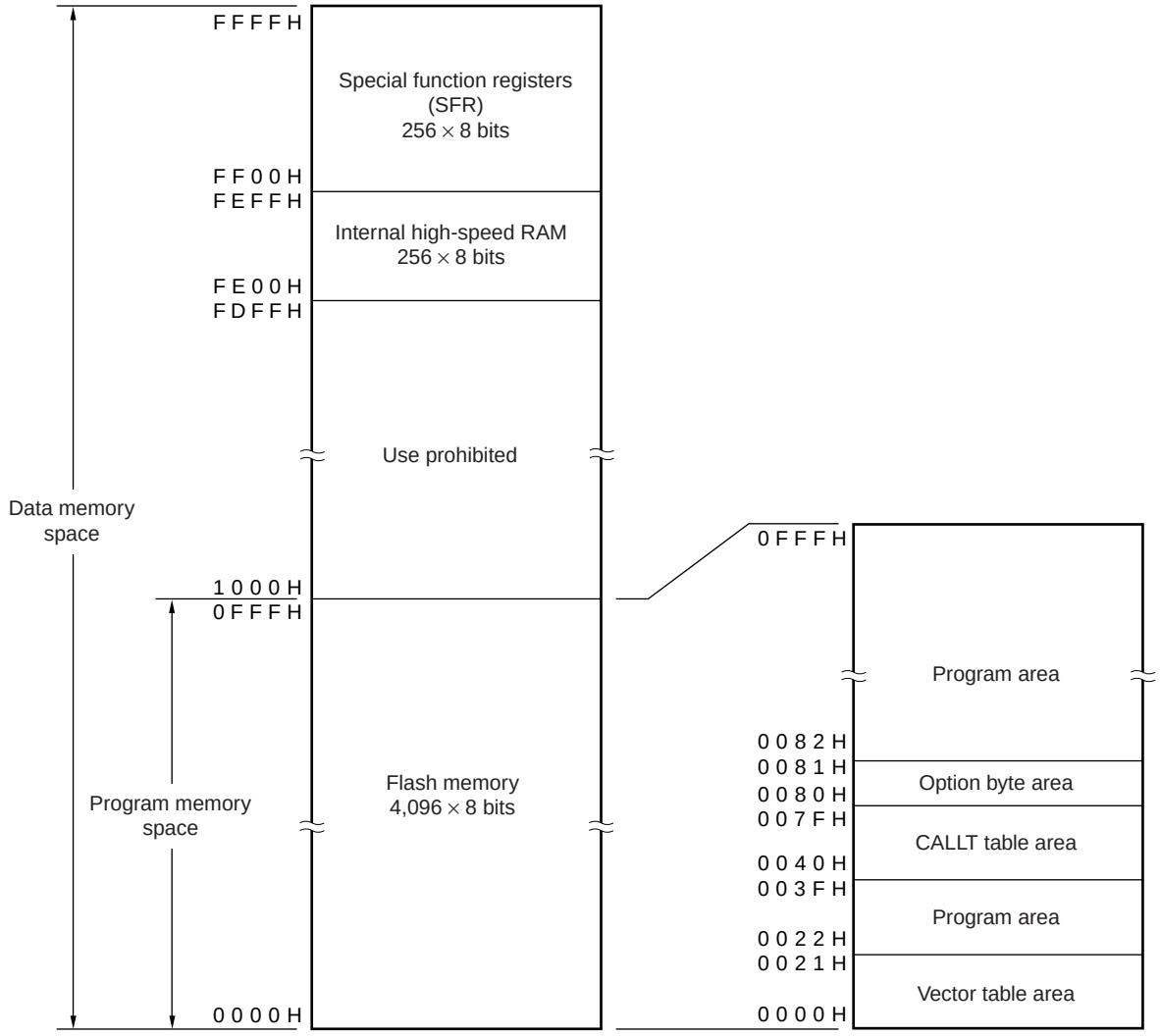


CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

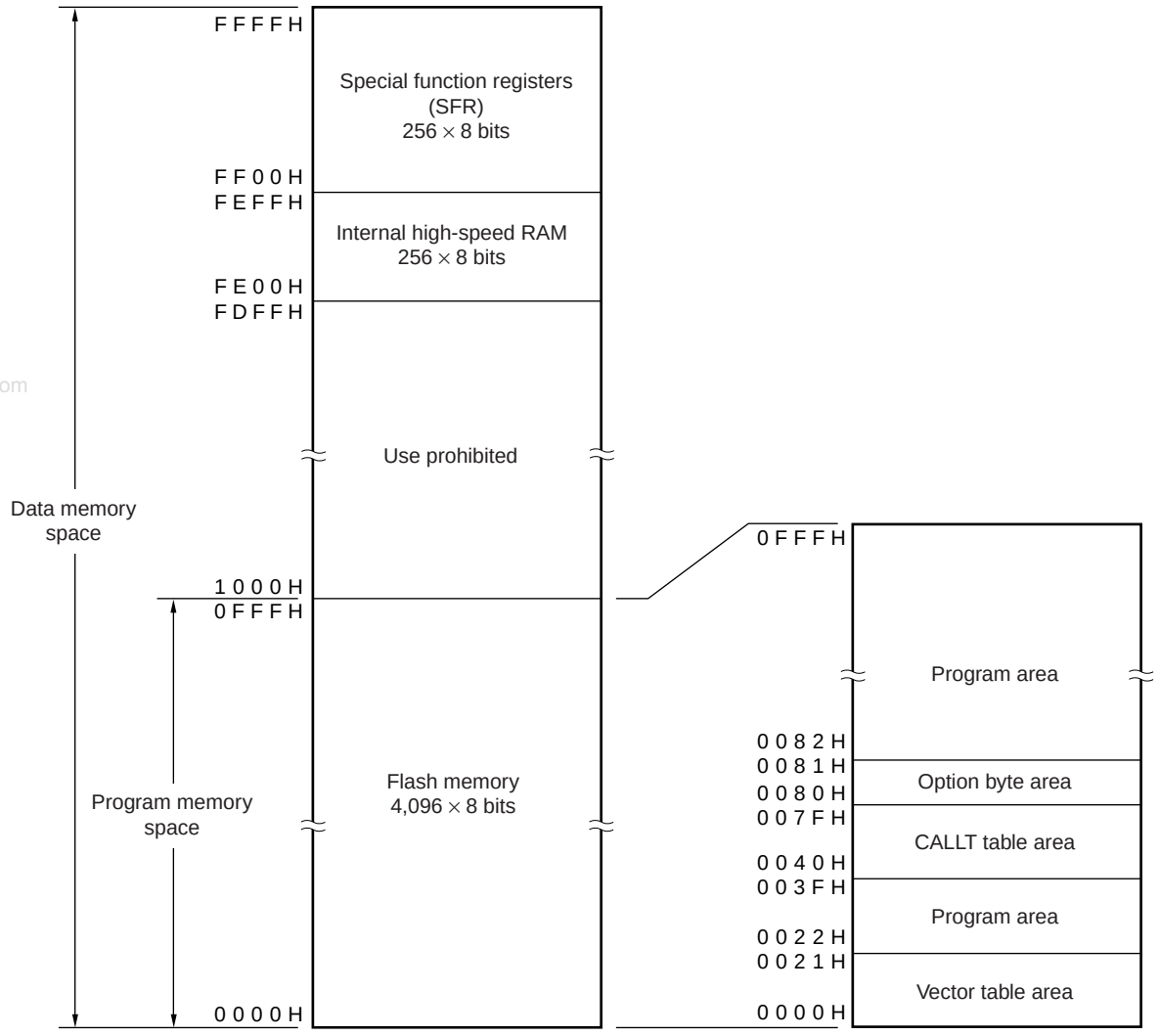
The 78K0S/KA1+ can access up to 64 KB of memory space. Figures 3-1 and 3-2 show the memory maps.

Figure 3-1. Memory Map (μPD78F9221)



Remark The option byte is one byte at 0080H.

Figure 3-2. Memory Map (μPD78F9222)



Remark The option byte is one byte at 0080H.

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 78K0S/KA1+ provides the following internal ROMs (or flash memory) containing the following capacities.

Table 3-1. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD78F9221	Flash memory	2,048 × 8 bits
μPD78F9222		4,096 × 8 bits

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

(1) Vector table area

The 34-byte area of addresses 0000H to 0021H is reserved as a vector table area. This area stores program start addresses to be used when branching by $\overline{\text{RESET}}$ input or interrupt request generation. Of a 16-bit 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	Reset input	0014H	INTFLC
0006H	INTLVI	0016H	INTP2
0008H	INTP0	0018H	INTP3
000AH	INP1	001AH	INTTM80
000CH	INTTMH1	001CH	INTSRE6
000EH	INTTM000	001EH	INTSR6
0010H	INTTM010	0020H	INTST6
0012H	INTAD		

(2) CALLT instruction table area

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

(3) Option byte area

The option byte area is the 1-byte area of address 0080H. For details, refer to **CHAPTER 17 OPTION BYTE**.

3.1.2 Internal data memory space

128-byte internal high-speed RAM is provided in the μPD78F9221 and 256-byte in the μPD78F9222.

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 the area of FF00H to FFFFH (see **Table 3-3**).

3.1.4 Data memory addressing

The 78K0S/KA1+ is provided with a wide range of addressing modes to make memory manipulation as efficient as possible. The data memory area (FE80H to FFFFH or FE00H to FFFFH) can be accessed using a unique addressing mode according to its use, such as a special function register (SFR). Figures 3-3 and 3-4 illustrate the data memory addressing.

Figure 3-3. Data Memory Addressing (μPD78F9221)

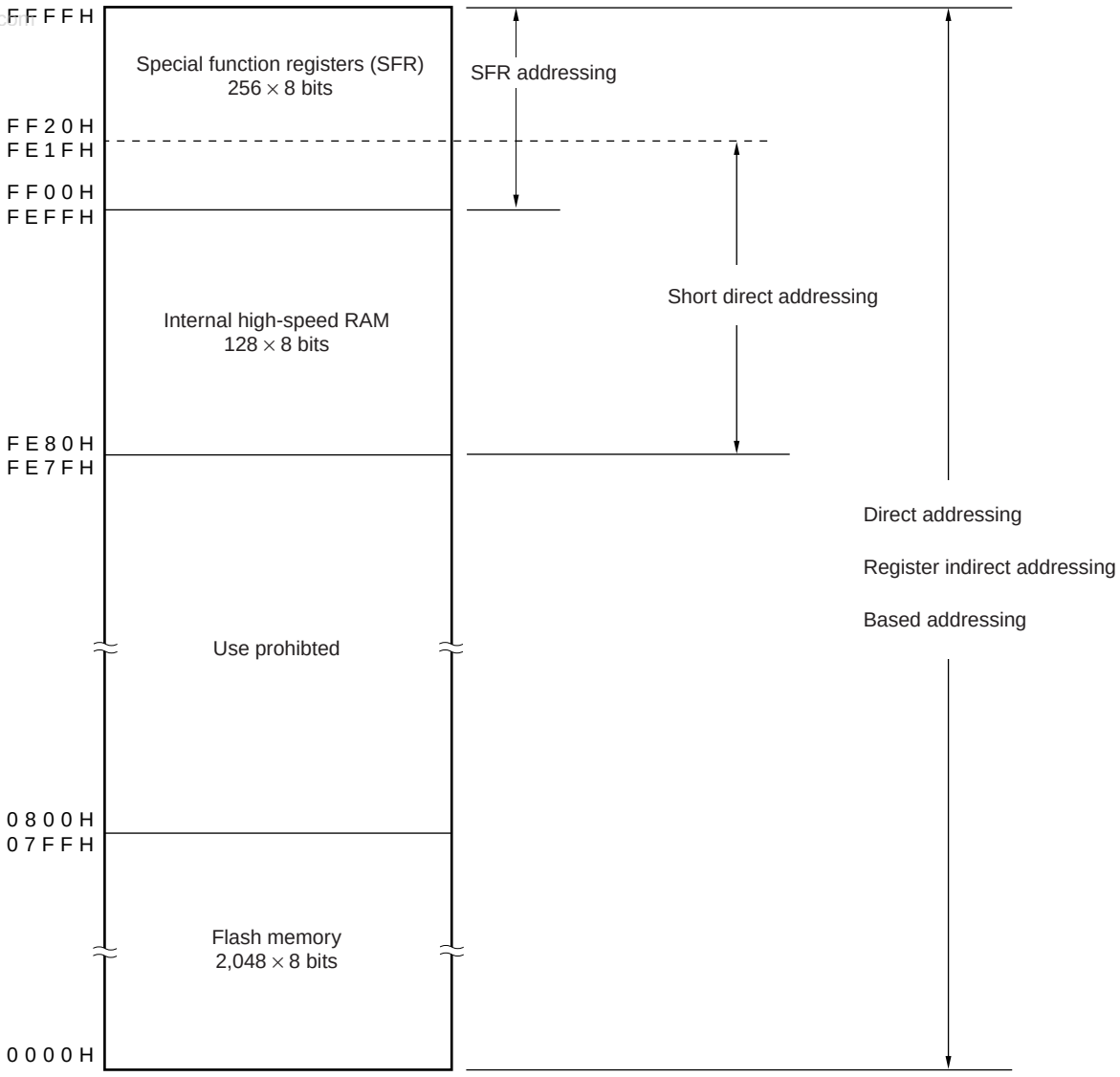
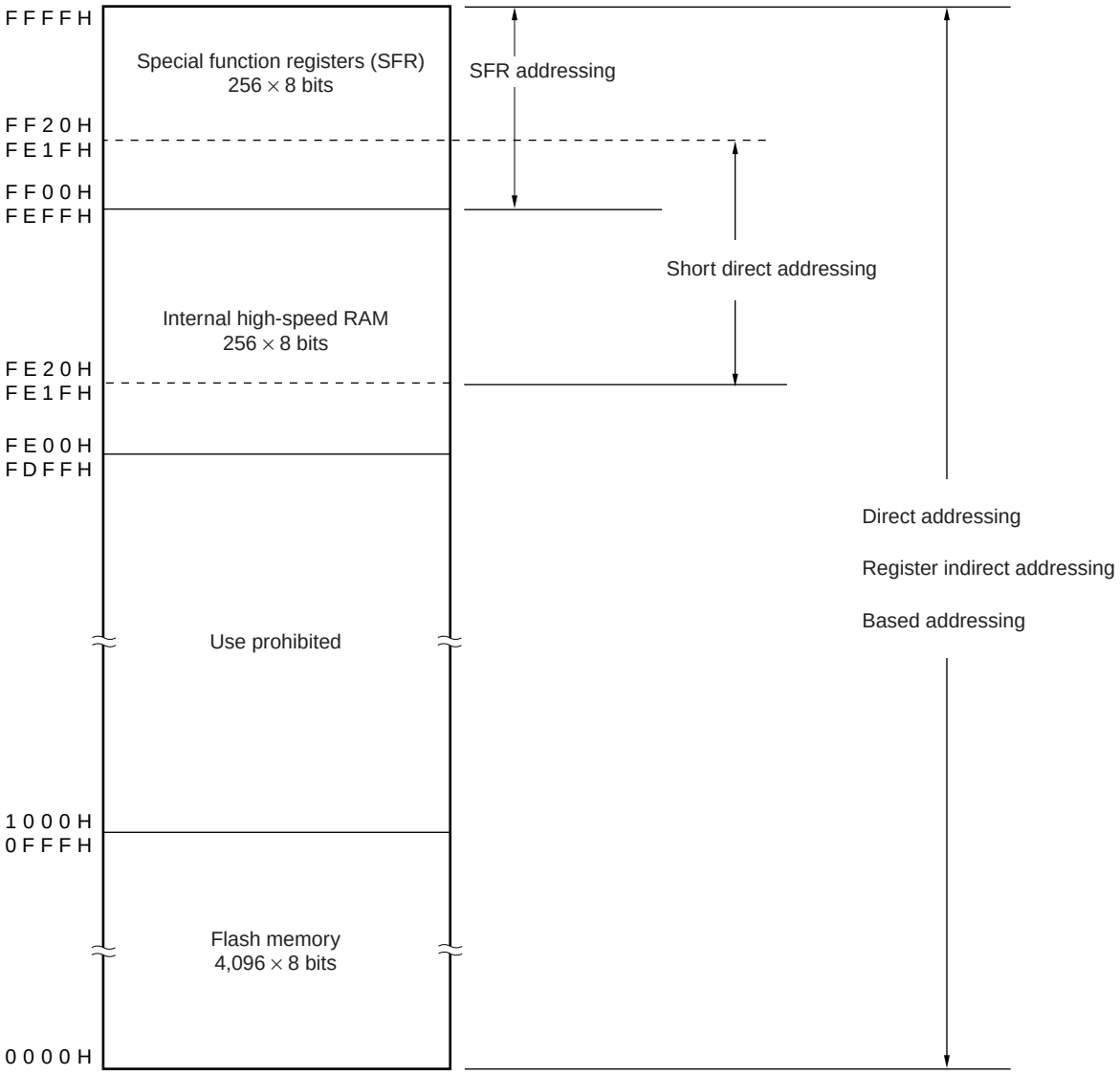


Figure 3-4. Data Memory Addressing (μPD78F9222)



3.2 Processor Registers

The 78K0S/KA1+ provides 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.

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

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

Figure 3-5. Program Counter Configuration

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

(2) Program status word (PSW)

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

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

Figure 3-6. Program Status Word Configuration

	7							0
PSW	IE	Z	0	AC	0	0	1	CY

(a) Interrupt enable flag (IE)

This flag controls interrupt request acknowledge operations of the 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 acknowledgment 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 3-7. Stack Pointer Configuration

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

The SP is decremented before 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 3-8 and 3-9.

Caution Since reset input makes SP contents undefined, be sure to initialize the SP before instruction execution.

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

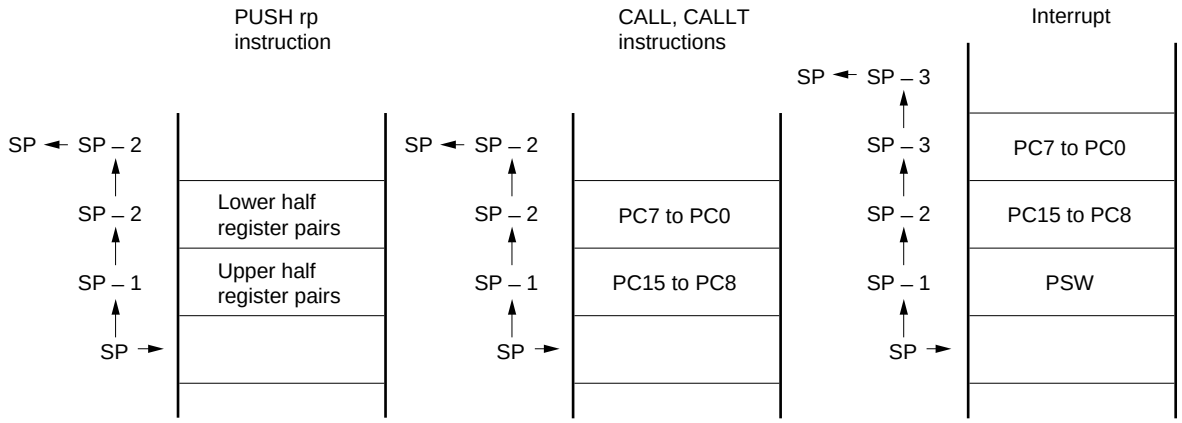
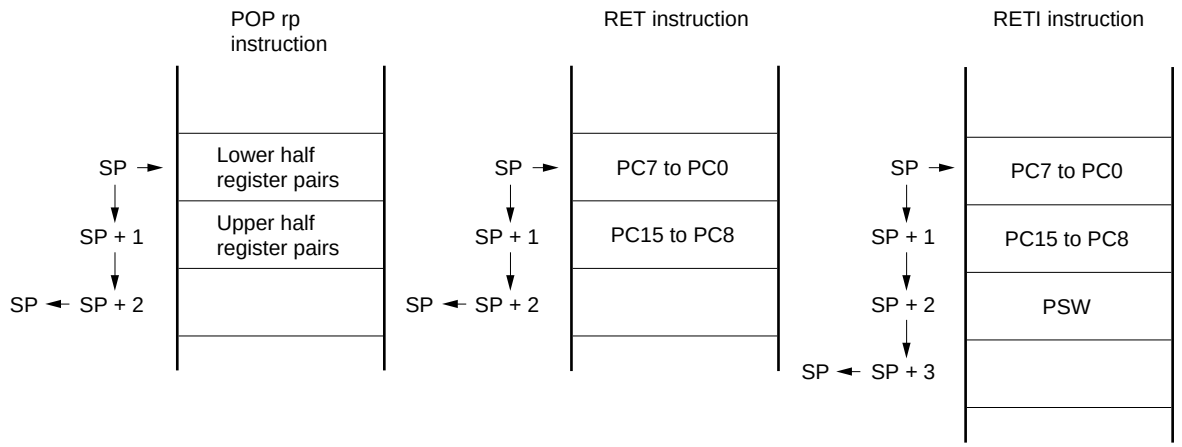


Figure 3-9. Data to Be Restored from Stack Memory



3.2.2 General-purpose registers

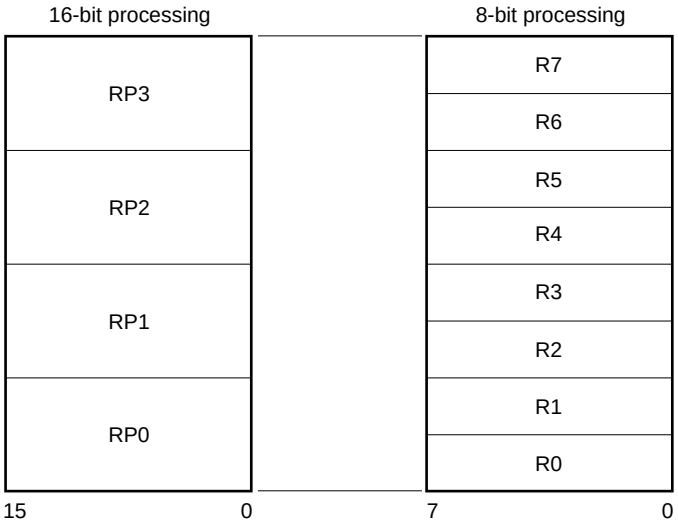
A general-purpose register consists of eight 8-bit registers (X, A, C, B, E, D, L, and H).

In addition each register being 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).

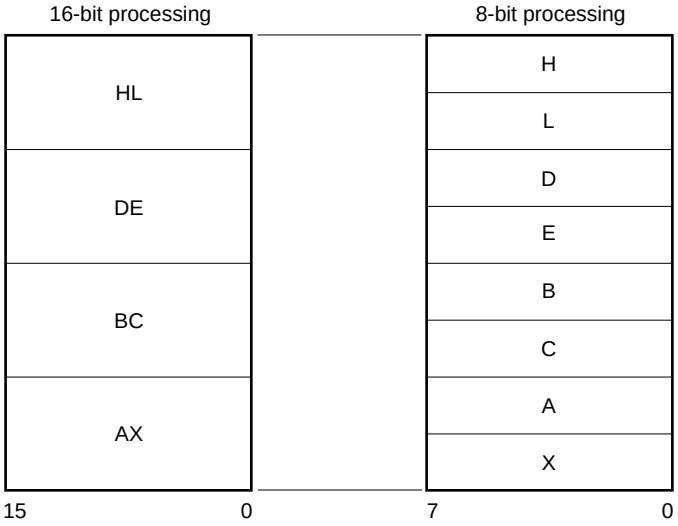
Registers can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute names (R0 to R7 and RP0 to RP3).

Figure 3-10. General-Purpose Register Configuration

(a) Absolute names



(b) Function names



3.2.3 Special function registers (SFRs)

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

The special function registers are allocated to the 256-byte area FF00H to FFFFH.

The special function registers can be manipulated, like the general-purpose registers, with 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 by the 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 by the 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 by the assembler for the 16-bit manipulation instruction operand. When specifying an address, describe an even address.

Table 3-3 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 reserved words in the assembler, and have already been defined in a header file called “sfrbit.h” in the C compiler. Therefore, these symbols can be used as instruction operands if an 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

- Number of bits manipulated simultaneously
  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 a reset is input.

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Number of Bits Manipulated Simultaneously			After Reset
				1 Bit	8 Bits	16 Bits	
FF02H	Port register 2	P2	R/W Note 1	√	√	–	00H
FF03H	Port register 3	P3		√	√	–	
FF04H	Port register 4	P4		√	√	–	
FF0CH	Port register 12	P12		√	√	–	
FF0DH	Port register 13	P13	W	√	√	–	
FF0EH	8-bit timer H compare register 01	CMP01	R/W	–	√	–	
FF0FH	8-bit timer H compare register 11	CMP11		–	√	–	
FF12H	16-bit timer counter 00	TM00	R	–	–	√ Note 2	0000H
FF13H							
FF14H	16-bit timer capture/compare register 000	CR000	R/W	–	–	√ Note 2	0000H
FF15H							
FF16H	16-bit timer capture/compare register 010	CR010		–	–	√ Note 2	0000H
FF17H							
FF18H	10-bit A/D conversion result register	ADCR	R	–	–	√ Note 2	Undefined
FF19H							
FF1AH	8-bit A/D conversion result register	ADCRH		–	√	–	
FF22H	Port mode register 2	PM2	R/W	√	√	–	FFH
FF23H	Port mode register 3	PM3		√	√	–	
FF24H	Port mode register 4	PM4		√	√	–	
FF2CH	Port mode register 12	PM12		√	√	–	
FF32H	Pull-up resistance option register 2	PU2		√	√	–	00H
FF33H	Pull-up resistance option register 3	PU3		√	√	–	
FF34H	Pull-up resistance option register 4	PU4		√	√	–	
FF3CH	Pull-up resistance option register 12	PU12		√	√	–	
FF48H	Watchdog timer mode register	WDTM		–	√	–	67H
FF49H	Watchdog timer enable register	WDTE		–	√	–	9AH
FF50H	Low voltage detect register	LVIM		–	√	–	00H
FF51H	Low voltage detection level select register	LVIS		–	√	–	
FF54H	Reset control flag register	RESF	R	–	√	–	00H Note 3
FF58H	Low-speed Ring-OSC mode register	LSRCM	R/W	–	√	–	00H
FF5AH	High-speed Ring-OSC mode register	HSRCM		–	√	–	
FF60H	16-bit timer mode control register 00	TMC00		√	√	–	
FF61H	Prescaler mode register 00	PRM00		√	√	–	
FF62H	Capture/compare control register 00	CRC00		√	√	–	
FF63H	16-bit timer output control register 00	TOC00		√	√	–	
FF70H	8-bit timer H mode register 1	TMHMD1		√	√	–	

Notes 1. Only P34 is an input-only port.

2. A 16-bit access is possible only by the short direction addressing.

3. Varies depending on the reset cause.

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Number of Bits Manipulated Simultaneously			After Reset
				1 Bit	8 Bits	16 Bits	
FF80H	A/D converter mode register	ADM	R/W	√	√	–	00H
FF81H	Analog input channel specify register	ADS		√	√	–	
FF84H	Port mode control register 2	PMC2		√	√	–	
FF8CH	Input switching control register	ISC		√	√	–	
FF90H	Asynchronous serial interface operation mode register 6	ASIM6		√	√	–	01H
FF92H	Reception buffer register 6	RXB6	R	–	√	–	FFH
FF93H	Asynchronous serial interface reception error status register 6	ASIS6		–	√	–	00H
FF94H	Transmission buffer register 6	TXB6	R/W	–	√	–	FFH
FF95H	Asynchronous serial interface transmission status register 6	ASIF6	R	–	√	–	00H
FF96H	Clock selection register 6	CKSR6	R/W	–	√	–	
FF97H	Baud rate generator control register 6	BRGC6		–	√	–	FFH
FF98H	Asynchronous serial interface control register 6	ASICL6		√	√	–	16H
FFCCH	8-bit timer mode control register 80	TMC80		√	√	–	00H
FFCDH	8-bit compare register 80	CR80	W	–	√	–	Undefined
FFCEH	8-bit timer counter 80	TM80	R	–	√	–	00H
FFE0H	Interrupt request flag register 0	IF0	R/W	√	√	–	
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
FFFDH	External interrupt mode register 1	INTM1		–	√	–	
FFF3H	Preprocessor clock control register	PPCC		√	√	–	02H
FFF4H	Oscillation stabilization time selection register	OSTS		–	√	–	Undefined Note
FFFBH	Processor clock control register	PCC		√	√	–	02H

Note The oscillation stabilization time that elapses after release of reset is selected by the option byte. For details, refer to **CHAPTER 17 OPTION BYTE**.

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 address information is set to the PC to branch by the following addressing (for details of each instruction, refer to **78K/0S Series Instructions User's Manual (U11047E)**).

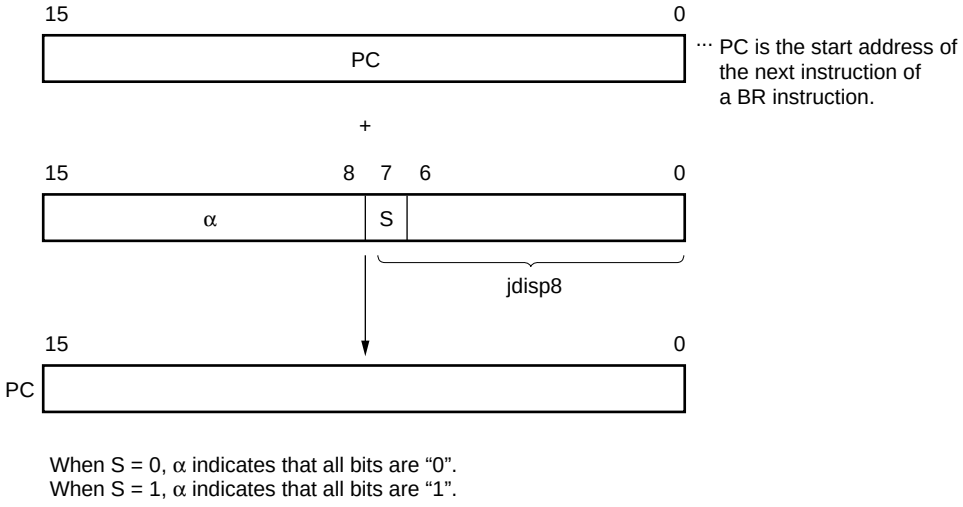
3.3.1 Relative addressing

[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 the 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 that all bits are "0".
When S = 1, α indicates that all bits are "1".

3.3.2 Immediate addressing

[Function]

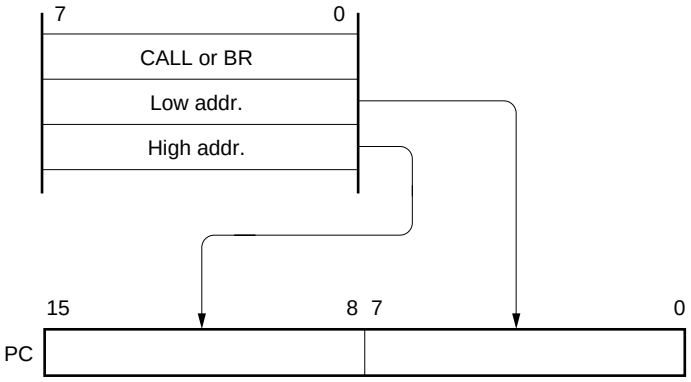
Immediate data in the instruction word is transferred to the program counter (PC) to branch.

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



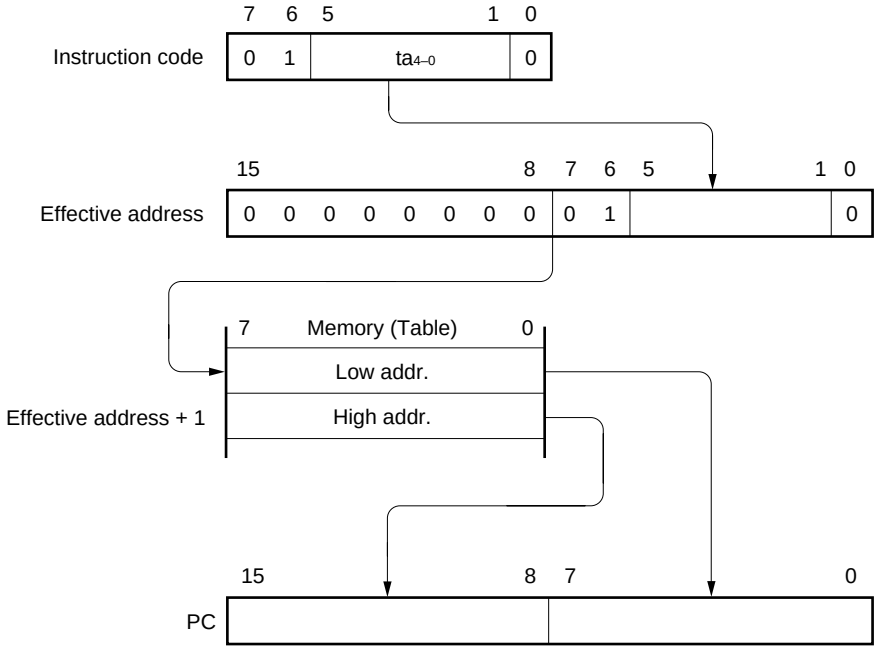
3.3.3 Table indirect addressing

[Function]

The 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]



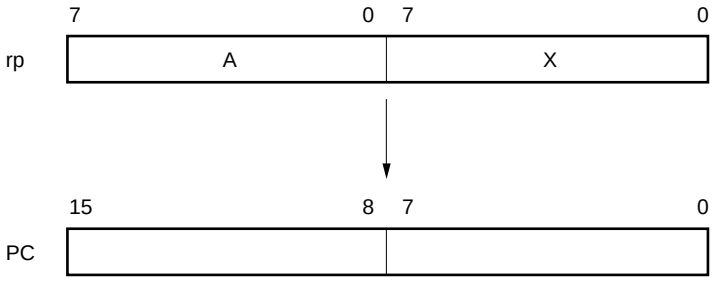
3.3.4 Register addressing

[Function]

The 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 (addressing) are available to specify the register and memory to 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]

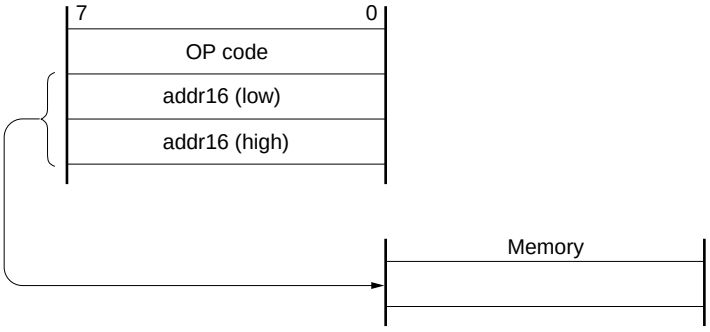
Identifier	Description
addr16	Label or 16-bit immediate data

[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 the 8-bit data in an instruction word.

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

The SFR area where short direct addressing is applied (FF00H to FF1FH) is a part of the total SFR area. 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 cleared to 0. When it is at 00H to 1FH, bit 8 is set to 1. See **[Illustration]** below.

[Operand format]

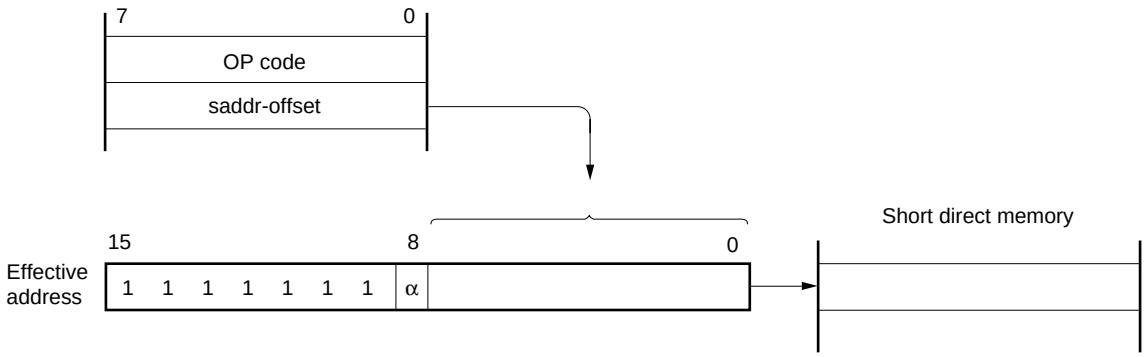
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

[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]

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

This addressing is applied to the 256-byte space FF00H to FFFFH. However, SFRs mapped at FF00H to FF1FH are accessed with short direct addressing.

[Operand format]

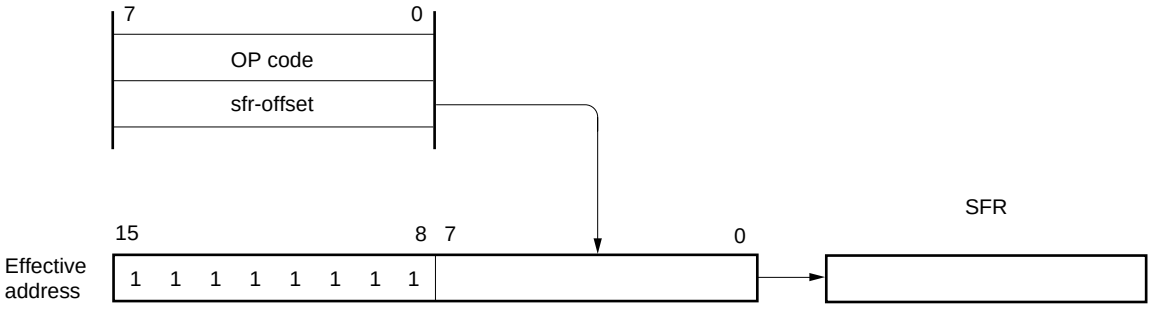
Identifier	Description
sfr	Special function register name

[Description example]

MOV PM0, A; When selecting PM0 for sfr

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

[Illustration]



3.4.4 Register addressing

[Function]

A general-purpose register is accessed as an operand.

The general-purpose register to be accessed is specified with the 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.

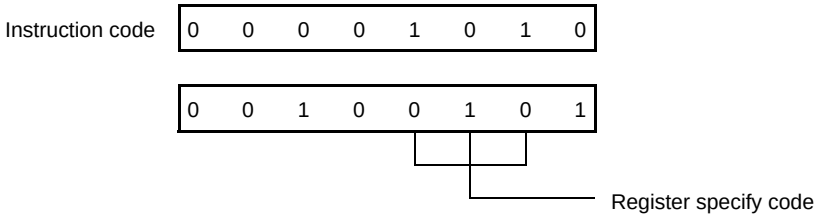
[Operand format]

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

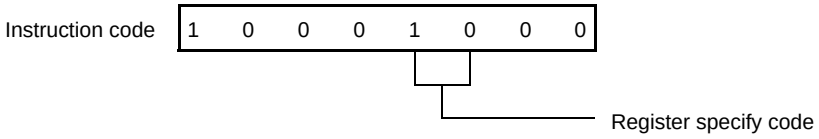
'r' and 'rp' can be described with absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[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]

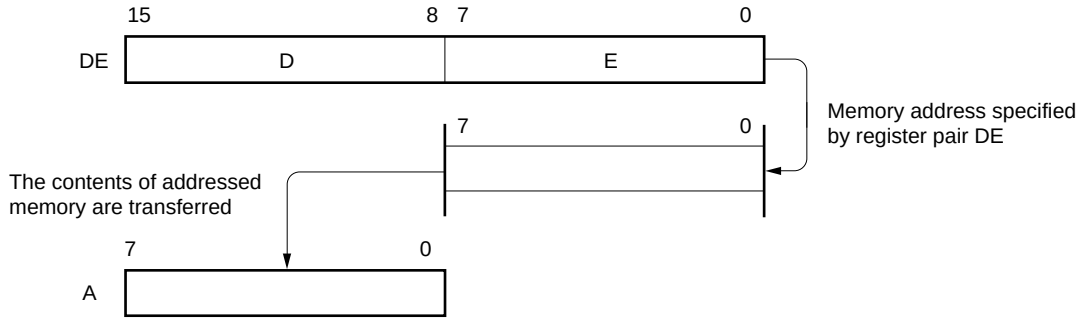
Identifier	Description
–	[DE], [HL]

[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/restored upon interrupt request generation.

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

CHAPTER 4 PORT FUNCTIONS

4.1 Functions of Ports

The 78K0S/KA1+ has the ports shown in Figure 4-1, which can be used for various control operations. Table 4-1 shows the functions of each port.

In addition to digital I/O port functions, each of these ports has an alternate function. For details, refer to **CHAPTER 2 PIN FUNCTIONS**.

Figure 4-1. Port Functions

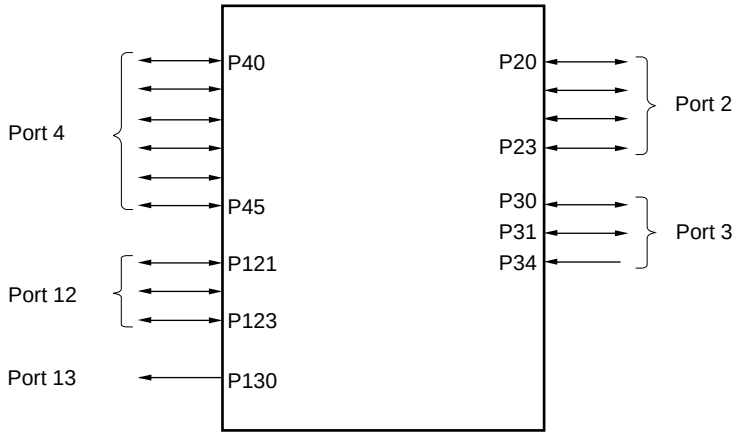


Table 4-1. Port Functions

Pin Name	I/O	Function		After Reset	Alternate-Function Pin
P20 to P23	I/O	Port 2. 4-bit I/O port. Can be set to input or output mode in 1-bit units. On-chip pull-up resistor can be connected by setting software.		Input	ANI0 to ANI3
P30	I/O	Port 3	Can be set to input or output mode in 1-bit units. On-chip pull-up resistor can be connected by setting software.	Input	TI000/INTP0
P31					TI010/TO00/INTP2
P34	Input		Input only	Input	$\overline{\text{RESET}}$
P40	I/O	Port 4. 6-bit I/O port. Can be set to input or output mode in 1-bit units. On-chip pull-up resistor can be connected setting software.		Input	–
P41					INTP3
P42					TOH1
P43					TxD6/INTP1
P44					RxD6
P45					–
P121	I/O	Port 12. 3-bit I/O port. Can be set to input or output mode in 1-bit units. On-chip pull-up resistor can be connected only to P123 by setting software.		Input	X1
P122					X2
P123					–
P130	Output	Port 13. 1-bit output-only port.		Output	–

Remarks 1. P121 and P122 can be allocated when the high-speed Ring-OSC is selected as the system clock.

2. P121 can be allocated when an external clock is selected as the system clock.

4.2 Port Configuration

Ports consist of the following hardware units.

Table 4-2. Configuration of Ports

Item	Configuration
Control registers	Port mode registers (PM2, PM3, PM4, PM12) Port mode control register 2 (PMC2) Port registers (P2, P3, P4, P12, P13) Pull-up resistor option registers (PU2, PU3, PU4, PU12)
Ports	Total: 17 (CMOS I/O: 15, CMOS input: 1, CMOS output: 1)
Pull-up resistor	Total: 13

4.2.1 Port 2

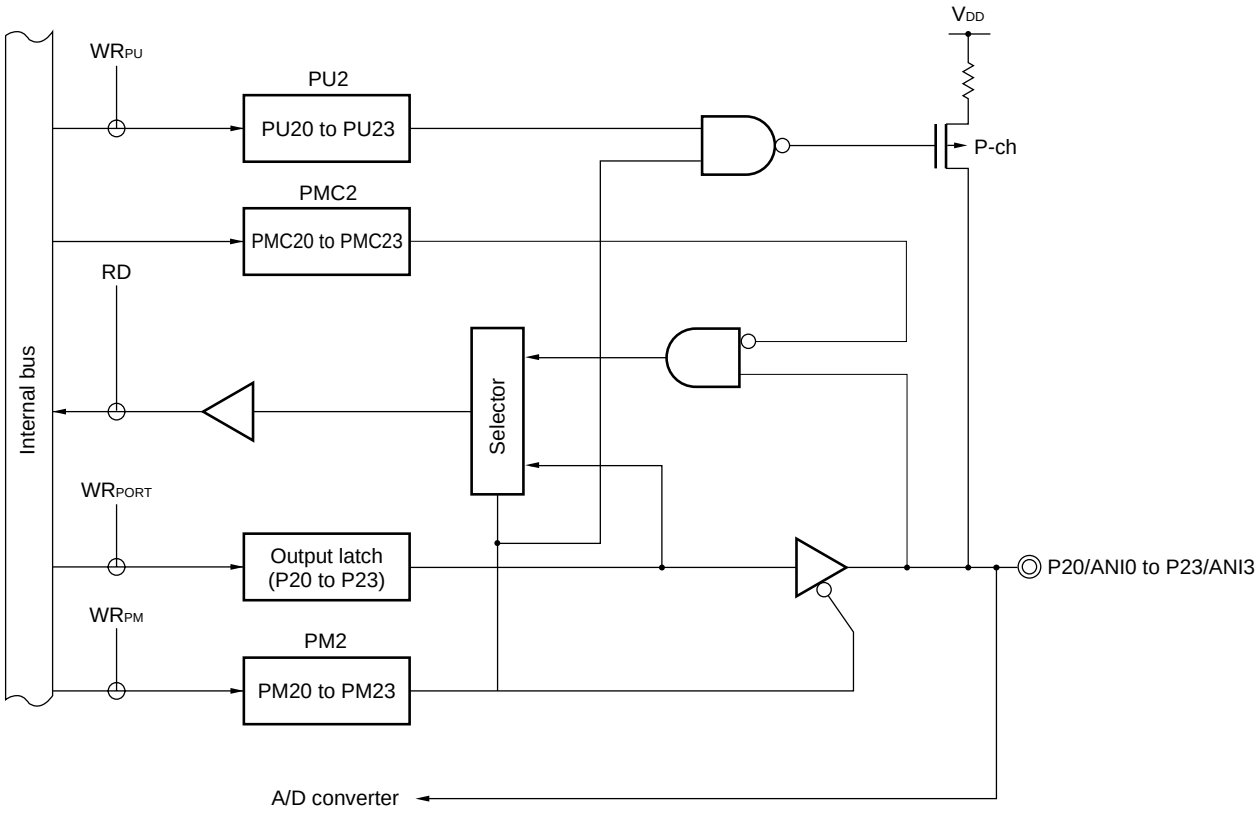
Port 2 is a 4-bit I/O port with an output latch. Each bit of this port can be set to the input or output mode by using port mode register 2 (PM2). When the P20 to P23 pins are used as an input port, an on-chip pull-up resistor can be connected in 1-bit units by using pull-up resistor option register 2 (PU2).

This port is also used as the analog input pins of the internal A/D converter.

Reset input sets port 2 to the input mode.

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

Figure 4-2. Block Diagram of P20 to P23



PU2: Pull-up resistor option register 2
PM2: Port mode register 2
PMC2: Port mode control register 2
RD: Read signal
WR××: Write signal

4.2.2 Port 3

Pins P30 and P31 constitute a 2-bit I/O port with an output latch. Each bit of this port can be set to the input or output mode by using port mode register 3 (PM3). When the P30 to P31 pins are used as an input port, an on-chip pull-up resistor can be connected in 1-bit units by using pull-up resistor option register 3 (PU3). This port is also used as external interrupt request input pins.

The P34 pin is a 1-bit input-only port and functions alternately as the $\overline{\text{RESET}}$ pin.

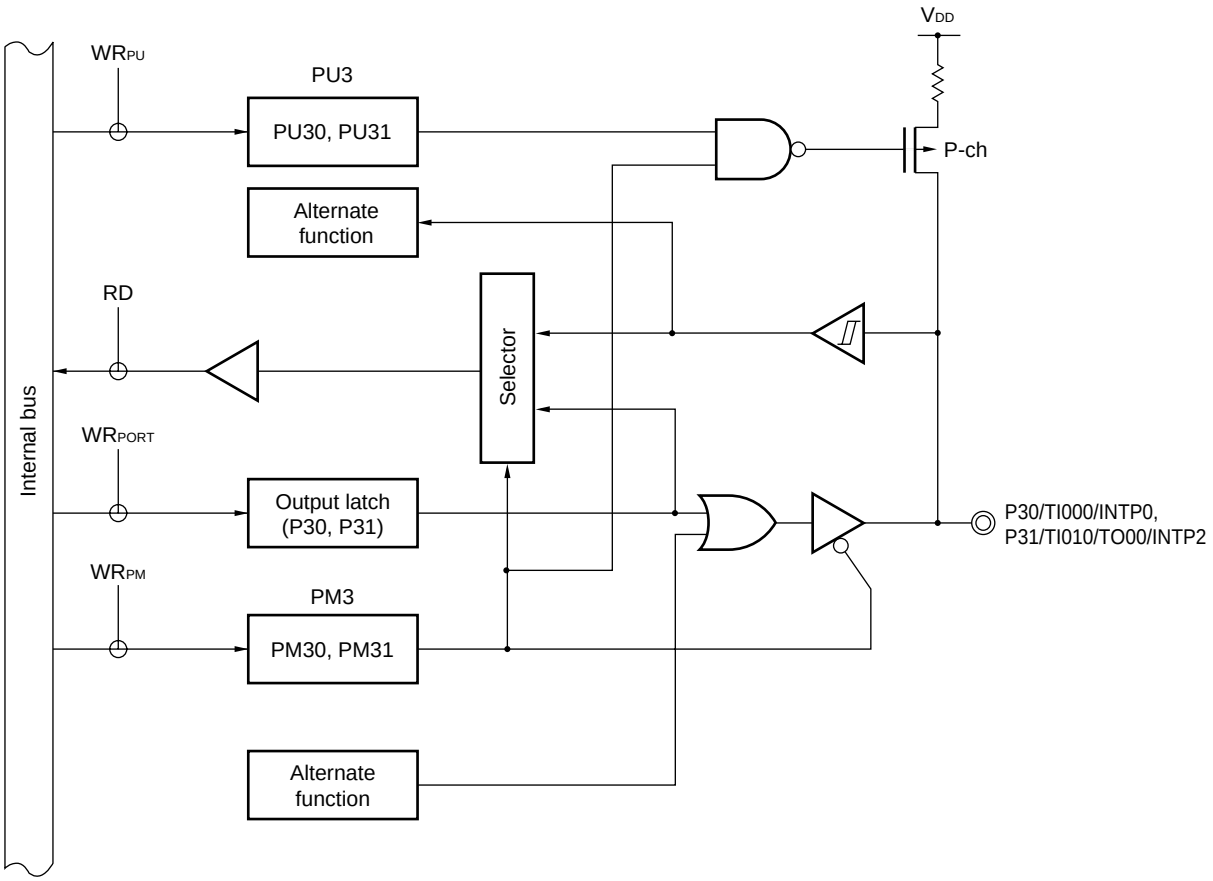
Reset input sets port 3 to the input mode.

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

Caution Because the P34 pin functions alternately as the $\overline{\text{RESET}}$ pin, if it is used as an input port pin, the function to input an external reset signal to the $\overline{\text{RESET}}$ pin cannot be used. The function of the port is selected by the option byte. For details, refer to CHAPTER 17 OPTION BYTE.

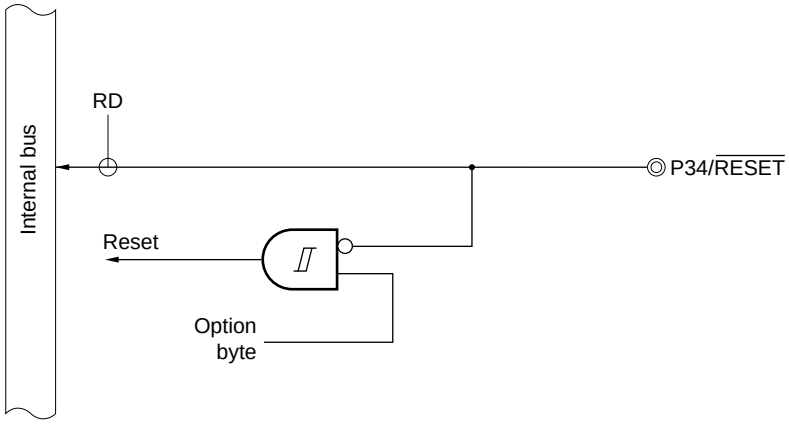
If a low level is input to the $\overline{\text{RESET}}$ pin before the option byte is referenced again after reset is released by the POC circuit, the 78K0S/KA1+ is reset and is held in the reset state until a high level is input to the $\overline{\text{RESET}}$ pin.

Figure 4-3. Block Diagram of P30 and P31



PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
WR××: Write signal

Figure 4-4. Block Diagram of P34



RD: Read signal

4.2.3 Port 4

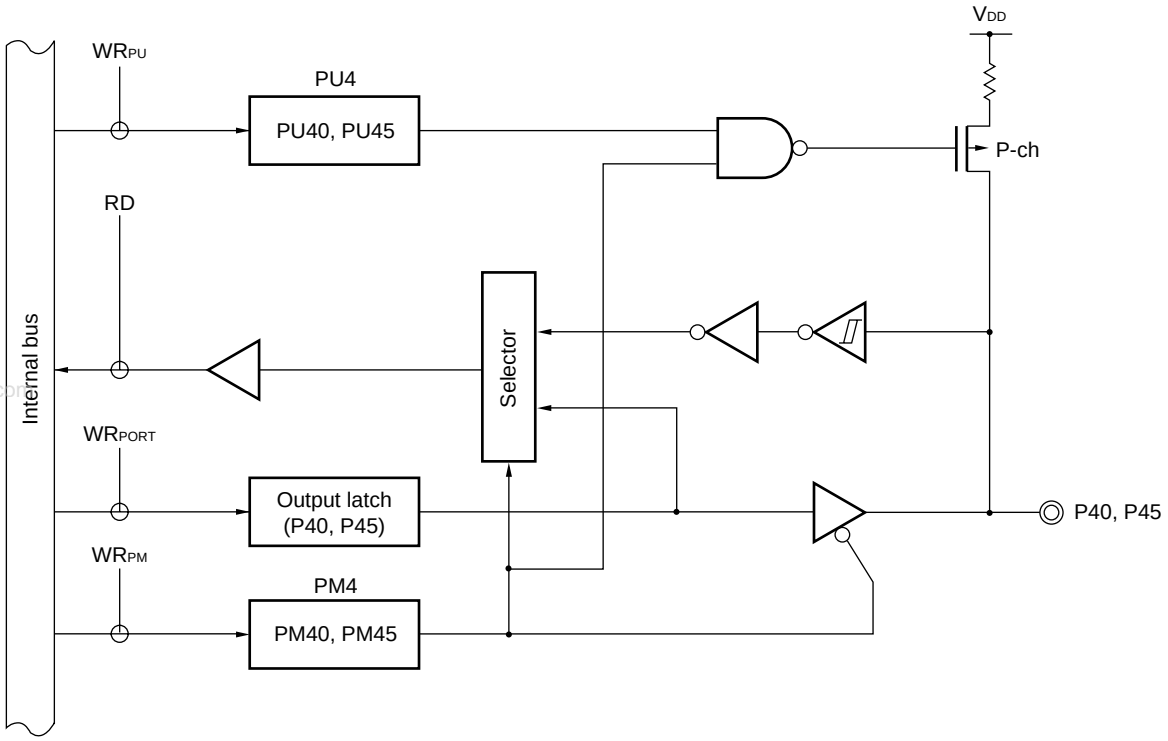
Port 4 is a 6-bit I/O port with an output latch. Each bit of this port can be set to the input or output mode by using port mode register 4 (PM4). When the P40 to P45 pins are used as an input port, an on-chip pull-up resistor can be connected in 1-bit units by using pull-up resistor option register 4 (PU4).

Alternate functions include external interrupt request input, serial interface data I/O, and timer output.

Reset input sets port 4 to the input mode.

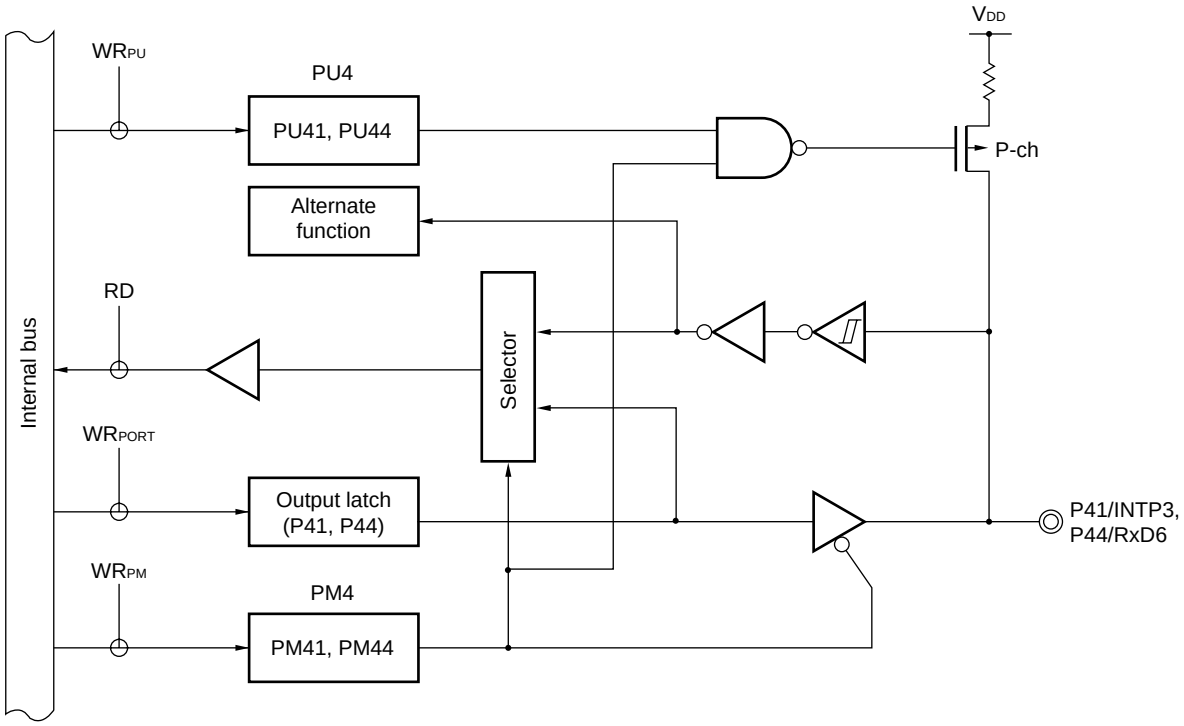
Figures 4-5 to 4-8 show the block diagrams of port 4.

Figure 4-5. Block Diagram of P40 and P45



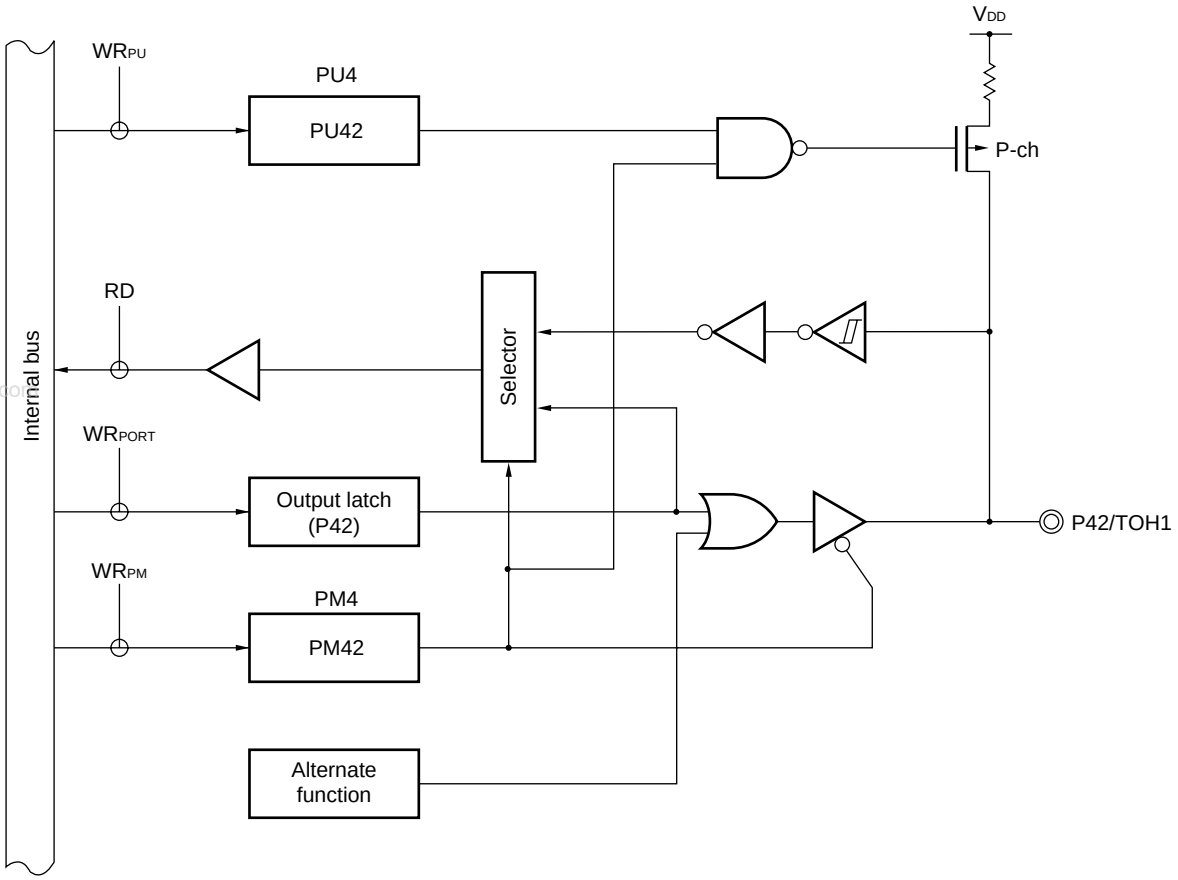
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
RD: Read signal
WR××: Write signal

Figure 4-6. Block Diagram of P41 and P44



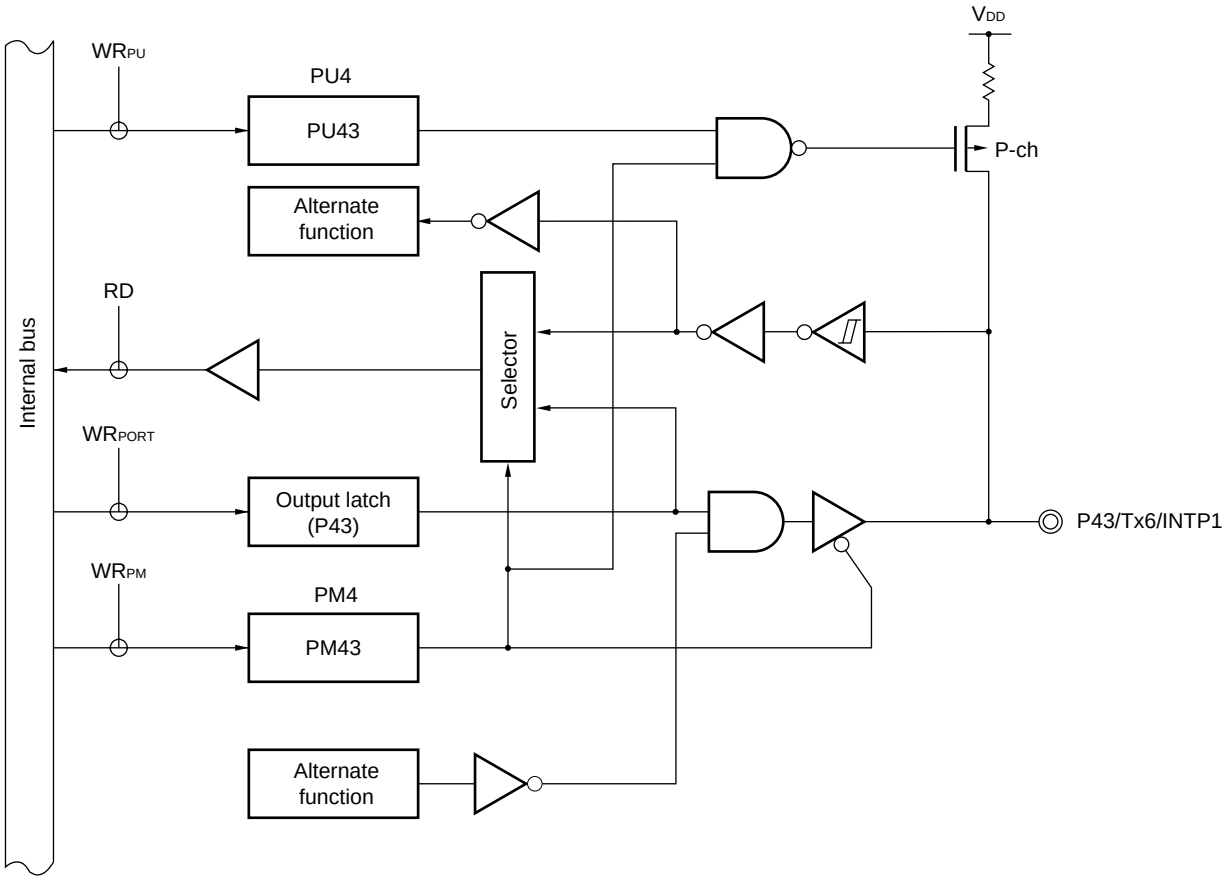
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
RD: Read signal
WR××: Write signal

Figure 4-7. Block Diagram of P42



PU4: Pull-up resistor option register 4
PM4: Port mode register 4
RD: Read signal
WR××: Write signal

Figure 4-8. Block Diagram of P43



PU4: Pull-up resistor option register 4
PM4: Port mode register 4
RD: Read signal
WR××: Write signal

4.2.4 Port 12

Port 12 is a 3-bit I/O port with an output latch. Each bit of this port can be set to the input or output mode by using port mode register 12 (PM12). When the P123 pin is used as an input port, an on-chip pull-up resistor can be connected by using pull-up resistor option register 12 (PU12).

Reset input sets port 12 to the input mode.

The P121 and P122 pins are also used as the X1 and X2 pins of the system clock oscillator. The functions of the P121 and P122 pins differ, therefore, depending on the selected system clock oscillator. The following three system clock oscillators can be used.

(1) High-speed Ring-OSC circuit

The P121 and P122 pins can be used as I/O port pins.

(2) Crystal/ceramic oscillator

The P121 and P122 pins cannot be used as I/O port pins because they are used as the X1 and X2 pins.

(3) External clock input

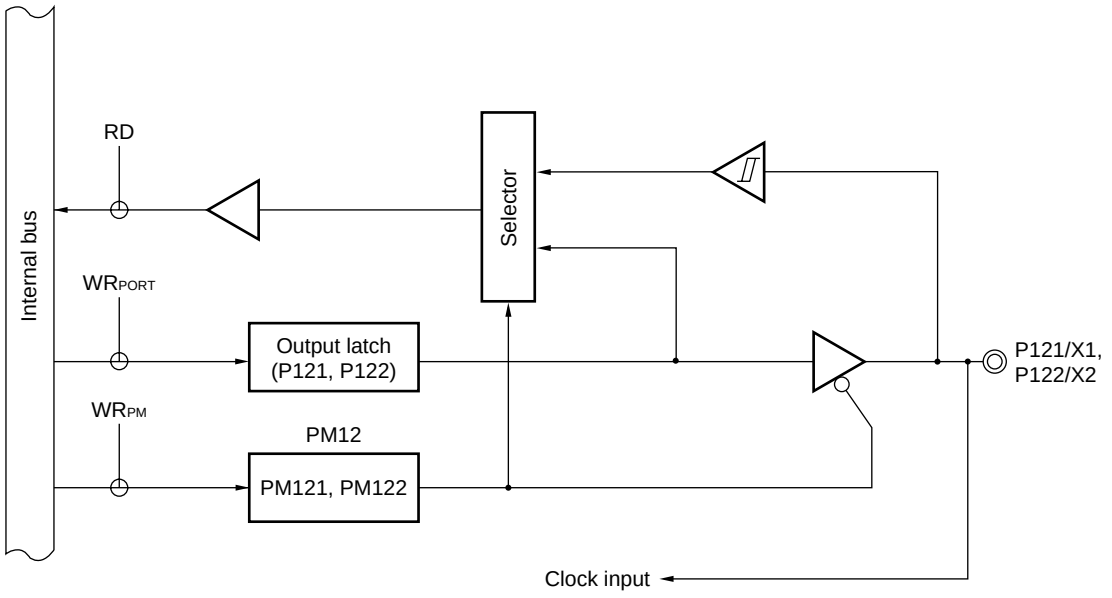
The P121 pin is used as the X1 pin to input an external clock, and therefore it cannot be used as an I/O port pin.

The P122 pin can be used as an I/O port pin.

The system clock oscillation is selected by the option byte. For details, refer to **CHAPTER 17 OPTION BYTE**.

Figures 4-9 to 4-10 show the block diagrams of port 12.

Figure 4-9. Block Diagram of P121 and P122

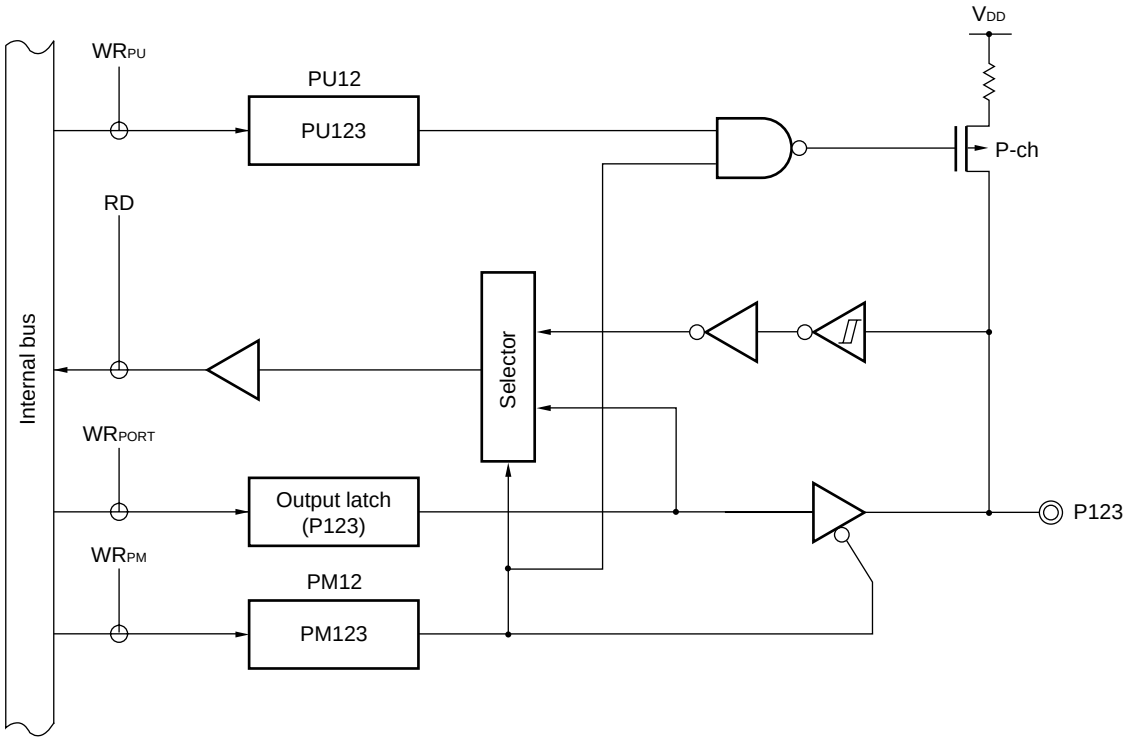


PM12: Port mode register 12

RD: Read signal

WR××: Write signal

Figure 4-10. Block Diagram of P123



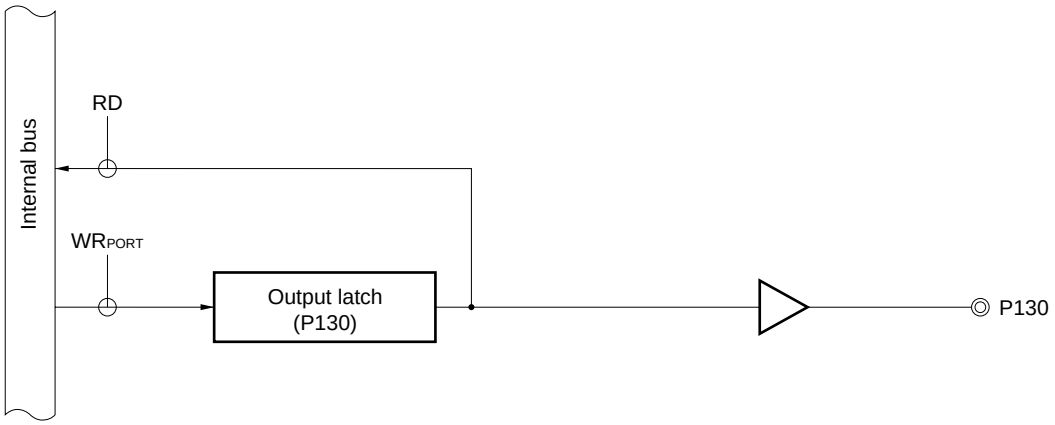
PU12: Pull-up resistor option register 12
PM12: Port mode register 12
RD: Read signal
WR××: Write signal

4.2.5 Port 13

This is a 1-bit output-only port.

Figure 4-11 shows the block diagram of port 13.

Figure 4-11. Block Diagram of P130



RD: Read signal
WR××: Write signal

Remark When a reset is input, P130 outputs a low level. If P130 outputs a high level immediately after reset is released, the output signal of P130 can be used as a dummy CPU reset signal.

4.3 Registers Controlling Port Functions

The ports are controlled by the following four types of registers.

- Port mode registers (PM2, PM3, PM4, PM12)
- Port registers (P2, P3, P4, P12, P13)
- Port mode control register 2 (PMC2)
- Pull-up resistor option registers (PU2, PU3, PU4, PU12)

(1) Port mode registers (PM2, PM3, PM4, PM12)

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

Each port mode register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets these registers to FFH.

When a port pin is used as an alternate-function pin, set its port mode register and output latch as shown in Table 4-3.

Caution Because P30, P31, and P43 are also used as external interrupt pins, the corresponding interrupt request flag is set if each of these pins is set to the output mode and its output level is changed. To use the port pin in the output mode, therefore, set the corresponding interrupt mask flag to 1 in advance.

Figure 4-12. Format of Port Mode Register

Address: FF22H, After reset: FFH, R/W

Symbol	7	6	5	4	3	2	1	0
PM2	1	1	1	1	PM23	PM22	PM21	PM20

Address: FF23H, After reset: FFH, R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	1	PM31	PM30

Address: FF24H, After reset: FFH, R/W

Symbol	7	6	5	4	3	2	1	0
PM4	1	1	PM45	PM44	PM43	PM42	PM41	PM40

Address: FF2CH, After reset: FFH, R/W

Symbol	7	6	5	4	3	2	1	0
PM12	1	1	1	1	PM123	PM122	PM121	1

PMmn	Selection of I/O mode of Pmn pin (m = 2, 3, 4, or 12; n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

(2) Port registers (P2, P3, P4, P12, P13)

These registers are used to write data to be output from the corresponding port pin to an external device connected to the chip.

When a port register is read, the pin level is read in the input mode, and the value of the output latch of the port is read in the output mode.

P20 to P23, P30, P31, P34, P40 to P45, P121 to P123, and P130 are set by using a 1-bit or 8-bit memory manipulation instruction.

Reset input sets these registers to 00H.

Figure 4-13. Format of Port Register

Address: FF02H, After reset: 00H (Output latch) R/W

Symbol	7	6	5	4	3	2	1	0
P2	0	0	0	0	P23	P22	P21	P20

Address: FF03H, After reset: 00H[Note] (Output latch) R/W[Note]

Symbol	7	6	5	4	3	2	1	0
P3	0	0	0	P34	0	0	P31	P30

Address: FF04H, After reset: 00H (Output latch) R/W

Symbol	7	6	5	4	3	2	1	0
P4	0	0	P45	P44	P43	P42	P41	P40

Address: FF0CH, After reset: 00H (Output latch) R/W

Symbol	7	6	5	4	3	2	1	0
P12	0	0	0	0	P123	P122	P121	0

Address: FF0DH, After reset: 00H (Output latch) R/W

Symbol	7	6	5	4	3	2	1	0
P13	0	0	0	0	0	0	0	P130

Pmn	m = 2, 3, 4, 12, or 13; n = 0-7	
	Controls of output data (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Note Because P34 is read-only, its reset value is undefined.

(3) Port mode control register 2 (PMC2)

This register specifies the port mode or alternate function (A/D converter) of port 2.

Each bit of the PMC2 register corresponds to each pin of port 2 and can be specified in 1-bit units.

PMC2 is set by using a 1-bit or 8-bit memory manipulation instruction.

Reset input sets PMC2 to 00H.

Figure 4-14. Format of Port Mode Control Register 2

Address: FF84H, After reset: R/W

Symbol	7	6	5	4	3	2	1	0
PMC2	0	0	0	0	PMC23	PMC22	PMC21	PMC20

PMC2n	Specification of operation mode (n = 0 to 3)
0	Port mode
1	Alternate-function mode (A/D converter)

Table 4-3. Setting of Port Mode Register, Port Register (Output Latch), and Port Mode Control Register When Alternate Function Is Used

Pin Name	Alternate-Function Pin		PM××	P××	PMC2n (n = 0 to 3)
	Name	I/O			
P20 to P23	ANI0 to ANI3	Input	1	×	1
P30	TI000	Input	1	×	–
	INTP0	Input	1	×	–
P31	TO00	Output	0	0	–
	TI010	Input	1	×	–
	INTP2	Input	1	×	–
P41	INTP3	Input	1	×	–
P42	TOH1	Output	0	0	–
P43	TxD6	Output	0	1	–
	INTP1	Input	1	×	–
P44	RxD6	Input	1	×	–

Remark ×: don't care
PM××: Port mode register, P××: Port register (output latch of port)
PMC2×: Port mode control register

(4) Pull-up resistor option registers (PU2, PU3, PU4, PU12)

These registers are used to specify whether an on-chip pull-up resistor is connected to P20 to P23, P30, P31, P40 to P45, and P123. By setting PU2, PU3, PU4, or PU12, an on-chip pull-up resistor can be connected to the port pin corresponding to the bit of PU2, PU3, PU4, or PU12.

PU2, PU3, PU4, and PU12 are set by using a 1-bit or 8-bit memory manipulation instruction.

Reset input set these registers to 00H.

Figure 4-15. Format of Pull-up Resistor Option Register

Address: FF32H, After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU2	0	0	0	0	PU23	PU22	PU21	PU20

Address: FF33H, After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU3	0	0	0	0	0	0	PU31	PU30

Address: FF34H, After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU4	0	0	PU45	PU44	PU43	PU42	PU41	PU40

Address: FF3CH, After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PU12	0	0	0	0	PU123	0	0	0

PUmn	Selection of connection of on-chip pull-up resistor of Pmn (m = 2, 3, 4, or 12; n = 0 to 7)
0	Does not connect on-chip pull-up resistor
1	Connects on-chip pull-up resistor

4.4 Operation of Port Function

The operation of a port differs, as follows, depending on the setting of the I/O mode.

Caution Although a 1-bit memory manipulation instruction manipulates 1 bit, it accesses a port in 8-bit units. Therefore, the contents of the output latch of a pin in the input mode, even if it is not subject to manipulation by the instruction, are undefined in a port with a mixture of inputs and outputs.

4.4.1 Writing to I/O port

(1) In output mode

A value can be written to the output latch by a transfer instruction. In addition, the contents of the output latch are output from the pin. Once data is written to the output latch, it is retained until new data is written to the output latch.

Reset input cleans the data in the output latch.

(2) In input mode

A value can be written to the output latch by a transfer instruction. Because the output buffer is off, however, the pin status remains unchanged.

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

4.4.2 Reading from I/O port

(1) In output mode

The contents of the output latch can be read by a transfer instruction. The contents of the output latch remain unchanged.

(2) In input mode

The pin status can be read by a transfer instruction. The contents of the output latch remain unchanged.

4.4.3 Operations on I/O port

(1) In output mode

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

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

Reset input clears the data in the output latch.

(2) In input mode

The pin level is read and an operation is performed on its contents. The operation result is written to the output latch. However, the pin status remains unchanged because the output buffer is off.

CHAPTER 5 CLOCK GENERATORS

5.1 Functions of Clock Generators

The clock generators include a circuit that generates a clock (system clock) to be supplied to the CPU and peripheral hardware, and a circuit that generates a clock (interval time generation clock) to be supplied to the watchdog timer and 8-bit timer H1 (TMH1).

5.1.1 System clock oscillators

The following three types of system clock oscillators are used.

- High-speed Ring-OSC oscillator
  This circuit internally oscillates a clock of 8 MHz (TYP.). Its oscillation can be stopped by execution of the STOP instruction.
  If the high-speed Ring-OSC oscillator is selected to supply the system clock, the X1 and X2 pins can be used as I/O port pins.

- Crystal/ceramic oscillator
  This circuit oscillates a clock with a crystal/ceramic oscillator connected across the X1 and X2 pins. It can oscillate a clock of 500 kHz to 10 MHz. Oscillation of this circuit can be stopped by execution of the STOP instruction.

- External clock input circuit
  This circuit supplies a clock from an external IC to the X1 pin. A clock of 500 kHz to 10 MHz can be supplied. Internal clock supply can be stopped by execution of the STOP instruction.
  If the external clock input is selected as the system clock, the X2 pin can be used as an I/O port pin.

The system clock source is selected by using the option byte. For details, refer to **CHAPTER 17 OPTION BYTE**.
When using the X1 and X2 pins as I/O port pins, refer to **CHAPTER 4 PORT FUNCTIONS** for details.

5.1.2 Clock oscillator for interval time generation

The following circuit is used as a clock oscillator for interval time generation.

- Low-speed Ring-OSC oscillator
  This circuit oscillates a clock of 240 kHz (TYP.). Its oscillation can be stopped by using the low-speed Ring-OSC mode register (LSRCM) when it is specified by the option byte that its oscillation can be stopped by software.

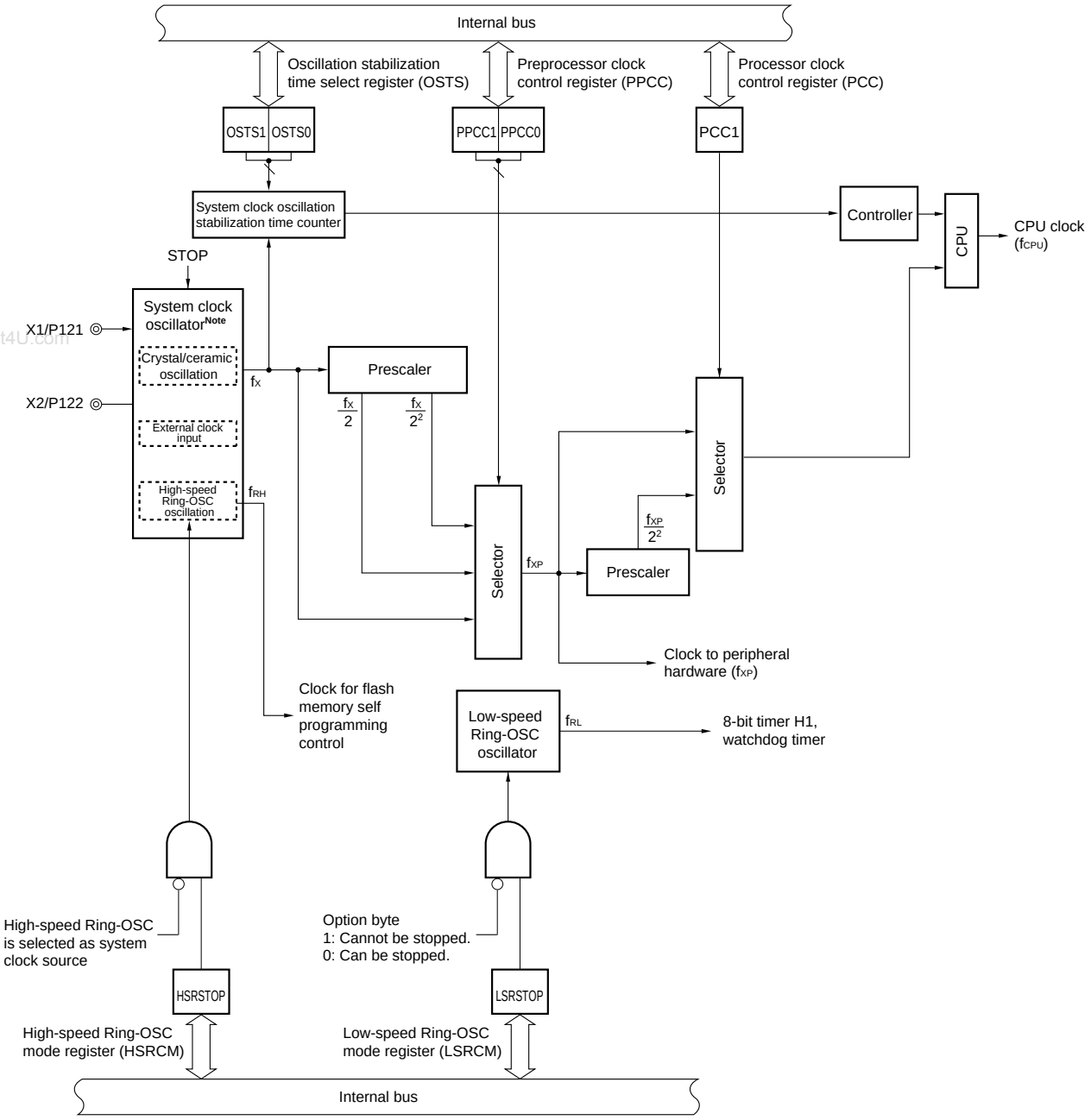
5.2 Configuration of Clock Generators

The clock generators consist of the following hardware.

Table 5-1. Configuration of Clock Generators

Item	Configuration
Control registers	Processor clock control register (PCC) Preprocessor clock control register (PPCC) Low-speed Ring-OSC mode register (LSRCM) High-speed Ring-OSC mode register (HSRCM) Oscillation stabilization time select register (OSTS)
Oscillators	Crystal/ceramic oscillator High-speed Ring-OSC oscillator External clock input circuit Low-speed Ring-OSC oscillator

Figure 5-1. Block Diagram of Clock Generators



Note Select the high-speed Ring-OSC oscillator, crystal/ceramic oscillator, or external clock input as the system clock source by using the option byte.

5.3 Registers Controlling Clock Generators

The clock generators are controlled by the following five registers.

- Processor clock control register (PCC)
- Preprocessor clock control register (PPCC)
- Low-speed Ring-OSC mode register (LSRCM)
- High-speed Ring-OSC mode register (HSRCM)
- Oscillation stabilization time select register (OSTS)

(1) Processor clock control register (PCC) and pre-processor clock control register (PPCC)

These registers are used to specify the division ratio of the system clock.

PCC and PPCC are set by using a 1-bit or 8-bit memory manipulation instruction.

Reset input sets PCC and PPCC to 02H.

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

Address: FFFBH, After reset: 02H, R/W

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

Figure 5-3. Format of Preprocessor Clock Control Register (PPCC)

Address: FFF3H, After reset: 02H, R/W

Symbol	7	6	5	4	3	2	1	0
PPCC	0	0	0	0	0	0	PPCC1	PPCC0

PPCC1	PPCC0	PCC1	Selection of CPU clock (f_{CPU})
0	0	0	f_X
0	1	0	$f_X/2$ Note 1
0	0	1	$f_X/2^2$
1	0	0	$f_X/2^2$ Note 2
0	1	1	$f_X/2^3$ Note 1
1	0	1	$f_X/2^4$ Note 2
Other than above			Setting prohibited

Notes 1. If PPCC = 01H, the clock (f_{XP}) supplied to the peripheral hardware is $f_X/2$.

2. If PPCC = 02H, the clock (f_{XP}) supplied to the peripheral hardware is $f_X/2^2$.

The fastest instruction of the 78K0S/KA1+ is executed in two CPU clocks. Therefore, the relationship between the CPU clock (f_{CPU}) and the minimum instruction execution time is as shown in Table 5-2.

Table 5-2. Relationship Between CPU Clock and Minimum Instruction Execution Time

CPU Clock (f_{CPU}) Note	Minimum Instruction Execution Time: $2/f_{CPU}$	
	High-speed Ring-OSC clock (at 8.0 MHz (TYP.))	Crystal/ceramic oscillation clock or external clock input (at 10.0 MHz)
f_X	0.25 μs	0.2 μs
$f_X/2$	0.5 μs	0.4 μs
$f_X/2^2$	1.0 μs	0.8 μs
$f_X/2^3$	2.0 μs	1.6 μs
$f_X/2^4$	4.0 μs	3.2 μs

Note The CPU clock (high-speed Ring-OSC clock, crystal/ceramic oscillation clock, or external clock input) is selected by the option byte.

(2) Low-speed Ring-OSC mode register (LSRCM)

This register is used to select the operation mode of the low-speed Ring-OSC oscillator (240 kHz (TYP.)).

This register is valid when it is specified by the option byte that the low-speed Ring-OSC oscillator can be stopped by software. If it is specified by the option byte that the low-speed Ring-OSC oscillator cannot be stopped by software, setting of this register is invalid, and the low-speed Ring-OSC oscillator continues oscillating. In addition, the source clock of WDT is fixed to the low-speed Ring-OSC oscillator. For details, refer to **CHAPTER 9 WATCHDOG TIMER**.

LSRCM can be set by using an 8-bit memory manipulation instruction.

Reset input sets LSRCM to 00H.

Figure 5-4. Format of Low-Speed Ring-OSC Mode Register (LSRCM)

Address: FF58H, After reset: 00H, R/W

Symbol	7	6	5	4	3	2	1	0
LSRCM	0	0	0	0	0	0	0	LSRSTOP

LSRSTOP	Oscillation/stop of low-speed Ring-OSC
0	Low-speed Ring-OSC oscillates
1	Low-speed Ring-OSC stops

(3) High-speed Ring-OSC mode register (HSRCM)

This register is used to select the operation mode of the high-speed Ring-OSC oscillator that generates a clock (8 MHz (TYP.)) for controlling self programming of the flash memory.

This register is valid when crystal/ceramic oscillation or external clock input is selected as the system clock source by the option byte. Setting of this register is invalid when the high-speed Ring-OSC oscillator is selected by the option byte.

If crystal/ceramic oscillation or external clock input is selected as the system clock source, the high-speed Ring-OSC oscillator must be oscillated during self-programming of the flash memory. While self-programming is not executed, stop oscillation of the high-speed Ring-OSC oscillator to reduce the current consumption. For self-programming of the flash memory, refer to **CHAPTER 18 FLASH MEMORY**.

HSRCM is set by using an 8-bit memory manipulation instruction.

Reset input sets HSRCM to 00H.

Figure 5-5. Format of High-Speed Ring-OSC Mode Register (HSRCM)

Address: FF5AH, After reset: 00H, R/W

Symbol	7	6	5	4	3	2	1	0
HSRCM	0	0	0	0	0	0	0	HSRSTOP

HSRSTOP	Oscillation/stops of high-speed Ring-OSC
0	High-speed Ring-OSC oscillates
1	High-speed Ring-OSC oscillates stops

(4) Oscillation stabilization time select register (OSTS)

This register is used to select oscillation stabilization time of the clock supplied from the oscillator when the STOP mode is released. The wait time set by OSTS is valid only when the crystal/ceramic oscillation clock is selected as the system clock and after the STOP mode is released. If the high-speed Ring-OSC oscillator or external clock input is selected as the system clock source, no wait time elapses.

The system clock oscillator and the oscillation stabilization time that elapses after power application or release of reset are selected by the option byte. For details, refer to **CHAPTER 17 OPTION BYTE**.

OSTS is set by using an 8-bit memory manipulation instruction.

Figure 5-6. Format of Oscillation Stabilization Time Select Register (OSTS)

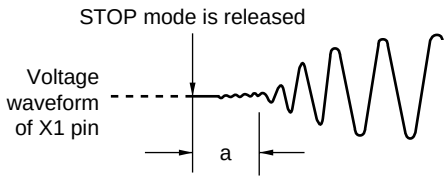
Address: FFF4H, After reset: Undefined, R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	0	OSTS1	OSTS0

OSTS1	OSTS0	Selection of oscillation stabilization time
0	0	$2^{10}/f_X$ (102.4 μs)
0	1	$2^{12}/f_X$ (409.6 μs)
1	0	$2^{15}/f_X$ (3.27 ms)
1	1	$2^{17}/f_X$ (13.1 ms)

Cautions 1. To set and then release the STOP mode, set the oscillation stabilization time as follows.
Expected oscillation stabilization time of resonator ≤ Oscillation stabilization time set by OSTS

2. The wait time after the STOP mode is released does not include the time from the release of the STOP mode to the start of clock oscillation ("a" in the figure below), regardless of whether STOP mode was released by reset input or interrupt generation.



Caution 3. The oscillation stabilization time that elapses on power application or after release of reset is selected by the option byte. For details, refer to CHAPTER 17 OPTION BYTE.

Remarks 1. ( ): f_X = 10 MHz

2. Determine the oscillation stabilization time of the resonator by checking the characteristics of the resonator to be used.

5.4 System Clock Oscillators

The following three types of system clock oscillators are available.

- High-speed Ring-OSC oscillator: Internally oscillates a clock of 8 MHz (TYP.).
- Crystal/ceramic oscillator: Oscillates a clock of 500 kHz to 10 MHz.
- External clock input circuit: Supplies a clock of 500 kHz to 10 MHz to the X1 pin.

5.4.1 High-speed Ring-OSC oscillator

The 78K0S/KA1+ includes a high-speed Ring-OSC oscillator (8 MHz (TYP.)).

If the high-speed Ring-OSC is selected by the option byte as the clock source, the X1 and X2 pins can be used as I/O port pins.

For details of the option byte, refer to **CHAPTER 17 OPTION BYTE**. For details of I/O ports, refer to **CHAPTER 4 PORT FUNCTIONS**.

5.4.2 Crystal/ceramic oscillator

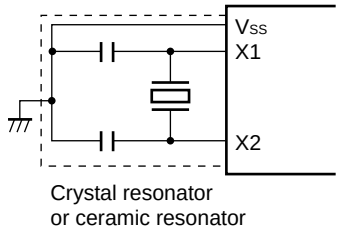
The crystal/ceramic oscillator oscillates using a crystal or ceramic resonator connected between the X1 and X2 pins.

If the crystal/ceramic oscillator is selected by the option byte as the system clock source, the X1 and X2 pins are used as crystal or ceramic resonator connection pins.

For details of the option byte, refer to **CHAPTER 17 OPTION BYTE**. For details of I/O ports, refer to **CHAPTER 4 PORT FUNCTIONS**.

Figure 5-7 shows the external circuit of the crystal/ceramic oscillator.

Figure 5-7. External Circuit of Crystal/Ceramic Oscillator



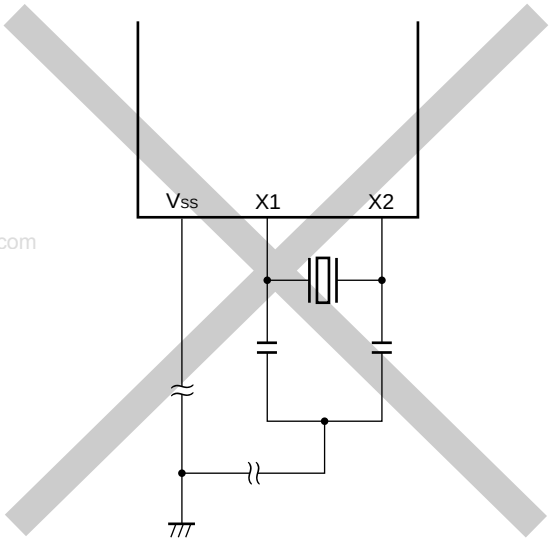
Caution When using the crystal/ceramic oscillator, wire as follows in the area enclosed by the broken lines in Figure 5-7 to avoid an adverse effect from wiring capacitance.

- **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 Vss. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

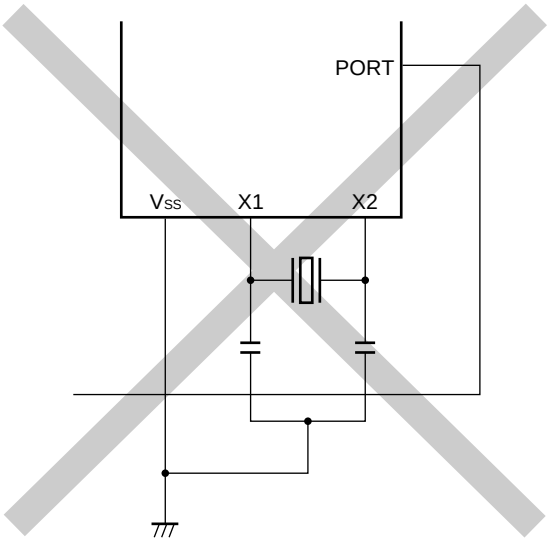
Figure 5-8 shows examples of incorrect resonator connection.

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

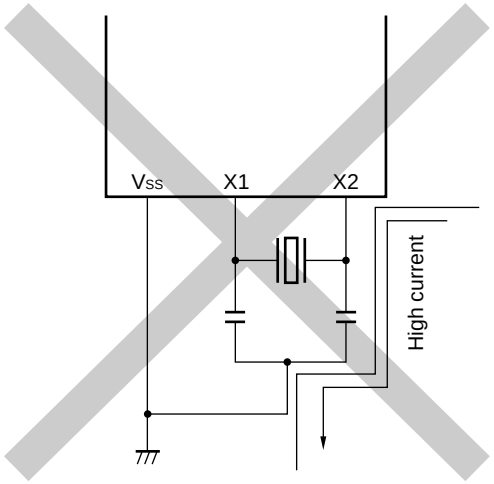
(a) Too long wiring of connected circuit



(b) Crossed signal lines



(c) Wiring near high fluctuating current



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

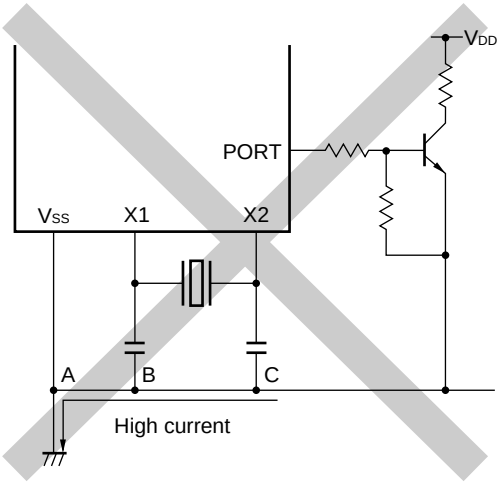
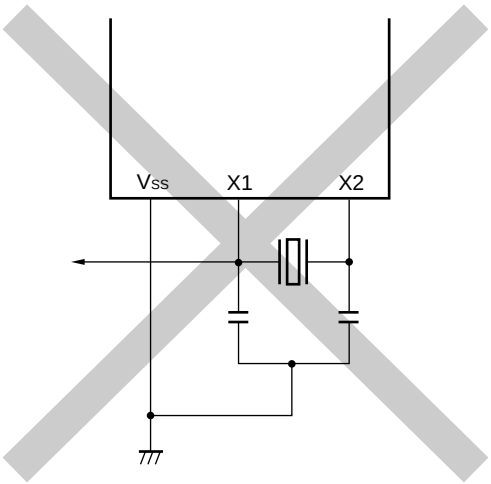


Figure 5-8. Examples of Incorrect Resonator Connection (2/2)

(e) Signals are fetched



5.4.3 External clock input circuit

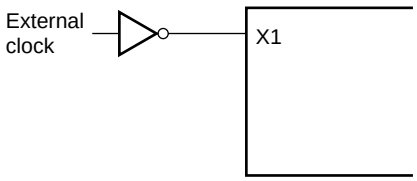
This circuit supplies a clock from an external IC to the X1 pin.

If external clock input is selected by the option byte as the system clock source, the X2 pin can be used as an I/O port pin.

For details of the option byte, refer to **CHAPTER 17 OPTION BYTE**. For details of I/O ports, refer to **CHAPTER 4 PORT FUNCTIONS**.

Figure 5-9 shows an external circuit of the external clock input circuit.

Figure 5-9. External Circuit of External Clock Input Circuit



5.4.4 Prescaler

The prescaler divides the clock (f_X) output by the system clock oscillator to generate a clock (f_{XP}) to be supplied to the peripheral hardware. It also divides the clock to peripheral hardware (f_{XP}) to generate a clock to be supplied to the CPU.

Remark The clock output by the oscillator selected by the option byte (high-speed Ring-OSC oscillator, crystal/ceramic oscillator, or external clock input circuit) is divided. For details of the option byte, refer to **CHAPTER 17 OPTION BYTE**.

5.5 Operation of CPU Clock Generator

A clock (f_{CPU}) is supplied to the CPU from the system clock (f_X) oscillated by one of the following three types of oscillators.

- High-speed Ring-OSC oscillator: Internally oscillates a clock of 8 MHz (TYP.).
- Crystal/ceramic oscillator: Oscillates a clock of 500 kHz to 10 MHz.
- External clock input circuit: Supplies a clock of 500 kHz to 10 MHz to X1 pin.

The system clock oscillator is selected by the option byte. For details of the option byte, refer to **CHAPTER 17 OPTION BYTE**.

(1) High-speed Ring-OSC oscillator

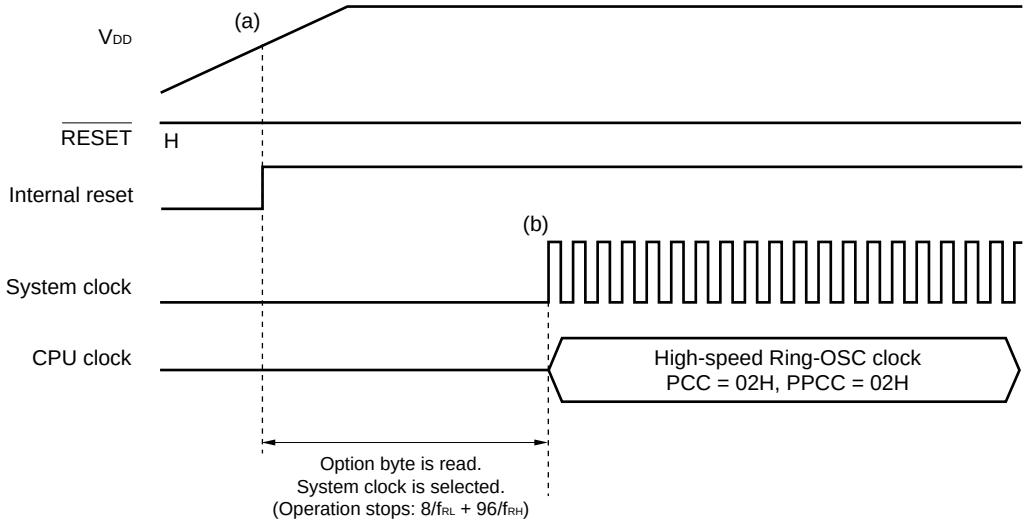
When the high-speed Ring-OSC oscillator is selected by the option byte, the following is possible.

- Shortening of start time
  If the high-speed Ring-OSC oscillator is selected as the oscillator, the CPU can be started without having to wait for the oscillation stabilization time of the system clock. Therefore, the start time can be shortened.
- Improvement of expandability
  If the high-speed Ring-OSC oscillator is selected as the oscillator, the X1 and X2 pins can be used as I/O port pins. For details, refer to **CHAPTER 4 PORT FUNCTIONS**.

Figures 5-10 and 5-11 show the timing chart and status transition diagram of the default start by the high-speed Ring-OSC oscillator.

Remark When the high-speed Ring-OSC oscillator is used, the clock accuracy is ±5%.

Figure 5-10. Timing Chart of Default Start by High-Speed Ring-OSC Oscillator

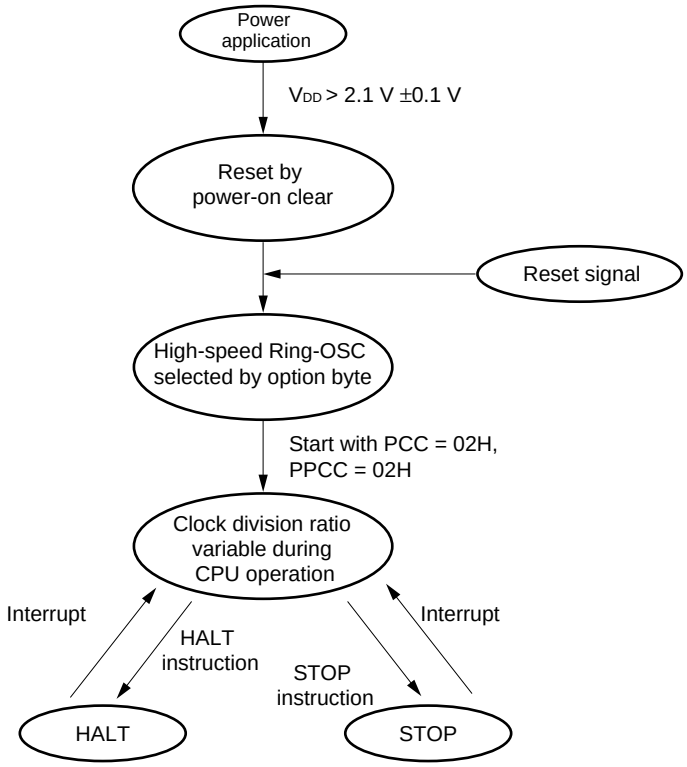


Remark f_{RL}: Low-speed Ring-OSC clock oscillation frequency
f_{RH}: High-speed Ring-OSC clock oscillation frequency

(a) The internal reset signal is generated by the power-on clear function on power application, the option byte is referenced after reset, and the system clock is selected.

(b) The option byte is referenced and the system clock is selected. Then the high-speed Ring-OSC clock operates as the system clock.

Figure 5-11. Status Transition of Default Start by High-Speed Ring-OSC



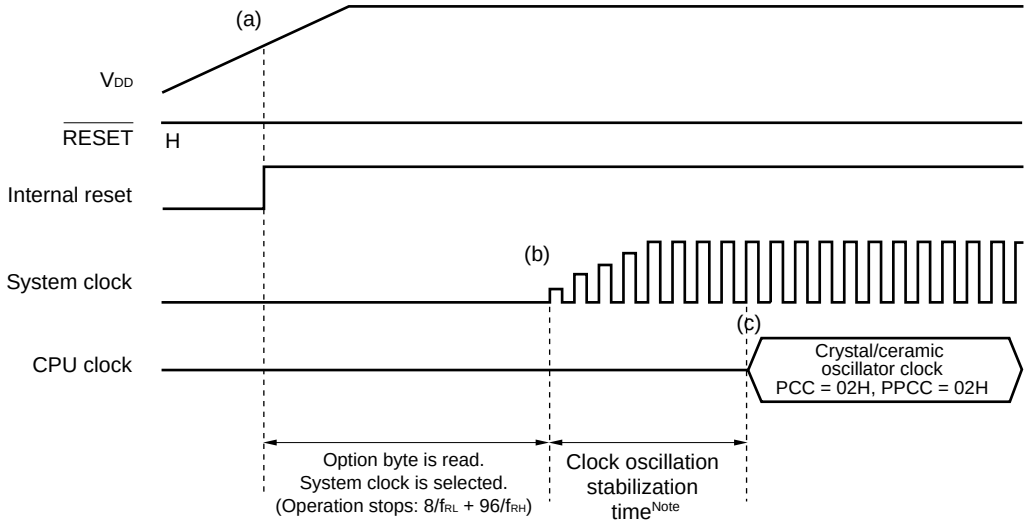
Remark PCC: Processor clock control register
PPCC: Preprocessor clock control register

(2) Crystal/ceramic oscillator

If crystal/ceramic oscillation is selected by the option byte, a clock frequency of 500 kHz to 10 MHz can be selected and the accuracy of processing is improved because the frequency deviation is small, as compared with high-speed Ring-OSC oscillation (8 MHz (TYP.)).

Figures 5-12 and 5-13 show the timing chart and status transition diagram of default start by the crystal/ceramic oscillator.

Figure 5-12. Timing Chart of Default Start by Crystal/Ceramic Oscillator



Note The clock oscillation stabilization time for default start is selected by the option byte. For details, refer to **CHAPTER 17 OPTION BYTE**. The oscillation stabilization time that elapses after the STOP mode is released is selected by the oscillation stabilization time select register (OSTS).

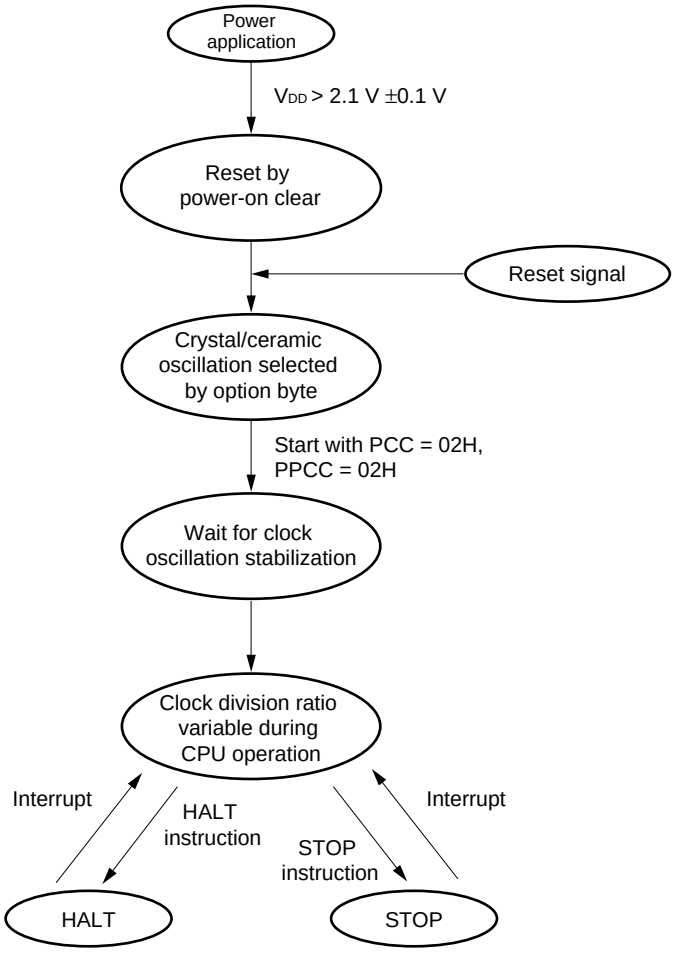
Remark f_{RL}: Low-speed Ring-OSC clock oscillation frequency
f_{RH}: High-speed Ring-OSC clock oscillation frequency

(a) The internal reset signal is generated by the power-on clear function on power application, the option byte is referenced after reset, and the system clock is selected.

(b) After high-speed Ring-OSC clock is generated, the option byte is referenced and the system clock is selected. In this case, the crystal/ceramic oscillator clock is selected as the system clock.

(c) If the system clock is the crystal/ceramic oscillator clock, it starts operating as the CPU clock after clock oscillation is stabilized. The wait time is selected by the option byte. For details, refer to **CHAPTER 17 OPTION BYTE**.

Figure 5-13. Status Transition of Default Start by Crystal/Ceramic Oscillation



Remark PCC: Processor clock control register
PPCC: Preprocessor clock control register

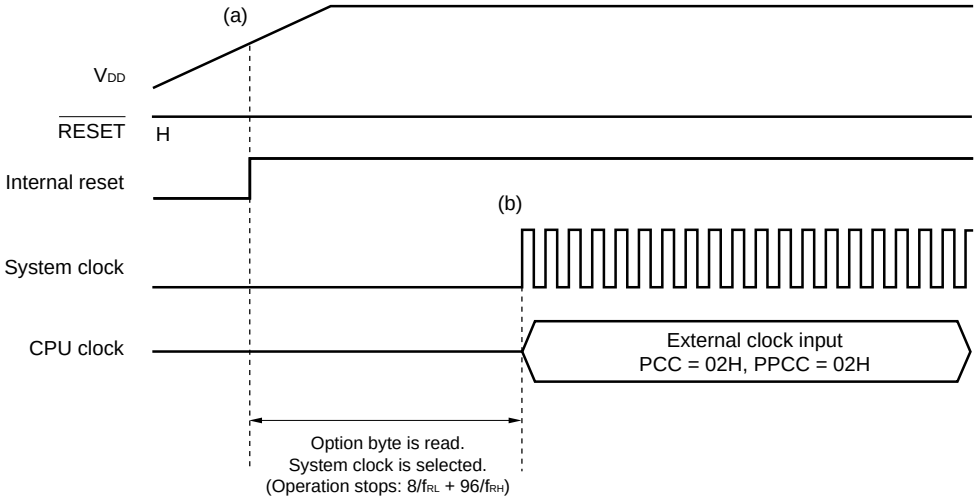
(3) External clock input circuit

If external clock input is selected by the option byte, the following is possible.

- High-speed operation
  The accuracy of processing is improved as compared with high-speed Ring-OSC oscillation (8 MHz (TYP.)) because an oscillation frequency of 500 kHz to 10 MHz can be selected and an external clock with a small frequency deviation can be supplied.
- Improvement of expandability
  If the external clock input circuit is selected as the oscillator, the X2 pin can be used as an I/O port pin. For details, refer to **CHAPTER 4 PORT FUNCTIONS**.

Figures 5-14 and 5-15 show the timing chart and status transition diagram of default start by external clock input.

Figure 5-14. Timing of Default Start by External Clock Input

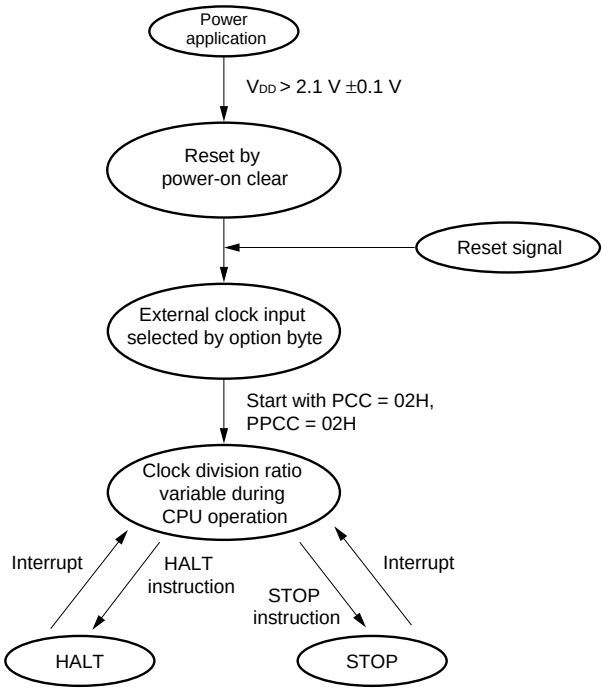


Remark f_{RL}: Low-speed Ring-OSC clock oscillation frequency
f_{RH}: High-speed Ring-OSC clock oscillation frequency

(a) The internal reset signal is generated by the power-on clear function on power application, the option byte is referenced after reset, and the system clock is selected.

(b) The option byte is referenced and the system clock is selected. Then the external clock operates as the system clock.

Figure 5-15. Status Transition of Default Start by External Clock Input



Remark PCC: Processor clock control register
PPCC: Preprocessor clock control register

5.6 Operation of Clock Generator Supplying Clock to Peripheral Hardware

The following two types of clocks are supplied to the peripheral hardware.

- Clock to peripheral hardware (f_{XP})
- Low-speed Ring-OSC clock (f_{RL})

(1) Clock to peripheral hardware

The clock to the peripheral hardware is supplied by dividing the system clock (f_X). The division ratio is selected by the pre-processor clock control register (PPCC).

Three types of frequencies are selectable: "f_X", "$f_X/2$", and "$f_X/2^2$". Table 5-3 lists the clocks supplied to the peripheral hardware.

Table 5-3. Clocks to Peripheral Hardware

PPCC1	PPCC0	Selection of clock to peripheral hardware (f_{XP})
0	0	f_X
0	1	$f_X/2$
1	0	$f_X/2^2$
1	1	Setting prohibited

(2) Low-speed Ring-OSC clock

The low-speed Ring-OSC oscillator of the clock oscillator for interval time generation is always started after release of reset, and oscillates at 240 kHz (TYP.).

It can be specified by the option byte whether the low-speed Ring-OSC oscillator can or cannot be stopped by software. If it is specified that the low-speed Ring-OSC oscillator can be stopped by software, oscillation can be started or stopped by using the low-speed Ring-OSC mode register (LSRCM). If it is specified that it cannot be stopped by software, the clock source of WDT is fixed to the low-speed Ring-OSC clock (fRL).

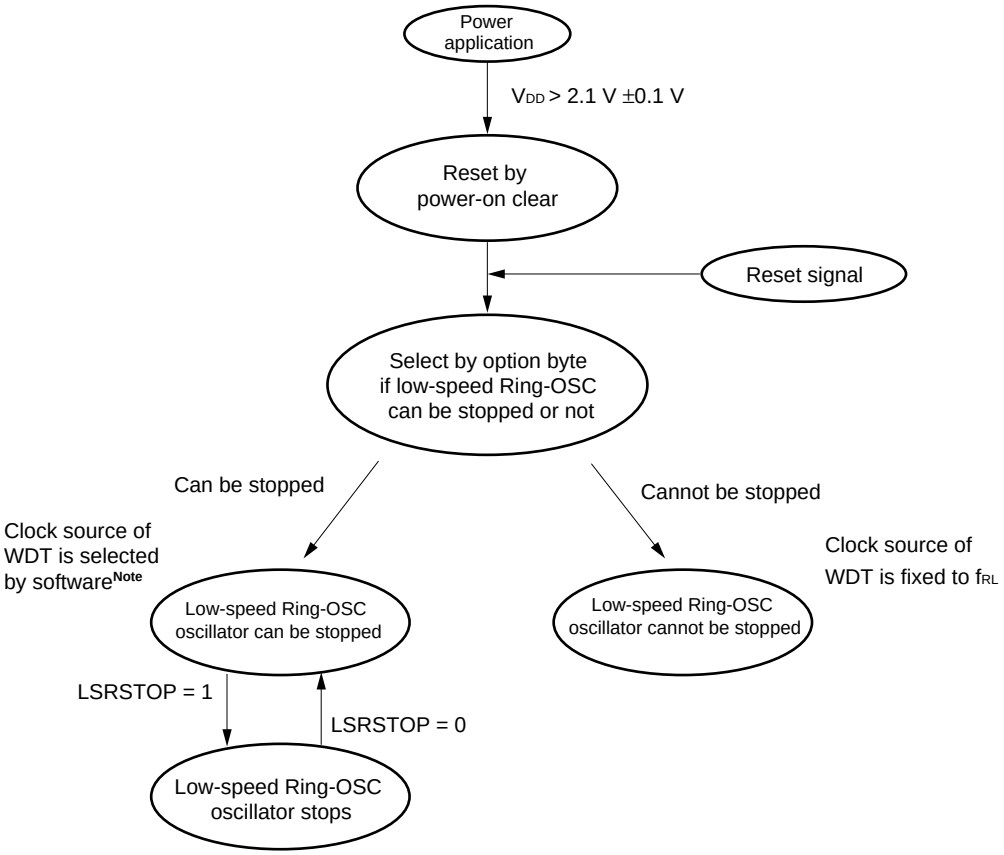
The low-speed Ring-OSC oscillator is independent of the CPU clock. If it is used as the source clock of WDT, therefore, a hang-up can be detected even if the CPU clock is stopped. If the low-speed Ring-OSC oscillator is used as a count clock source of 8-bit timer H1, 8-bit timer H1 can operate even in the standby status.

Table 5-4 shows the operation status of the low-speed Ring-OSC oscillator when it is selected as the source clock of WDT and the count clock of 8-bit timer H1. Figure 5-16 shows the status transition of the low-speed Ring-OSC oscillator.

Table 5-4. Operation Status of Low-Speed Ring-OSC Oscillator

Option Byte Setting		CPU Status	WDT Status	TMH1 Status
Can be stopped by software	LSRSTOP = 1	Operation mode	Stopped	Stopped
	LSRSTOP = 0		Operates	Operates
	LSRSTOP = 1	Standby	Stopped	Stopped
	LSRSTOP = 0		Stopped	Operates
Cannot be stopped		Operation mode	Operates	
		Standby		

Figure 5-16. Status Transition of Low-Speed Ring-OSC Oscillator



Note The clock source of the watchdog timer (WDT) is selected from f_{XP} or f_{RL}, or it may be stopped. For details, refer to **CHAPTER 9 WATCHDOG TIMER**.

CHAPTER 6 16-BIT TIMER/EVENT COUNTER 00

6.1 Functions of 16-Bit Timer/Event Counter 00

16-bit timer/event counter 00 has the following functions.

(1) Interval timer

16-bit timer/event counter 00 generates interrupt requests at the preset time interval.

- Number of counts: 2 to 65536

(2) External event counter

16-bit timer/event counter 00 can measure the number of pulses with a high-/low-level width of a signal input externally.

- Valid level pulse width: $16/f_{XP}$ or more

(3) Pulse width measurement

16-bit timer/event counter 00 can measure the pulse width of an externally input signal.

- Valid level pulse width: $2/f_{XP}$ or more

(4) Square-wave output

16-bit timer/event counter 00 can output a square wave with any selected frequency.

- Cycle: $(2 \times 2 \text{ to } 65536 \times 2) \times$ count clock cycle

(5) PPG output

16-bit timer/event counter 00 can output a square wave that have arbitrary cycle and pulse width.

- 2 < Pulse width < Cycle ≤ (FFFF + 1) H

(6) One-shot pulse output

16-bit timer/event counter 00 can output a one-shot pulse for which output pulse width can be set to any desired value.

6.2 Configuration of 16-Bit Timer/Event Counter 00

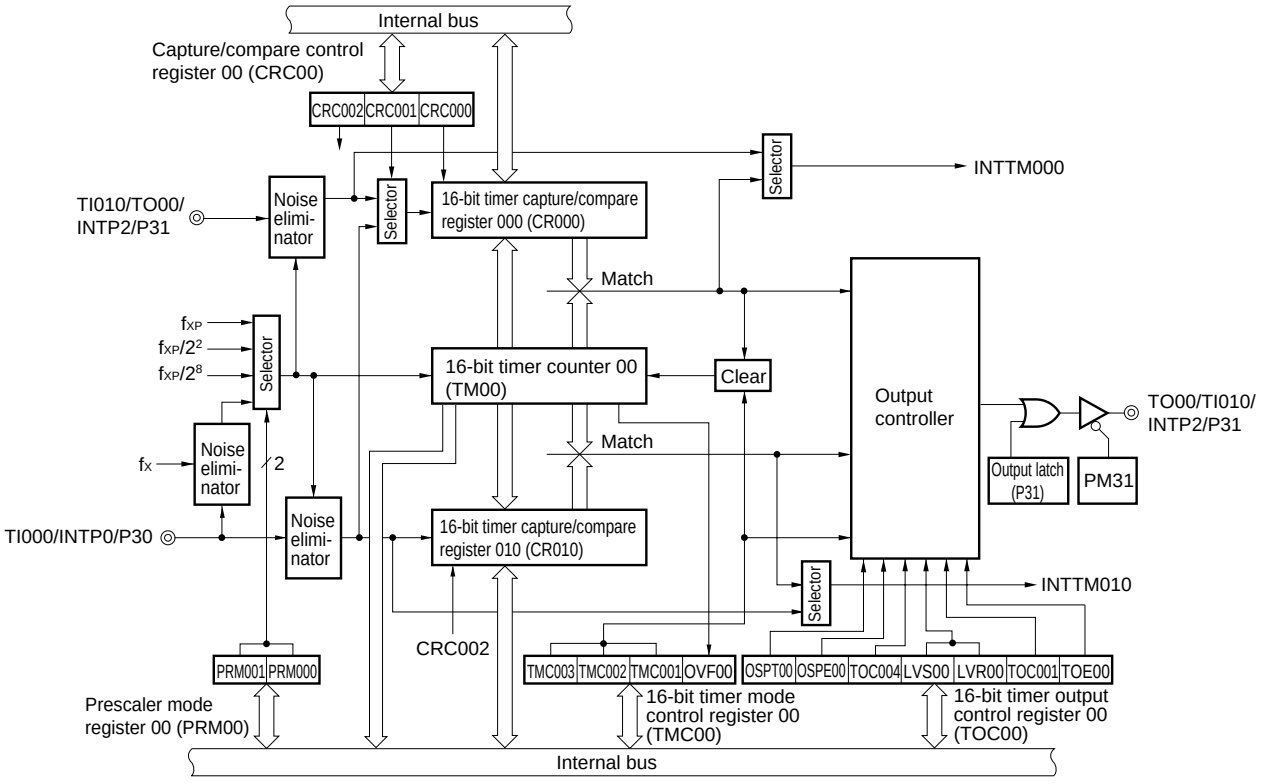
16-bit timer/event counter 00 consists of the following hardware.

Table 6-1. Configuration of 16-Bit Timer/Event Counter 00

Item	Configuration
Timer counter	16-bit timer counter 00 (TM00)
Register	16-bit timer capture/compare registers 000, 010 (CR000, CR010)
Timer input	TI000, TI010
Timer output	TO00, output controller
Control registers	16-bit timer mode control register 00 (TMC00) Capture/compare control register 00 (CRC00) 16-bit timer output control register 00 (TOC00) Prescaler mode register 00 (PRM00) Port mode register 3 (PM3) Port register 3 (P3)

Figures 6-1 shows a block diagram of these counters.

Figure 6-1. Block Diagram of 16-Bit Timer/Event Counter 00

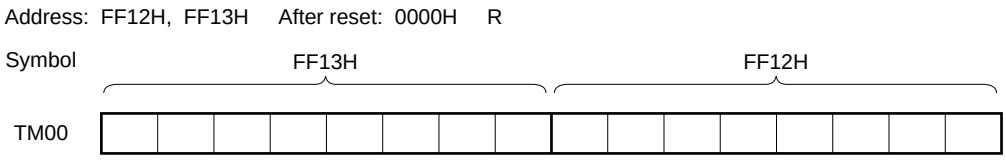


(1) 16-bit timer counter 00 (TM00)

TM00 is a 16-bit read-only register that counts count pulses.

The counter is incremented in synchronization with the rising edge of the count clock. If the count value is read during operation, input of the count clock is temporarily stopped, and the count value at that point is read.

Figure 6-2. Format of 16-Bit Timer Counter 00 (TM00)



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

<1> At reset input
<2> If TMC003 and TMC002 are cleared
<3> If the valid edge of TI000 is input in the clear & start mode entered by inputting the valid edge of TI000
<4> If TM00 and CR000 match in the clear & start mode entered on a match between TM00 and CR000
<5> If OSPT00 is set in the one-shot pulse output mode

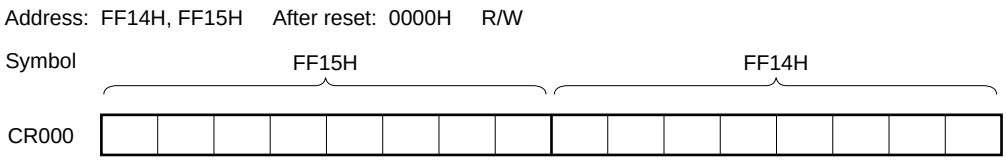
(2) 16-bit timer capture/compare register 000 (CR000)

CR000 is a 16-bit register which has the functions of both a capture register and a compare register. Whether it is used as a capture register or as a compare register is set by bit 0 (CRC000) of capture/compare control register 00 (CRC00).

CR000 is set by 16-bit memory manipulation instruction.

A reset clears CR000 to 0000H.

Figure 6-3. Format of 16-Bit Timer Capture/Compare Register 000 (CR000)



- **When CR000 is used as a compare register**

  The value set in CR000 is constantly compared with the 16-bit timer/counter 00 (TM00) count value, and an interrupt request (INTTM000) is generated if they match. It can also be used as the register that holds the interval time then TM00 is set to interval timer operation.

- **When CR000 is used as a capture register**

  It is possible to select the valid edge of the TI000 pin or the TI010 pin as the capture trigger. Setting of the TI000 or TI010 valid edge is performed by means of prescaler mode register 00 (PRM00) (refer to **Table 6-2**).

Table 6-2. CR000 Capture Trigger and Valid Edges of TI000 and TI010 Pins

(1) TI000 pin valid edge selected as capture trigger (CRC001 = 1, CRC000 = 1)

CR000 Capture Trigger	TI000 Pin Valid Edge		
		ES010	ES000
Falling edge	Rising edge	0	1
Rising edge	Falling edge	0	0
No capture operation	Both rising and falling edges	1	1

(2) TI010 pin valid edge selected as capture trigger (CRC001 = 0, CRC000 = 1)

CR000 Capture Trigger	TI010 Pin Valid Edge		
		ES110	ES100
Falling edge	Falling edge	0	0
Rising edge	Rising edge	0	1
Both rising and falling edges	Both rising and falling edges	1	1

Remarks 1. Setting ES010, ES000 = 1, 0 and ES110, ES100 = 1, 0 is prohibited.

2. ES010, ES000: Bits 5 and 4 of prescaler mode register 00 (PRM00)
ES110, ES100: Bits 7 and 6 of prescaler mode register 00 (PRM00)
CRC001, CRC000: Bits 1 and 0 of capture/compare control register 00 (CRC00)

Cautions 1. Set CR000 to a value other than 0000H in the clear & start mode entered on a match between TM00 and CR000. However, in the free-running mode and in the clear & start mode using the valid edge of TI000, if CR000 is set to 0000H, an interrupt request (INTTM000) is generated when CR000 changes from 0000H to 0001H following overflow (FFFFH).

2. If the new value of CR000 is less than the value of 16-bit timer counter 0 (TM00), TM00 continues counting, overflows, and then starts counting from 0 again. If the new value of CR000 is less than the old value, therefore, the timer must be reset to be restarted after the value of CR000 is changed.

3. When P31 is used as the input pin for the valid edge of TI010, it cannot be used as a timer output (TO00). Moreover, when P31 is used as TO00, it cannot be used as the input pin for the valid edge of TI010.

4. If the register read period and the input of the capture trigger conflict when CR000 is used as a capture register, the read data is undefined (the capture data itself is a normal value). Also, if the count stop input and the input of the capture trigger conflict, the capture trigger is undefined.

5. Changing the CR000 setting may cause a malfunction. To change the setting, refer to 6.5 Cautions Related to 16-Bit Timer/Event Counter 00 (11) Changing compare register during timer operation.

(3) 16-bit capture/compare register 010 (CR010)

CR010 is a 16-bit register which has the functions of both a capture register and a compare register. Whether it is used as a capture register or a compare register is set by bit 2 (CRC002) of capture/compare control register 00 (CRC00).

CR010 is set by 16-bit memory manipulation instruction.

Reset input clears CR010 to 0000H.

Figure 6-4. Format of 16-Bit Timer Capture/Compare Register 010 (CR010)

Address: FF16H, FF17H After reset: 0000H R/W

Symbol	FF17H								FF16H							
CR010																

- **When CR010 is used as a compare register**

  The value set in CR010 is constantly compared with the 16-bit timer counter 00 (TM00) count value, and an interrupt request (INTTM010) is generated if they match.

- **When CR010 is used as a capture register**

  It is possible to select the valid edge of the TI000 pin as the capture trigger. The TI000 valid edge is set by means of prescaler mode register 00 (PRM00) (refer to **Table 6-3**).

Table 6-3. CR010 Capture Trigger and Valid Edge of TI000 Pin (CRC002 = 1)

CR010 Capture Trigger	TI000 Pin Valid Edge		
		ES010	ES000
Falling edge	Falling edge	0	0
Rising edge	Rising edge	0	1
Both rising and falling edges	Both rising and falling edges	1	1

Remarks 1. Setting ES010, ES000 = 1, 0 is prohibited.

2. ES010, ES000: Bits 5 and 4 of prescaler mode register 00 (PRM00)

CRC002: Bit 2 of capture/compare control register 00 (CRC00)

Cautions 1. Set CR010 to other than 0000H in the clear & start mode entered on a match between TM00 and CR000. However, in the free-running mode and in the clear & start mode using the valid edge of the TI000 pin, if CR010 is set to 0000H, an interrupt request (INTTM010) is generated when CR010 changes from 0000H to 0001H following overflow (FFFFH).

2. If the register read period and the input of the capture trigger conflict when CR010 is used as a capture register, the read data is undefined (the capture data itself is a normal value). Also, if the count stop input and the input of the capture trigger conflict, the capture data is undefined.

3 Changing the CR010 setting during TM00 operation may cause a malfunction. To change the setting, refer to 6.5 Cautions Related to 16-Bit Timer/Event Counter 00 (11) Changing compare register during timer operation.

6.3 Registers to Control 16-Bit Timer/Event Counter 00

The following six types of registers are used to control 16-bit timer/event counter 00.

- 16-bit timer mode control register 00 (TMC00)
- Capture/compare control register 00 (CRC00)
- 16-bit timer output control register 00 (TOC00)
- Prescaler mode register 00 (PRM00)
- Port mode register 3 (PM3)
- Port register 3 (P3)

(1) 16-bit timer mode control register 00 (TMC00)

This register sets the 16-bit timer operating mode, the 16-bit timer counter 00 (TM00) clear mode, and output timing, and detects an overflow.

TMC00 is set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets the value of TMC00 to 00H.

Caution 16-bit timer counter 00 (TM00) starts operation at the moment TMC002 and TMC003 (operation stop mode) are set to a value other than 0, 0, respectively. Set TMC002 and TMC003 to 0, 0 to stop the operation.

Figure 6-5. Format of 16-Bit Timer Mode Control Register 00 (TMC00)

Address: FF60H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
TMC00	0	0	0	0	TMC003	TMC002	TMC001	OVF00

TMC003	TMC002	TMC001	Operating mode and clear mode selection	TO00 inversion timing selection	Interrupt request generation
0	0	0	Operation stop (TM00 cleared to 0)	No change	Not generated
0	0	1			
0	1	0	Free-running mode	Match between TM00 and CR000 or match between TM00 and CR010	Generated on match between TM00 and CR000, or match between TM00 and CR010
0	1	1		Match between TM00 and CR000, match between TM00 and CR010 or TI000 valid edge	
1	0	0	Clear & start occurs on valid edge of TI000 pin	–	
1	0	1			
1	1	0	Clear & start occurs on match between TM00 and CR000	Match between TM00 and CR000 or match between TM00 and CR010	
1	1	1		Match between TM00 and CR000, match between TM00 and CR010 or TI000 valid edge	

OVF00	Overflow detection of 16-bit timer counter 00 (TM00)
0	Overflow not detected
1	Overflow detected

Cautions 1. To write different data to TMC00, stop the timer operation before writing.

2. The timer operation must be stopped before writing to bits other than the OVF00 flag.

3. Set the valid edge of the TI000/INTP0/P30 pin with prescaler mode register 00 (PRM00).

4. If the clear & start mode entered on a match between TM00 and CR000, clear & start mode at the valid edge of the TI000 pin, or free-running mode is selected, when the set value of CR000 is FFFFH and the TM00 value changes from FFFFH to 0000H, the OVF00 flag is set to 1.

Remark TM00: 16-bit timer counter 00
CR000: 16-bit timer capture/compare register 000
CR010: 16-bit timer capture/compare register 010

(2) Capture/compare control register 00 (CRC00)

This register controls the operation of the 16-bit capture/compare registers (CR000, CR010).

CRC00 is set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets the value of CRC00 to 00H.

Figure 6-6. Format of Capture/Compare Control Register 00 (CRC00)

Address: FF62H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC00	0	0	0	0	0	CRC002	CRC001	CRC000

CRC002	CR010 operating mode selection
0	Operate as compare register
1	Operate as capture register

CRC001	CR000 capture trigger selection
0	Capture on valid edge of TI010 pin
1	Capture on valid edge of TI000 pin by reverse phase

CRC000	CR000 operating mode selection
0	Operate as compare register
1	Operate as capture register

Cautions 1. The timer operation must be stopped before setting CRC00.

2. When the clear & start mode entered on a match between TM00 and CR000 is selected by 16-bit timer mode control register 00 (TMC00), CR000 should not be specified as a capture register.

3. If both the rising and falling edges have been selected as the valid edges of the TI000 pin, capture is not performed.

4. To ensure the reliability of the capture operation, the capture trigger requires a pulse longer than two cycles of the count clock selected by prescaler mode register 00 (PRM00) (refer to Figure 6-17).

(3) 16-bit timer output control register 00 (TOC00)

This register controls the operation of the 16-bit timer/event counter output controller. It sets timer output F/F set/reset, output inversion enable/disable, 16-bit timer/event counter 00 timer output enable/disable, one-shot pulse output operation enable/disable, and output trigger of one-shot pulse by software.

TOC00 is set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets the value of TOC00 to 00H.

Figure 6-7. Format of 16-Bit Timer Output Control Register 00 (TOC00)

Address: FF63H After reset: 00H R/W

Symbol	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC00	0	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00

OSPT00	One-shot pulse output trigger control via software
0	No one-shot pulse trigger
1	One-shot pulse trigger

OSPE00	One-shot pulse output operation control
0	Successive pulse output mode
1	One-shot pulse output mode[Note]

TOC004	Timer output F/F control using match of CR010 and TM00
0	Disables inversion operation
1	Enables inversion operation

LVS00	LVR00	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TOC001	Timer output F/F control using match of CR000 and TM00
0	Disables inversion operation
1	Enables inversion operation

TOE00	Timer output control
0	Disables output (output fixed to level 0)
1	Enables output

Note The one-shot pulse output mode operates correctly only in the free-running mode and the mode in which clear & start occurs at the TI000 pin valid edge. In the mode in which clear & start occurs on a match between TM00 and CR000, one-shot pulse output is not possible because an overflow does not occur.

Cautions 1. Timer operation must be stopped before setting other than OSPT00.

2. If LVS00 and LVR00 are read, 0 is read.

3. OSPT00 is automatically cleared after data is set, so 0 is read.

4. Do not set OSPT00 to 1 other than in one-shot pulse output mode.

5. A write interval of two cycles or more of the count clock selected by prescaler mode register 00 (PRM00) is required to write to OSPT00 successively.

(4) Prescaler mode register 00 (PRM00)

This register is used to set the 16-bit timer counter 00 (TM00) count clock and the TI000, TI010 pin input valid edges.

PRM00 is set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets the value of PRM00 to 00H.

Figure 6-8. Format of Prescaler Mode Register 00 (PRM00)

Address: FF61H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM00	ES110	ES100	ES010	ES000	0	0	PRM001	PRM000

ES110	ES100	TI010 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES010	ES000	TI000 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM001	PRM000	Count clock selection
0	0	f_{XP} (10 MHz)
0	1	$f_{XP}/2^2$ (2.5 MHz)
1	0	$f_{XP}/2^8$ (39.06 kHz)
1	1	TI000 pin valid edge[Note]

Note The external clock requires a pulse longer than two cycles of the internal count clock (f_{XP}).

Cautions 1. Always set data to PRM00 after stopping the timer operation.

2. If the valid edge of the TI000 pin is to be set as the count clock, do not set the clear/start mode and the capture trigger at the valid edge of the TI000 pin.

3. If the TI000 or TI010 pin is high level immediately after system reset, the rising edge is immediately detected after the rising edge or both the rising and falling edges are set as the valid edge(s) of the TI000 pin or TI010 pin to enable the operation of 16-bit timer counter 00 (TM00). Be careful when pulling up the TI000 pin or the TI010 pin. However, when re-enabling operation after the operation has been stopped once, the rising edge is not detected.

4. When using P31 as the input pin of the TI010 pin valid edge, it cannot be used as a timer output (TO00). When using P31 as the TO00 pin, it cannot be used as the input pin of the TI010 pin valid edge.

Remarks 1. f_{XP}: Oscillation frequency of clock supplied to peripheral hardware

2. ( ): f_{XP} = 10 MHz

(5) Port mode register 3 (PM3)

This register sets port 3 input/output in 1-bit units.

When using the P31/TO00/TI010/INTP2 pin for timer output, set PM31 and the output latch of P31 to 0.

When using the P30/TI000/INTP0 and P31/TO00/TI010/INTP2 pins as a timer input, set PM30 and PM31 to 1. At this time, the output latches of P30 and P31 can be either 0 or 1.

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

Reset input sets the value of PM3 to FFH.

Figure 6-9. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	1	PM31	PM30

PM3n	P3n pin I/O mode selection (n = 0 or 1)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

6.4 Operation of 16-Bit Timer/Event Counter 00

6.4.1 Interval timer operation

Setting 16-bit timer mode control register 00 (TMC00) and capture/compare control register 00 (CRC00) as shown in Figure 6-10 allows operation as an interval timer.

Setting

The basic operation setting procedure is as follows.

<1> Set the CRC00 register (see **Figure 6-10** for the set value).
<2> Set any value to the CR000 register.
<3> Set the count clock by using the PRM00 register.
<4> Set the TMC00 register to start the operation (see **Figure 6-10** for the set value).

Caution Changing the CR000 setting during TM00 operation may cause a malfunction. To change the setting, refer to 6.5 Cautions Related to 16-Bit Timer/Event Counter 00 (11) Changing compare register during timer operation.

Remark For how to enable the INTTM000 interrupt, see **CHAPTER 12 INTERRUPT FUNCTIONS**.

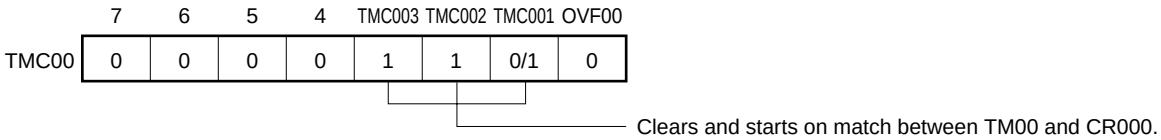
Interrupt requests are generated repeatedly using the count value set in 16-bit timer capture/compare register 000 (CR000) beforehand as the interval.

When the count value of 16-bit timer counter 00 (TM00) matches the value set to CR000, counting continues with the TM00 value cleared to 0 and the interrupt request signal (INTTM000) is generated.

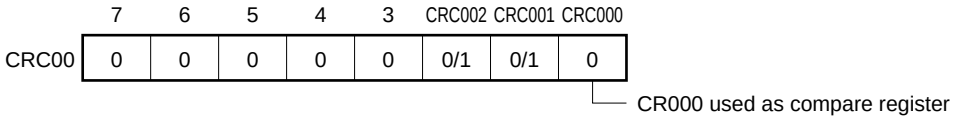
The count clock of the 16-bit timer/event counter can be selected using bits 0 and 1 (PRM000, PRM001) of prescaler mode register 00 (PRM00).

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

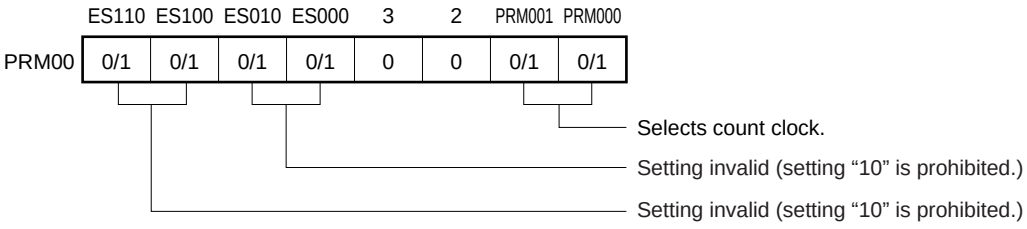
(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)

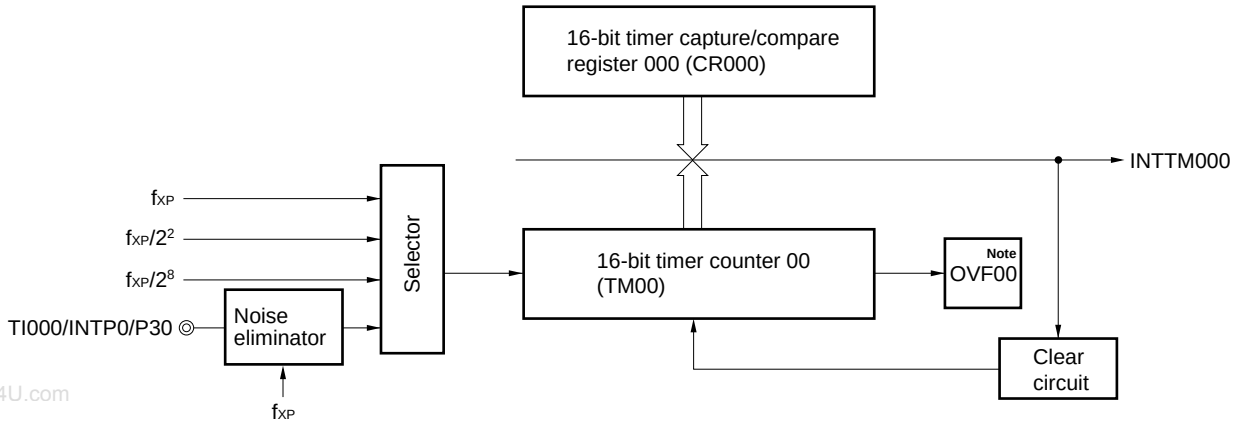


(c) Prescaler mode register 00 (PRM00)



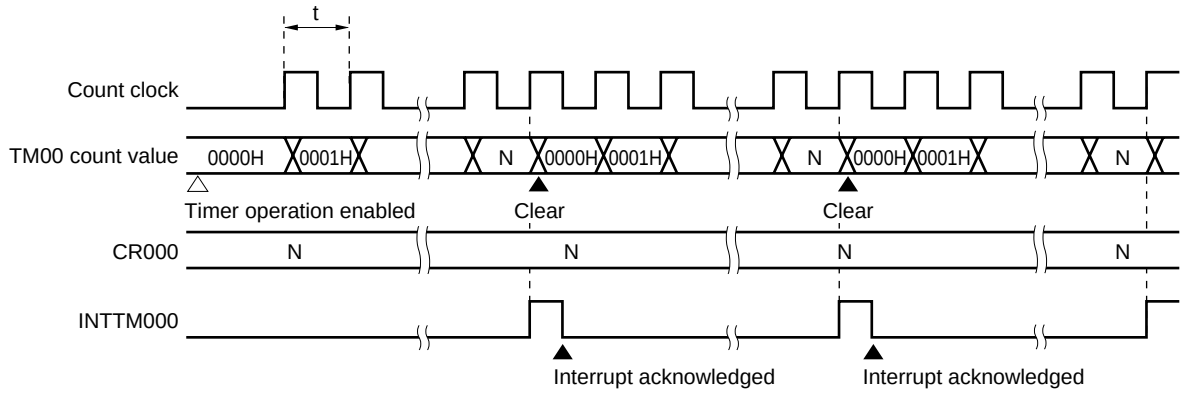
Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the interval timer. See the description of the respective control registers for details.

Figure 6-11. Interval Timer Configuration Diagram



Note OVF00 is set to 1 only when 16-bit timer capture/compare register 000 is set to FFFFH.

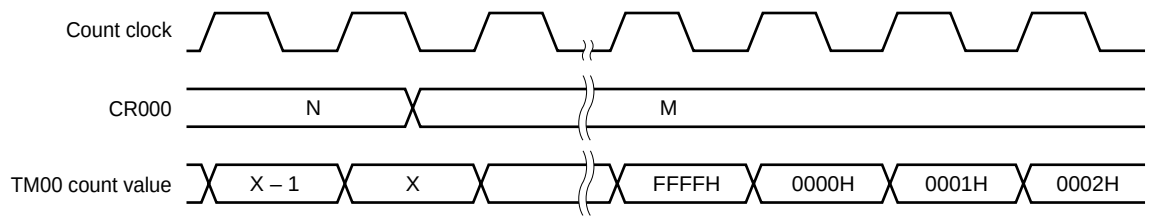
Figure 6-12. Timing of Interval Timer Operation



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

When the compare register is changed during timer count operation, if the value after 16-bit timer capture/compare register 000 (CR000) is changed is smaller than that of 16-bit timer counter 00 (TM00), TM00 continues counting, overflows and then restarts counting from 0. Thus, if the value (M) after the CR000 change is smaller than that (N) before the change, it is necessary to restart the timer after changing CR000.

Figure 6-13. Timing After Change of Compare Register During Timer Count Operation



Remark N > X > M

6.4.2 External event counter operation

Setting

The basic operation setting procedure is as follows.

<1> Set the CRC00 register (see **Figure 6-14** for the set value).
<2> Set the count clock by using the PRM00 register.
<3> Set any value to the CR000 register (0000H cannot be set).
<4> Set the TMC00 register to start the operation (see **Figure 6-14** for the set value).

Remarks 1. For the setting of the TI000 pin, see **6.3 (5) Port mode register 3 (PM3)**.
2. For how to enable the INTTM000 interrupt, see **CHAPTER 12 INTERRUPT FUNCTIONS**.

The external event counter counts the number of external clock pulses to be input to the TI000 pin with using 16-bit timer counter 00 (TM00).

TM00 is incremented each time the valid edge specified by prescaler mode register 00 (PRM00) is input.

When the TM00 count value matches the 16-bit timer capture/compare register 000 (CR000) value, TM00 is cleared to 0 and the interrupt request signal (INTTM000) is generated.

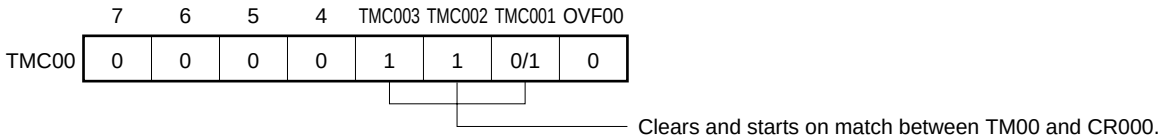
Input a value other than 0000H to CR000. (A count operation with a pulse cannot be carried out.)

The rising edge, the falling edge, or both edges can be selected using bits 4 and 5 (ES000 and ES010) of prescaler mode register 00 (PRM00).

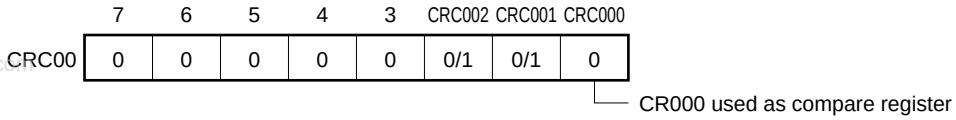
Because an operation is carried out only when the valid edge of the TI000 pin is detected twice after sampling with the internal clock (f_{XP}), noise with a short pulse width can be removed.

Figure 6-14. Control Register Settings in External Event Counter Mode (with Rising Edge Specified)

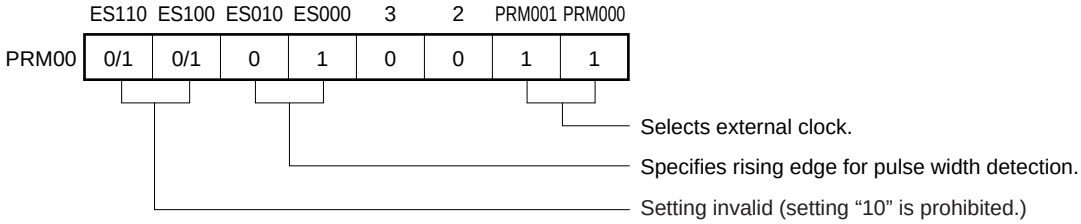
(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)

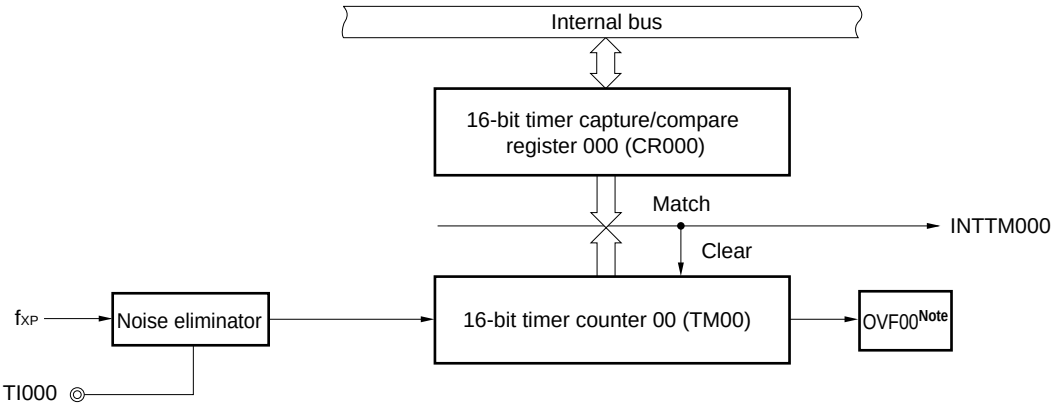


(c) Prescaler mode register 00 (PRM00)



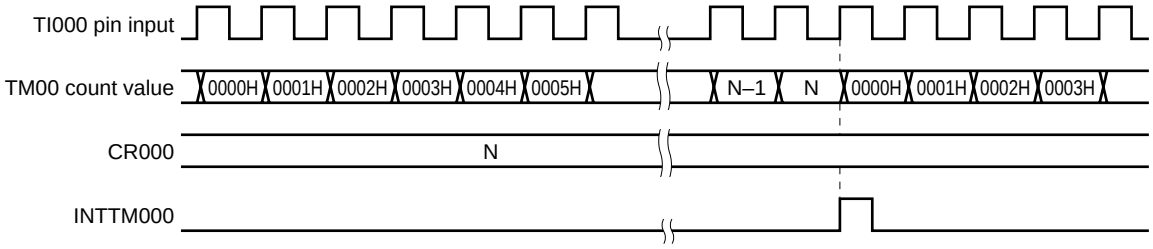
Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the external event counter. See the description of the respective control registers for details.

Figure 6-15. External Event Counter Configuration Diagram



Note OVF00 is 1 only when 16-bit timer capture/compare register 000 is set to FFFFH.

Figure 6-16. External Event Counter Operation Timing (with Rising Edge Specified)



Caution When reading the external event counter count value, TM00 should be read.

6.4.3 Pulse width measurement operations

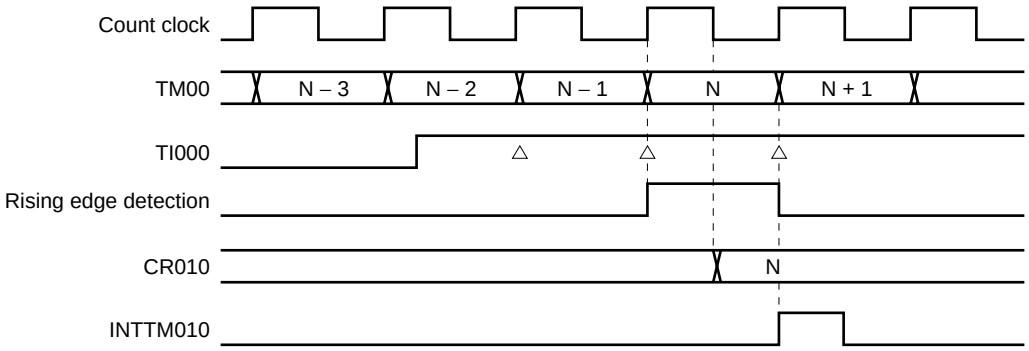
It is possible to measure the pulse width of the signals input to the TI000 pin and TI010 pin using 16-bit timer counter 00 (TM00).

There are two measurement methods: measuring with TM00 used in free-running mode, and measuring by restarting the timer in synchronization with the edge of the signal input to the TI000 pin.

When an interrupt occurs, read the valid value of the capture register, check the overflow flag, and then calculate the necessary pulse width. Clear the overflow flag after checking it.

The capture operation is not performed until the signal pulse width is sampled in the count clock cycle selected by prescaler mode register 00 (PRM00) and the valid level of the TI000 or TI010 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 6-17. CR010 Capture Operation with Rising Edge Specified



Setting

The basic operation setting procedure is as follows.

<1> Set the CRC00 register (see **Figures 6-18**, **6-21**, **6-23**, and **6-25** for the set value).
<2> Set the count clock by using the PRM00 register.
<3> Set the TMC00 register to start the operation (see **Figures 6-18**, **6-21**, **6-23**, and **6-25** for the set value).

Caution To use two capture registers, set the TI000 and TI010 pins.

Remarks 1. For the setting of the TI000 (or TI010) pin, see **6.3 (5) Port mode register 3 (PM3)**.

2. For how to enable the INTTM000 (or INTTM010) interrupt, see **CHAPTER 12 INTERRUPT FUNCTIONS**.

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

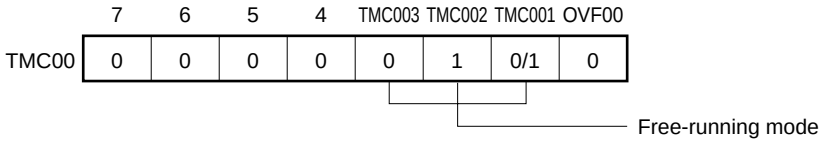
When 16-bit timer counter 00 (TM00) is operated in free-running mode, and the edge specified by prescaler mode register 00 (PRM00) is input to the TI000 pin, the value of TM00 is taken into 16-bit timer capture/compare register 010 (CR010) and an external interrupt request signal (INTTM010) is set.

Specify both the rising and falling edges by using bits 4 and 5 (ES000 and ES010) of PRM00.

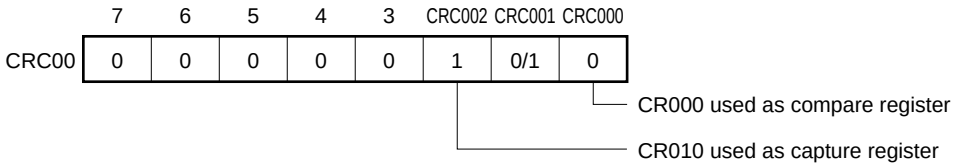
Sampling is performed using the count clock selected by PRM00, and a capture operation is only performed when a valid level of the TI000 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 6-18. Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register (When TI000 and CR010 Are Used)

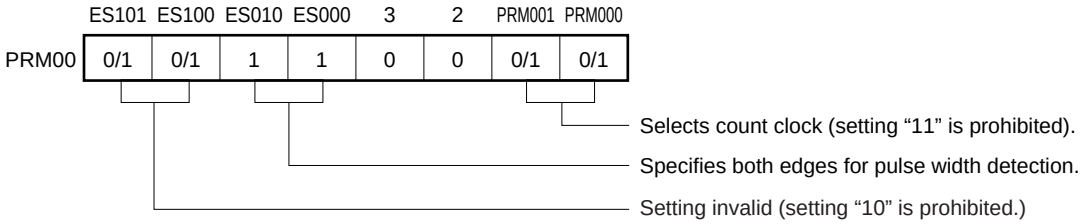
(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)



(c) Prescaler mode register 00 (PRM00)



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

Figure 6-19. Configuration Diagram for Pulse Width Measurement by Free-Running Counter

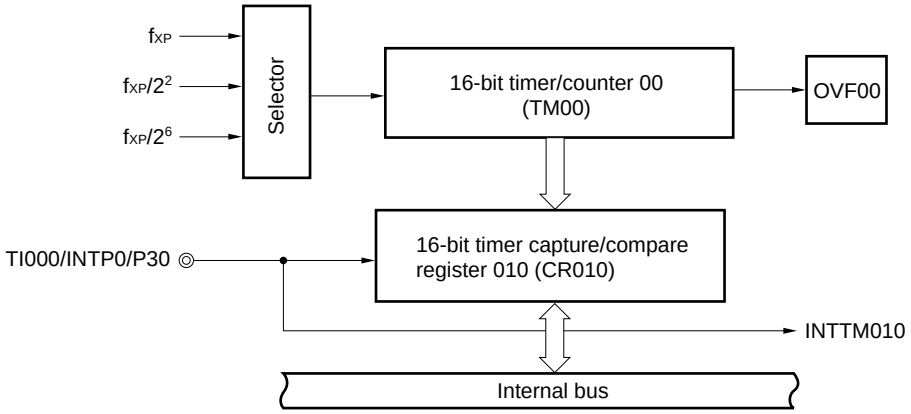
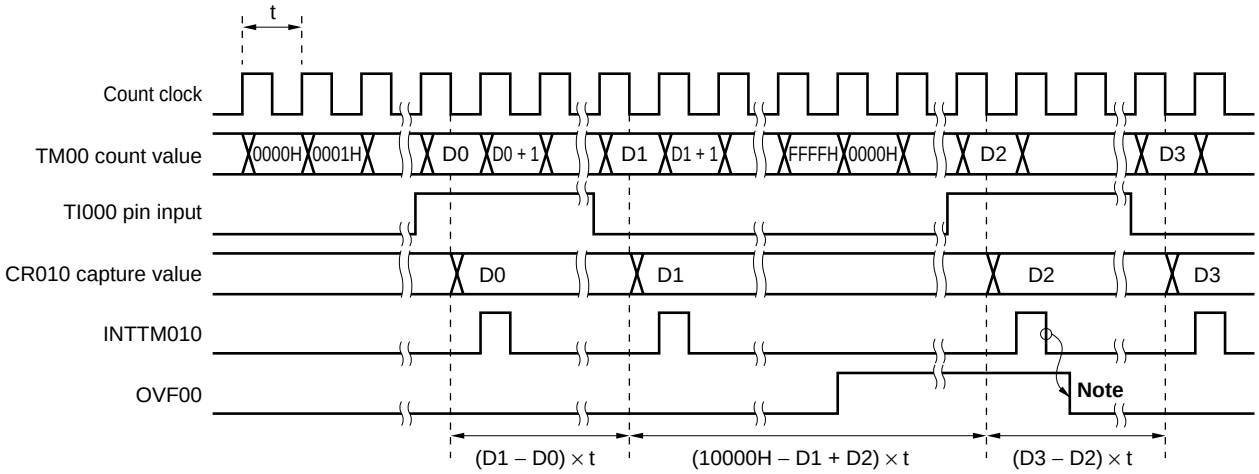


Figure 6-20. Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)



Note OVF00 must be cleared by software.

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

When 16-bit timer counter 00 (TM00) is operated in free-running mode, it is possible to simultaneously measure the pulse widths of the two signals input to the TI000 pin and the TI010 pin.

When the edge specified by bits 4 and 5 (ES000 and ES010) of prescaler mode register 00 (PRM00) is input to the TI000 pin, the value of TM00 is taken into 16-bit timer capture/compare register 010 (CR010) and an interrupt request signal (INTTM010) is set.

Also, when the edge specified by bits 6 and 7 (ES100 and ES110) of PRM00 is input to the TI010 pin, the value of TM00 is taken into 16-bit timer capture/compare register 000 (CR000) and an interrupt request signal (INTTM000) is set.

Specify both the rising and falling edges as the edges of the TI000 and TI010 pins, by using bits 4 and 5 (ES000 and ES010) and bits 6 and 7 (ES100 and ES110) of PRM00.

Sampling is performed using the count clock cycle selected by prescaler mode register 00 (PRM00), and a capture operation is only performed when a valid level of the TI000 or TI010 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 6-21. Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter

(a) 16-bit timer mode control register 00 (TMC00)

	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	0	1	0/1	0

Free-running mode

(b) Capture/compare control register 00 (CRC00)

	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	1	0	1

CR000 used as capture register

Captures valid edge of TI010 pin to CR000.

CR010 used as capture register

(c) Prescaler mode register 00 (PRM00)

	ES110	ES100	ES010	ES000	3	2	PRM001	PRM000
PRM00	1	1	1	1	0	0	0/1	0/1

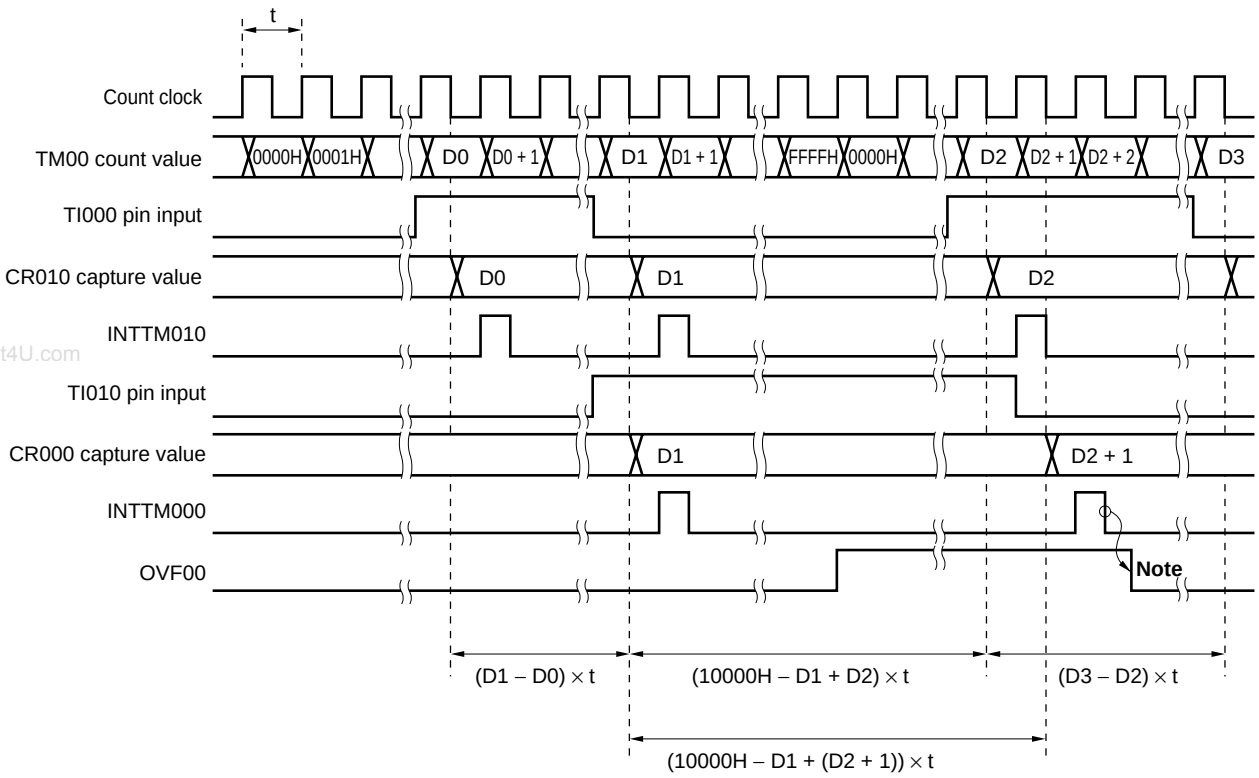
Selects count clock (setting “11” is prohibited).

Specifies both edges for pulse width detection.

Specifies both edges for pulse width detection.

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

Figure 6-22. Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified)



Note OVF00 must be cleared by software.

(3) Pulse width measurement with free-running counter and two capture registers

When 16-bit timer counter 00 (TM00) is operated in free-running mode, it is possible to measure the pulse width of the signal input to the TI000 pin.

When the rising or falling edge specified by bits 4 and 5 (ES000 and ES010) of prescaler mode register 00 (PRM00) is input to the TI000 pin, the value of TM00 is taken into 16-bit timer capture/compare register 010 (CR010) and an interrupt request signal (INTTM010) is set.

Also, when the inverse edge to that of the capture operation is input into CR010, the value of TM00 is taken into 16-bit timer capture/compare register 000 (CR000).

Sampling is performed using the count clock cycle selected by prescaler mode register 00 (PRM00), and a capture operation is only performed when a valid level of the TI000 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 6-23. Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers (with Rising Edge Specified)

(a) 16-bit timer mode control register 00 (TMC00)

	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	0	1	0/1	0

Free-running mode

(b) Capture/compare control register 00 (CRC00)

	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	1	1	1

CR000 used as capture register

Captures to CR000 at inverse edge to valid edge of TI000.

CR010 used as capture register

(c) Prescaler mode register 00 (PRM00)

	ES110	ES100	ES010	ES000	3	2	PRM001	PRM000
PRM00	0/1	0/1	0	1	0	0	0/1	0/1

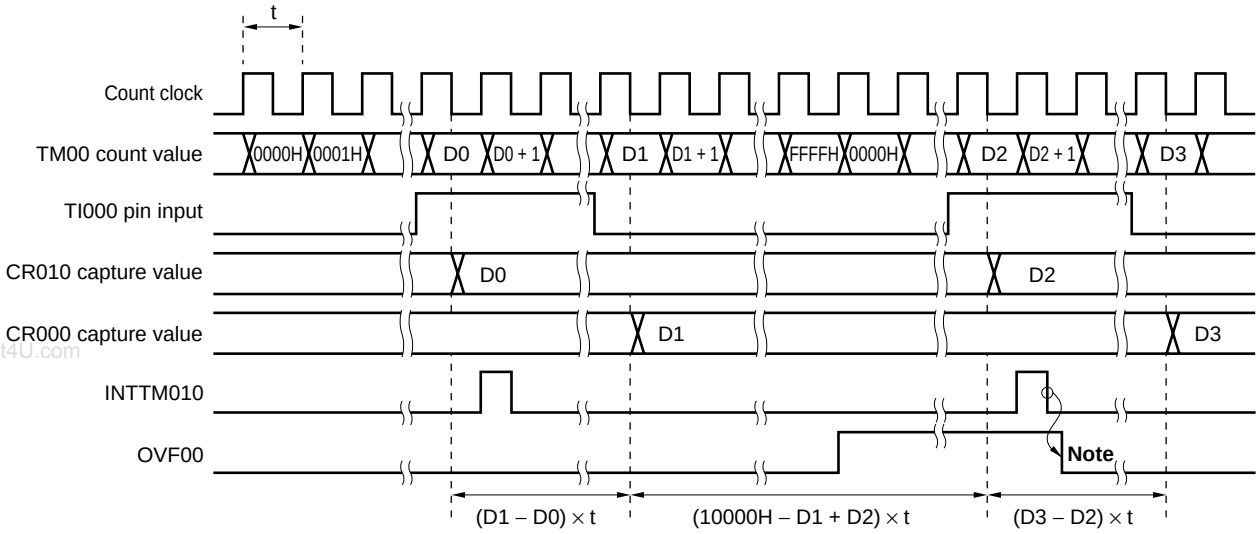
Selects count clock (setting “11” is prohibited).

Specifies rising edge for pulse width detection.

Setting invalid (setting “10” is prohibited.)

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

Figure 6-24. Timing of Pulse Width Measurement Operation by Free-Running Counter and Two Capture Registers (with Rising Edge Specified)



Note OVF00 must be cleared by software.

(4) Pulse width measurement by means of restart

When input of a valid edge to the TI000 pin is detected, the count value of 16-bit timer/counter 00 (TM00) is taken into 16-bit timer capture/compare register 010 (CR010), and then the pulse width of the signal input to the TI000 pin is measured by clearing TM00 and restarting the count.

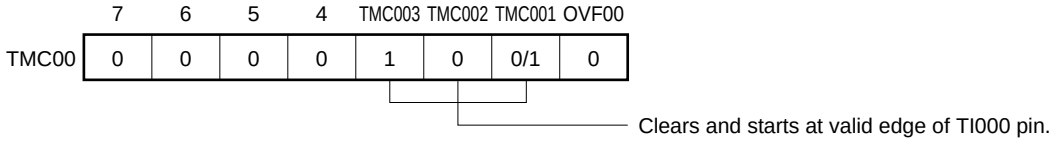
The edge specification can be selected from two types, rising or falling edges, by bits 4 and 5 (ES000 and ES010) of prescaler mode register 00 (PRM00)

Sampling is performed at the interval selected by prescaler mode register 00 (PRM00) and a capture operation is only performed when a valid level of the TI000 pin is detected twice, thus eliminating noise with a short pulse width.

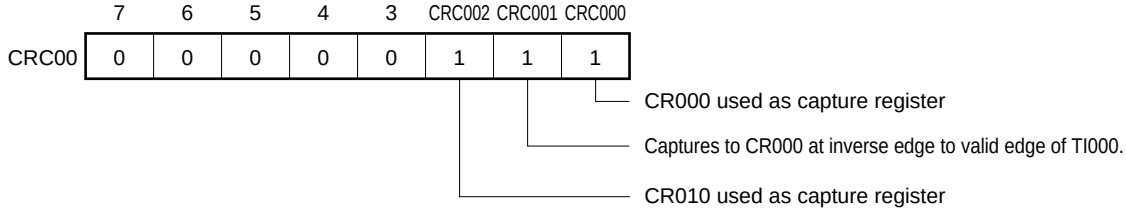
Caution If the valid edge of TI000 is specified to be both the rising and falling edges, 16-bit timer capture/compare register 000 (CR000) cannot perform the capture operation.

Figure 6-25. Control Register Settings for Pulse Width Measurement by Means of Restart (with Rising Edge Specified)

(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)



(c) Prescaler mode register 00 (PRM00)

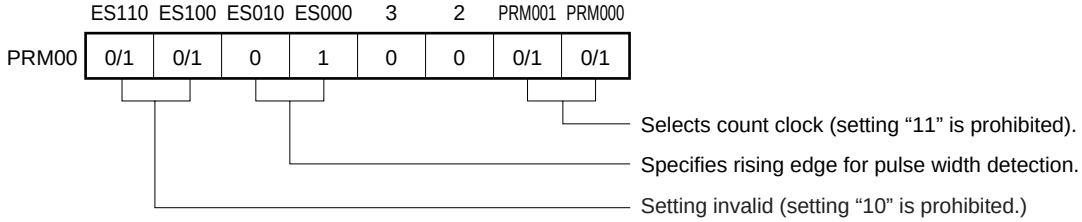
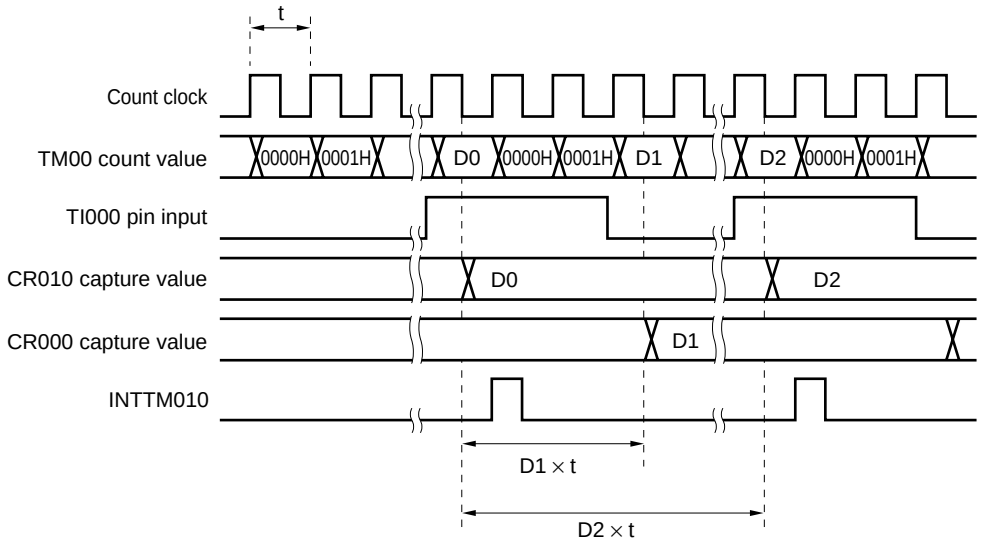


Figure 6-26. Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)



6.4.4 Square-wave output operation

Setting

The basic operation setting procedure is as follows.

<1> Set the count clock by using the PRM00 register.
<2> Set the CRC00 register (see **Figure 6-27** for the set value).
<3> Set the TOC00 register (see **Figure 6-27** for the set value).
<4> Set any value to the CR000 register (0000H cannot be set).
<5> Set the TMC00 register to start the operation (see **Figure 6-27** for the set value).

Caution Changing the CR000 setting during TM00 operation may cause a malfunction. To change the setting, refer to 6.5 Cautions Related to 16-Bit Timer/Event Counter 00 (11) Changing compare register during timer operation.

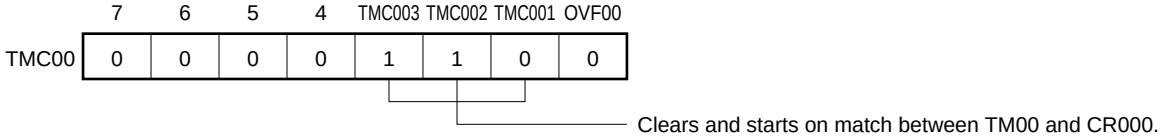
Remarks 1. For the setting of the TO00 pin, see **6.3 (5) Port mode register 3 (PM3)**.
2. For how to enable the INTTM000 interrupt, see **CHAPTER 12 INTERRUPT FUNCTIONS**.

A square wave with any selected frequency can be output at intervals determined by the count value preset to 16-bit timer capture/compare register 000 (CR000).

The TO00 pin output status is reversed at intervals determined by the count value preset to CR000 + 1 by setting bit 0 (TOE00) and bit 1 (TOC001) of 16-bit timer output control register 00 (TOC00) to 1. This enables a square wave with any selected frequency to be output.

Figure 6-27. Control Register Settings in Square-Wave Output Mode (1/2)

(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)

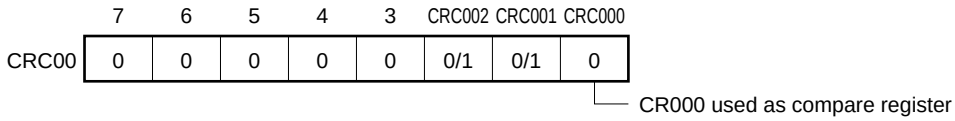
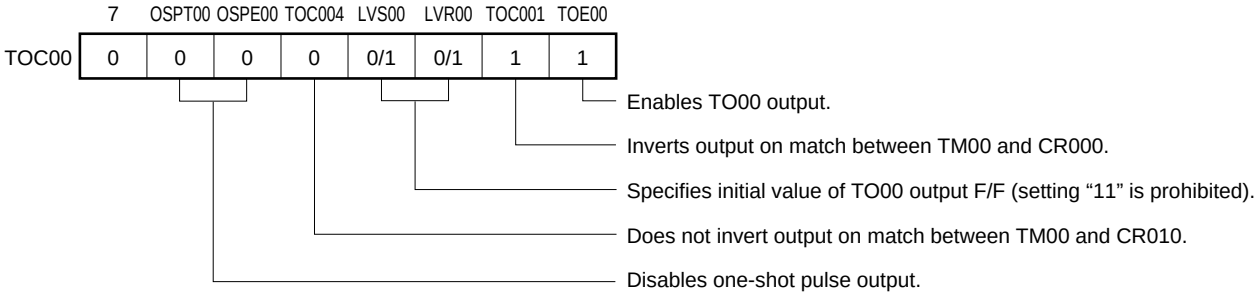


Figure 6-27. Control Register Settings in Square-Wave Output Mode (2/2)

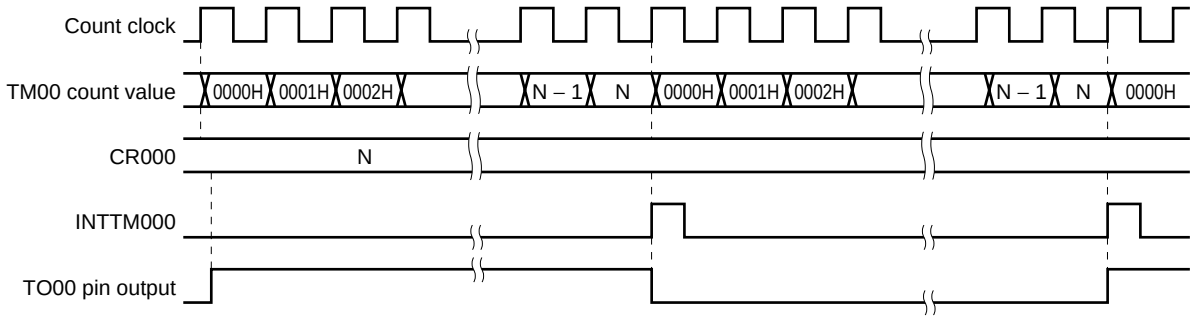
(c) 16-bit timer output control register 00 (TOC00)



(d) Prescaler mode register 00 (PRM00)

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with square-wave output. See the description of the respective control registers for details.

Figure 6-28. Square-Wave Output Operation Timing



6.4.5 PPG output operations

Setting 16-bit timer mode control register 00 (TMC00) and capture/compare control register 00 (CRC00) as shown in Figure 6-29 allows operation as PPG (Programmable Pulse Generator) output.

Setting

The basic operation setting procedure is as follows.

<1> Set the CRC00 register (see **Figure 6-29** for the set value).
<2> Set any value to the CR000 register as the cycle.
<3> Set any value to the CR010 register as the duty factor.
<4> Set the TOC00 register (see **Figure 6-29** for the set value).
<5> Set the count clock by using the PRM00 register.
<6> Set the TMC00 register to start the operation (see **Figure 6-29** for the set value).

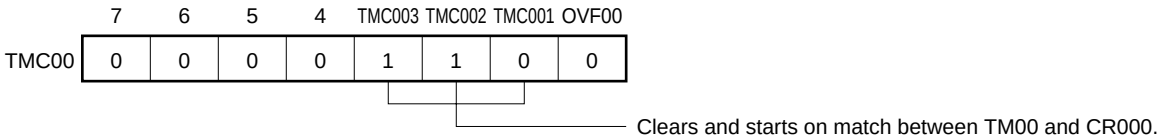
Caution Changing the CRC0n0 setting during TM00 operation may cause a malfunction. To change the setting, refer to 6.5 Cautions Related to 16-Bit Timer/Event Counter 00 (11) Changing compare register during timer operation.

Remarks 1. For the setting of the TO00 pin, see **6.3 (5) Port mode register 3 (PM3)**.
2. For how to enable the INTTM000 interrupt, see **CHAPTER 12 INTERRUPT FUNCTIONS**.
3. n = 0 or 1

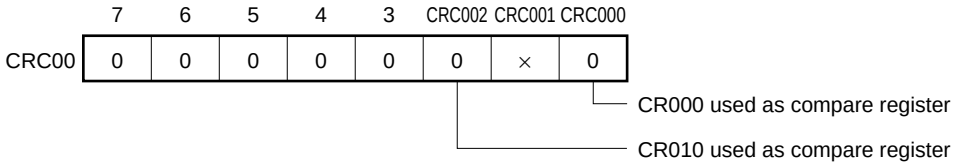
In the PPG output operation, rectangular waves are output from the TO00 pin with the pulse width and the cycle that correspond to the count values preset in 16-bit timer capture/compare register 010 (CR010) and in 16-bit timer capture/compare register 000 (CR000), respectively.

Figure 6-29. Control Register Settings for PPG Output Operation

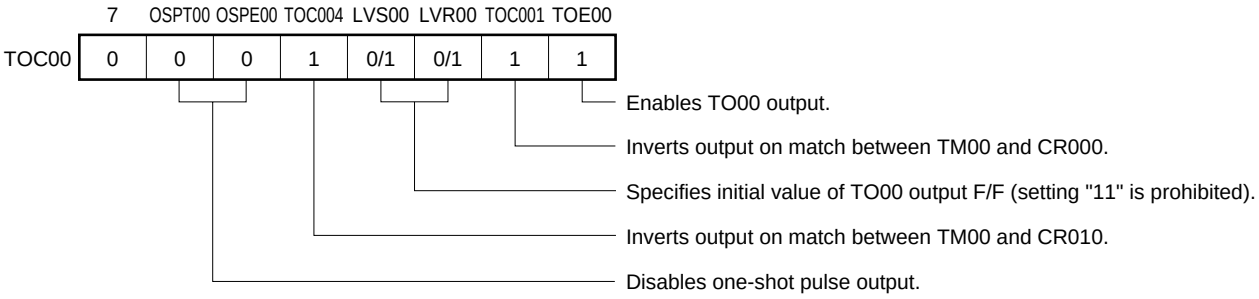
(a) 16-bit timer mode control register 00 (TMC00)



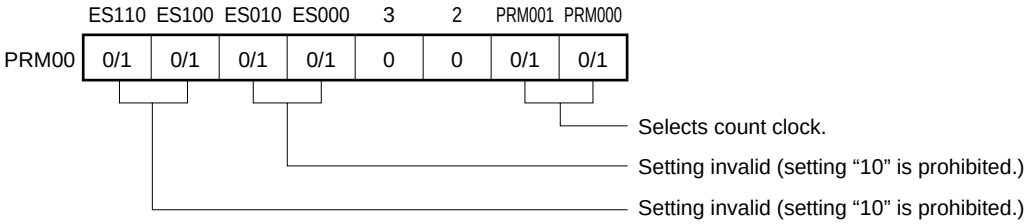
(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)



(d) Prescaler mode register 00 (PRM00)



Cautions 1. Values in the following range should be set in CR000 and CR010:
0000H ≤ CR010 < CR000 ≤ FFFFH (Setting CR000 to 0000H is prohibited.)
2. The cycle of the pulse generated through PPG output (CR000 setting value + 1) has a duty of (CR010 setting value + 1)/(CR000 setting value + 1).

Remark ×: Don’t care

Figure 6-30. Configuration Diagram of PPG Output

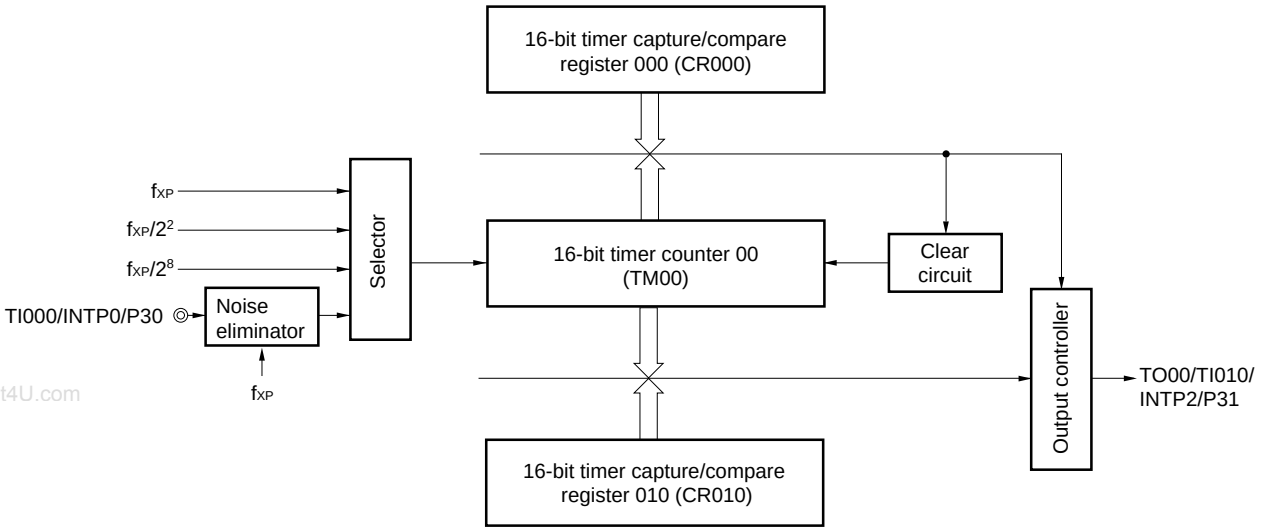
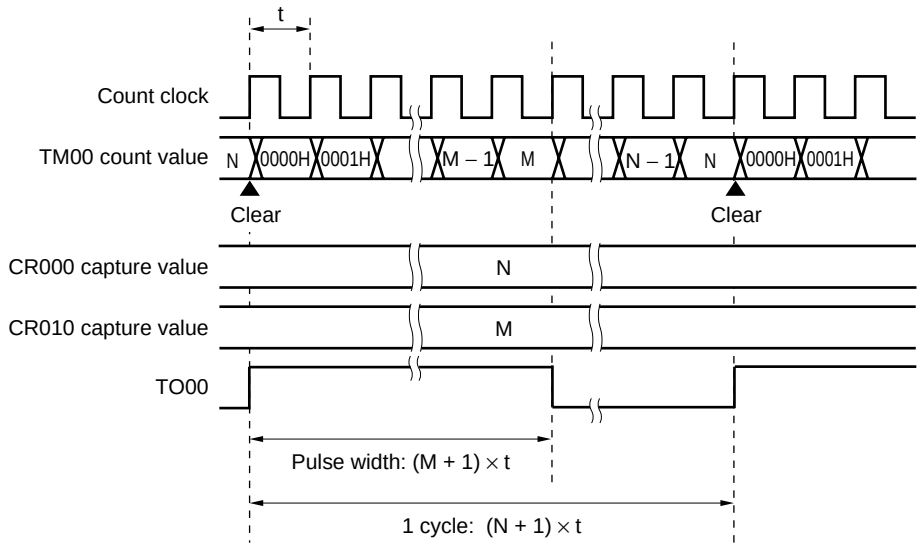


Figure 6-31. PPG Output Operation Timing



Remark 0000H ≤ M < N ≤ FFFFH

6.4.6 One-shot pulse output operation

16-bit timer/event counter 00 can output a one-shot pulse in synchronization with a software trigger or an external trigger (TI000 pin input).

Setting

The basic operation setting procedure is as follows.

<1> Set the count clock by using the PRM00 register.
<2> Set the CRC00 register (see **Figures 6-32** and **6-34** for the set value).
<3> Set the TOC00 register (see **Figures 6-32** and **6-34** for the set value).
<4> Set any value to the CR000 and CR010 registers (0000H cannot be set).
<5> Set the TMC00 register to start the operation (see **Figures 6-32** and **6-34** for the set value).

Remarks 1. For the setting of the TO00 pin, see **6.3 (5) Port mode register 3 (PM3)**.

2. For how to enable the INTTM000 (if necessary, INTTM010) interrupt, see **CHAPTER 12 INTERRUPT FUNCTIONS**.

(1) One-shot pulse output with software trigger

A one-shot pulse can be output from the TO00 pin by setting 16-bit timer mode control register 00 (TMC00), capture/compare control register 00 (CRC00), and 16-bit timer output control register 00 (TOC00) as shown in Figure 6-32, and by setting bit 6 (OSPT00) of the TOC00 register to 1 by software.

By setting the OSPT00 bit to 1, 16-bit timer/event counter 00 is cleared and started, and its output becomes active at the count value (N) set in advance to 16-bit timer capture/compare register 010 (CR010). After that, the output becomes inactive at the count value (M) set in advance to 16-bit timer capture/compare register 000 (CR000)[Note].

Even after the one-shot pulse has been output, the TM00 register continues its operation. To stop the TM00 register, the TMC003 and TMC002 bits of the TMC00 register must be cleared to 00.

Note The case where N < M is described here. When N > M, the output becomes active with the CR000 register and inactive with the CR010 register. Do not set N to M.

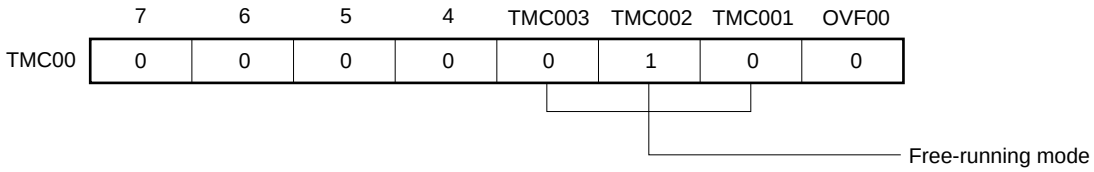
Cautions 1. Do not set the OSPT00 bit while the one-shot pulse is being output. To output the one-shot pulse again, wait until the current one-shot pulse output is completed.

2. When using the one-shot pulse output of 16-bit timer/event counter 00 with a software trigger, do not change the level of the TI000 pin or its alternate-function port pin.

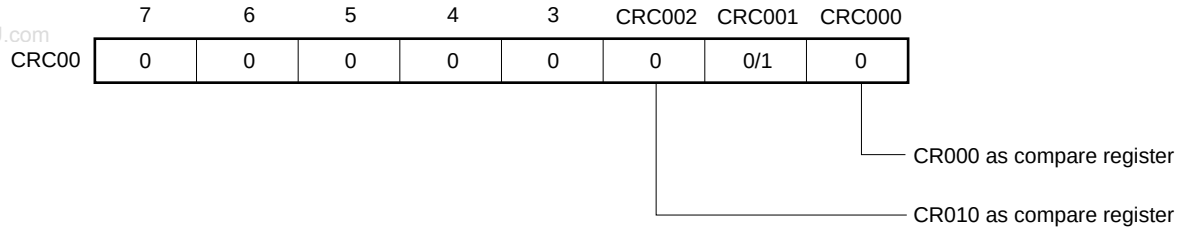
Because the external trigger is valid even in this case, the timer is cleared and started even at the level of the TI000 pin or its alternate-function port pin, resulting in the output of a pulse at an undesired timing.

Figure 6-32. Control Register Settings for One-Shot Pulse Output with Software Trigger

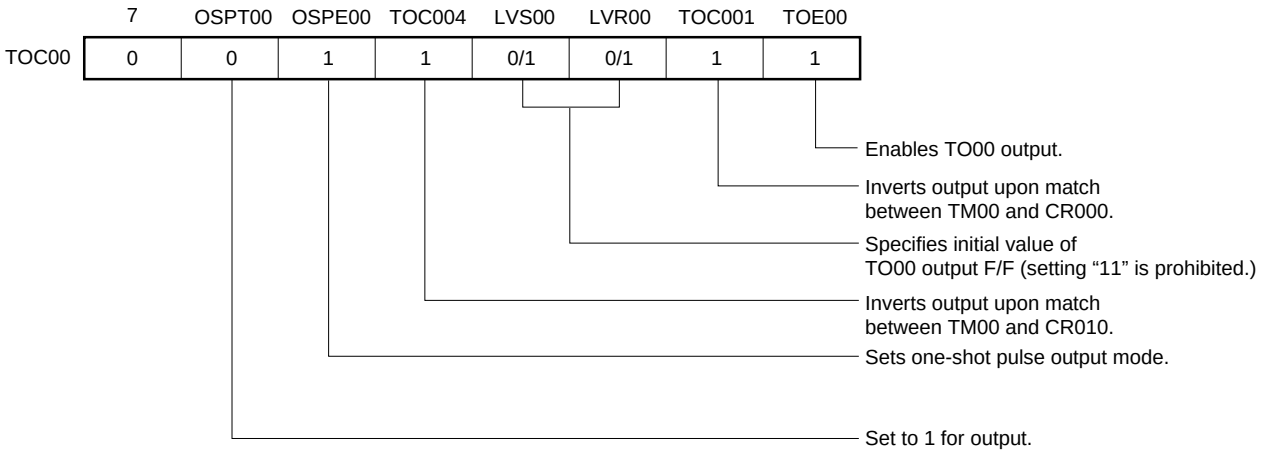
(a) 16-bit timer mode control register 00 (TMC00)



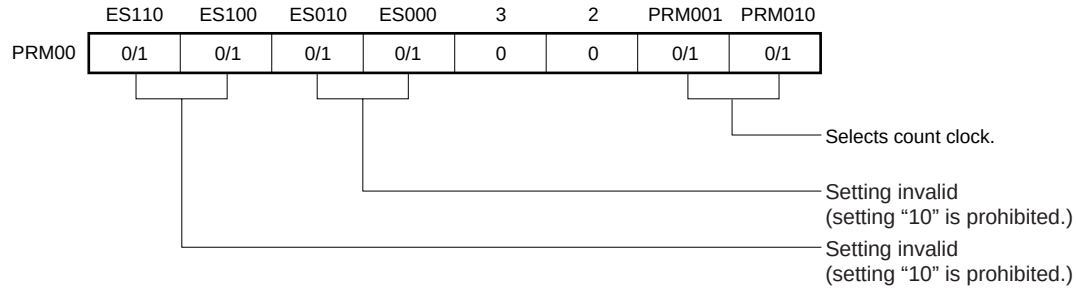
(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)

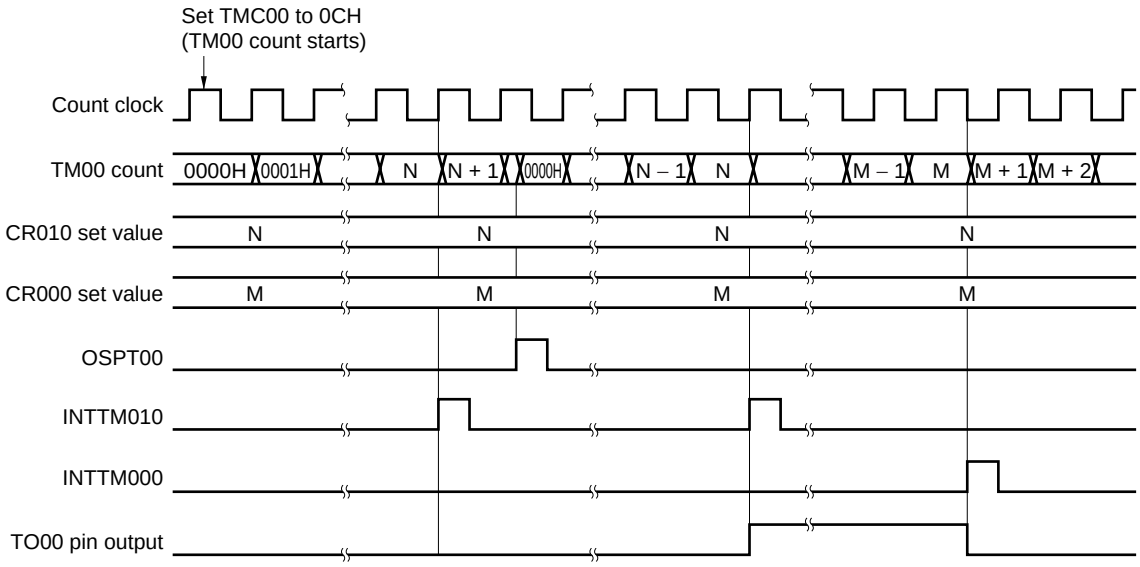


(d) Prescaler mode register 00 (PRM00)



Caution Do not set 0000H to the CR000 and CR010 registers.

Figure 6-33. Timing of One-Shot Pulse Output Operation with Software Trigger



Caution 16-bit timer counter 00 starts operating as soon as a value other than 00 (operation stop mode) is set to the TMC003 and TMC002 bits.

Remark N < M

(2) One-shot pulse output with external trigger

A one-shot pulse can be output from the TO00 pin by setting 16-bit timer mode control register 00 (TMC00), capture/compare control register 00 (CRC00), and 16-bit timer output control register 00 (TOC00) as shown in Figure 6-34, and by using the valid edge of the TI000 pin as an external trigger.

The valid edge of the TI000 pin is specified by bits 4 and 5 (ES000, ES010) of prescaler mode register 00 (PRM00). The rising, falling, or both the rising and falling edges can be specified.

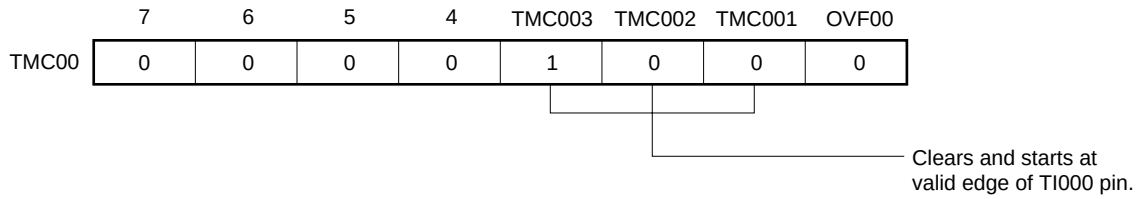
When the valid edge of the TI000 pin is detected, the 16-bit timer/event counter is cleared and started, and the output becomes active at the count value set in advance to 16-bit timer capture/compare register 010 (CR010). After that, the output becomes inactive at the count value set in advance to 16-bit timer capture/compare register 000 (CR000)[Note].

Note The case where N < M is described here. When N > M, the output becomes active with the CR000 register and inactive with the CR010 register. Do not set N to M.

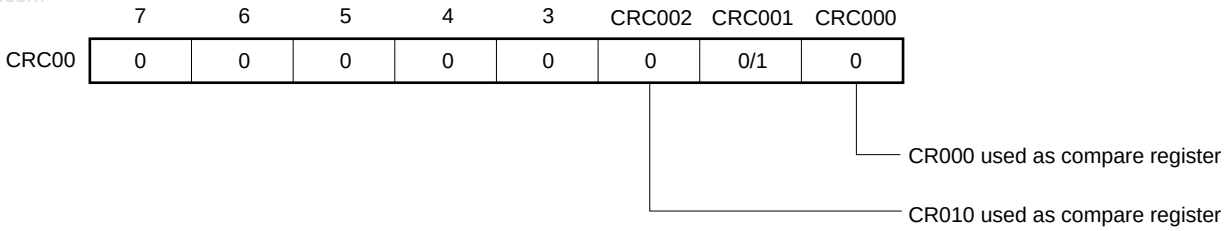
Caution Even if the external trigger is generated again while the one-shot pulse is output, it is ignored.

Figure 6-34. Control Register Settings for One-Shot Pulse Output with External Trigger (with Rising Edge Specified)

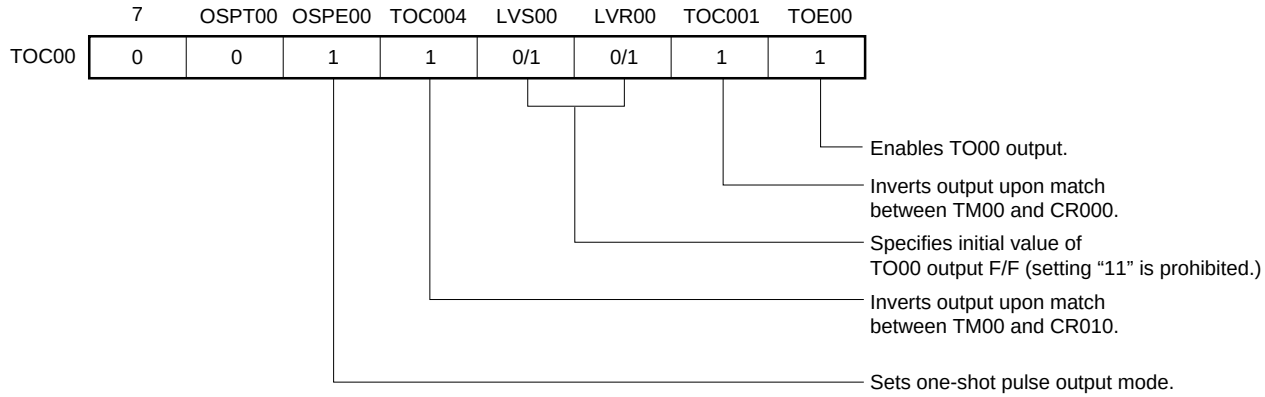
(a) 16-bit timer mode control register 00 (TMC00)



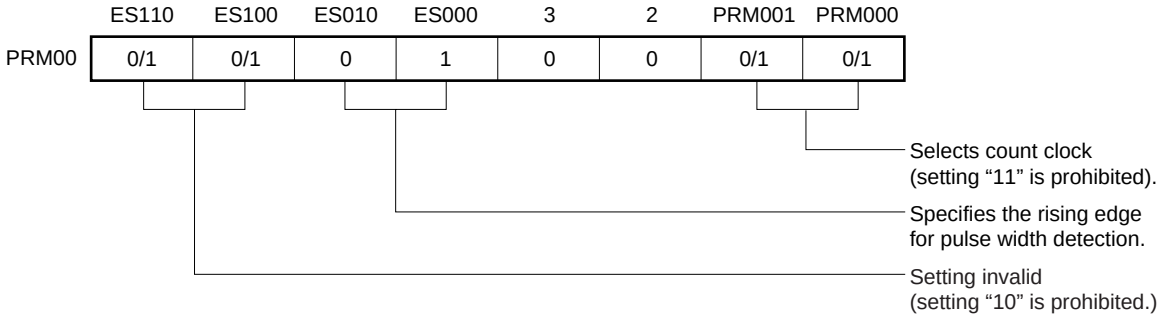
(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)

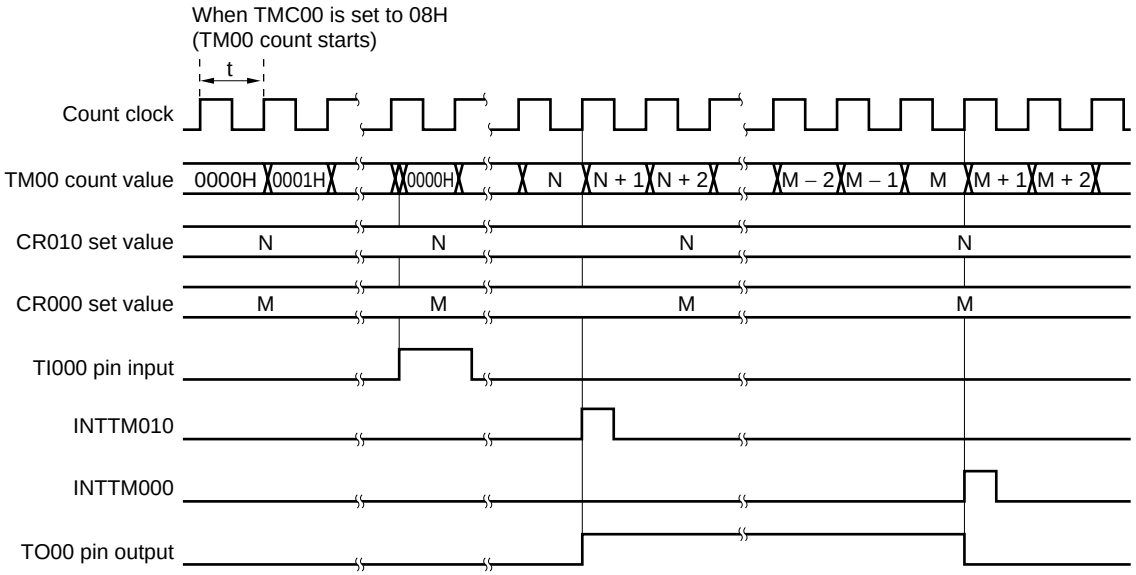


(d) Prescaler mode register 00 (PRM00)



Caution Do not set 0000H to the CR000 and CR010 registers.

Figure 6-35. Timing of One-Shot Pulse Output Operation with External Trigger (with Rising Edge Specified)



Caution 16-bit timer counter 00 starts operating as soon as a value other than 00 (operation stop mode) is set to the TMC002 and TMC003 bits.

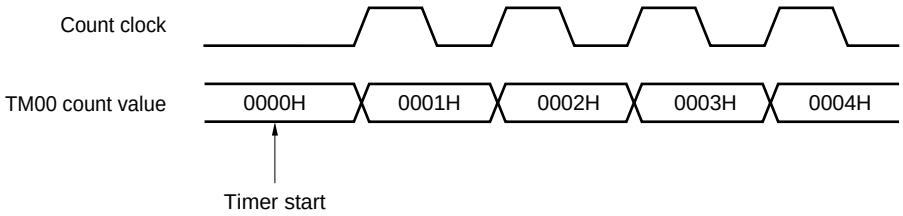
Remark N < M

6.5 Cautions Related to 16-Bit Timer/Event Counter 00

(1) Timer start errors

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because 16-bit timer counter 00 (TM00) is started asynchronously to the count clock.

Figure 6-36. Start Timing of 16-Bit Timer Counter 00 (TM00)



(2) 16-Bit timer capture/compare register setting

Set 16-bit timer capture/compare register 000 (CR000) to other than 0000H in the clear & start mode entered on match between TM00 and CR000. This means a 1-pulse count operation cannot be performed when this register is used as an external event counter.

(3) Capture register data retention

The values of 16-bit timer capture/compare registers 000, 010 (CR000, CR010) after 16-bit timer/event counter 00 has stopped are not guaranteed.

(4) Valid edge setting

Set the valid edge of the TI000 pin after setting bits 2 and 3 (TMC002 and TMC003) of 16-bit timer mode control register 00 (TMC00) to 0, 0, respectively, and then stopping the timer operation. The valid edge is set by bits 4 and 5 (ES000 and ES010) of prescaler mode register 00 (PRM00).

(5) Re-triggering one-shot pulse

(a) One-shot pulse output by software

When a one-shot pulse is output, do not set the OSPT00 bit to 1. Do not output the one-shot pulse again until INTTM000, which occurs upon a match with the CR000 register, or INTTM010, which occurs upon a match with the CR010 register, occurs.

(b) One-shot pulse output with external trigger

If the external trigger occurs again while a one-shot pulse is output, it is ignored.

(c) One-shot pulse output function

When using the one-shot pulse output of 16-bit timer/event counter 00 with a software trigger, do not change the level of the TI000 pin or its alternate function port pin.

Because the external trigger is valid even in this case, the timer is cleared and started even at the level of the TI000 pin or its alternate function port pin, resulting in the output of a pulse at an undesired timing.

(6) Operation of OVF00 flag

<1> The OVF00 flag is also set to 1 in the following case.
Either of the clear & start mode entered on a match between TM00 and CR000, clear & start at the valid edge of the TI000 pin, or free-running mode is selected.

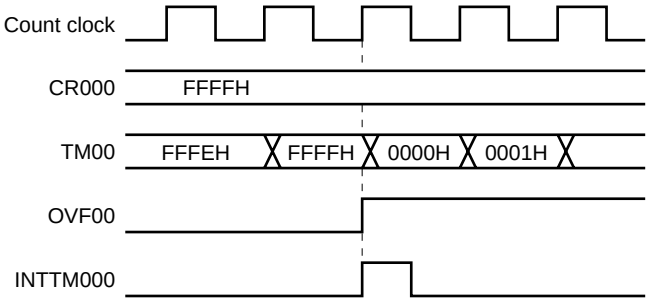
↓

CR000 is set to FFFFH.

↓

When TM00 is counted up from FFFFH to 0000H.

Figure 6-37. Operation Timing of OVF00 Flag



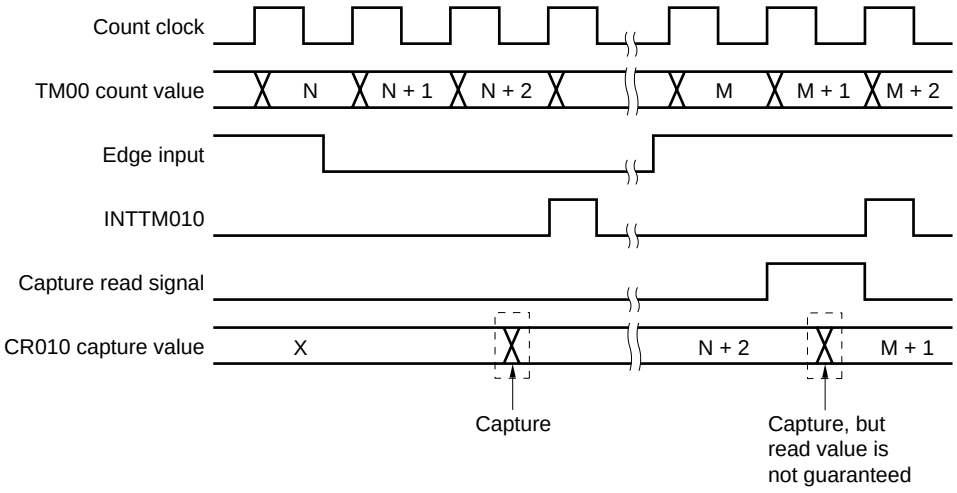
<2> Even if the OVF00 flag is cleared before the next count clock is counted (before TM00 becomes 0001H) after the occurrence of a TM00 overflow, the OVF00 flag is reset newly and clear is disabled.

(7) Conflicting operations

<1> When the 16-bit timer capture/compare register (CR000/CR010) is used as a compare register, if the write period and the match timing of 16-bit timer counter 00 (TM00) conflict, match determination is not successfully done. Do not perform a write operation of CR000/CR010 near the match timing.
When performing a write operation, refer to **(11) Changing compare register during timer operation**.

<2> If the read period and capture trigger input conflict when CR000/CR010 is used as a capture register, capture trigger input has priority. The data read from CR000/CR010 is undefined.

Figure 6-38. Capture Register Data Retention Timing



(8) Timer operation

<1> Even if 16-bit timer counter 00 (TM00) is read, the value is not captured by 16-bit timer capture/compare register 010 (CR010).

<2> Regardless of the CPU's operation mode, when the timer stops, the signals input to pins TI000/TI010 are not acknowledged.

<3> One-shot pulse output normally operates only in the free-running mode or in the clear & start mode at the valid edge of the TI000 pin. Because an overflow does not occur in the clear & start mode on a match between TM00 and CR000, one-shot pulse output is not possible.

(9) Capture operation

<1> If the TI000 pin is specified as the valid edge of the count clock, a capture operation by the capture register specified as the trigger for the TI000 pin is not possible.

<2> If both the rising and falling edges are selected as the valid edges of the TI000 pin, capture is not performed.

<3> To ensure the reliability of the capture operation, the capture trigger requires a pulse longer than two cycles of the count clock selected by prescaler mode register 00 (PRM00).

<4> The capture operation is performed at the fall of the count clock. A interrupt request input (INTTM0n0), however, occurs at the rise of the next count clock.

Remark n = 0, 1

(10) Compare operation

The capture operation may not be performed for CR000/CR010 set in compare mode even if a capture trigger is input.

(11) Changing compare register during timer operation

<1> When changing CR0n0 around the timing of a match between 16-bit timer counter 00 (TM00) and 16-bit timer capture/compare register 0n0 (CR0n0) during timer counting, the change timing may conflict with the timing of the match, so the operation is not guaranteed in such cases. To change CR0n0 during timer counting, follow the procedure below using an INTTM000 interrupt.

<Changing cycle (CR000)>

1. Disable the timer output inversion operation at the match between TM00 and CR000 (TOC001 = 0).
2. Disable the INTTM000 interrupt (TMMK000 = 1).
3. Rewrite CR000.
4. Wait for 1 cycle of the TM00 count clock.
5. Enable the timer output inversion operation at the match between TM00 and CR000 (TOC001 = 1).
6. Clear the interrupt request flag of INTTM000 (TMIF000 = 0).
7. Enable the INTTM000 interrupt (TMMK000 = 0).

<Changing duty (CR010)>

1. Disable the timer output inversion operation at the match between TM00 and CR010 (TOC004 = 0).
2. Disable the INTTM000 interrupt (TMMK000 = 1).
3. Rewrite CR010.
4. Wait for 1 cycle of the TM00 count clock.
5. Enable the timer output inversion operation at the match between TM00 and CR010 (TOC004 = 1).
6. Clear the interrupt request flag of INTTM000 (TMIF000 = 0).
7. Enable the INTTM000 interrupt (TMMK000 = 0).

While interrupts and timer output inversion are disabled (1 to 4 above), timer counting is continued. If the value to be set in CR0n0 is small, the value of TM00 may exceed CR0n0. Therefore, set the value, considering the time lapse of the timer clock and CPU after an INTTM000 interrupt has been generated.

Remark n = 0 or 1

<2> If CR010 is changed during timer counting without performing processing <1> above, the value in CR010 may be rewritten twice or more, causing an inversion of the output level of the TO00 pin at each rewrite.

(12) Edge detection

<1> If the TI000 pin or the TI010 pin is high level immediately after system reset and the rising edge or both the rising and falling edges are specified as the valid edge for the TI000 pin or TI010 pin to enable 16-bit timer counter 00 (TM00) operation, a rising edge is detected immediately. Be careful when pulling up the TI000 pin or the TI010 pin. However, the rising edge is not detected at restart after the operation has been stopped once.

<2> The sampling clock used to remove noise differs when a TI000 valid edge is used as the count clock and when it is used as a capture trigger. In the former case, the count clock is f_{XP}, and in the latter case the count clock is selected by prescaler mode register 00 (PRM00). The capture operation is not performed until the valid edge is sampled and the valid level is detected twice, thus eliminating, noise with a short pulse width.

(13) STOP mode or system clock stop mode setting

Except when TI000, TI010 input is selected, stop the timer operation before setting STOP mode or system clock stop mode; otherwise the timer may malfunction when the system clock starts.

CHAPTER 7 8-BIT TIMER 80

7.1 Function of 8-Bit Timer 80

8-bit timer 80 has an 8-bit interval timer function and generates an interrupt at intervals specified in advance.

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

	Minimum Interval Time	Maximum Interval Time	Resolution
f_{XP} = 8.0 MHz	$2^6/f_{XP}$ (8 μs)	$2^{14}/f_{XP}$ (2.05 ms)	$2^6/f_{XP}$ (8 μs)
	$2^8/f_{XP}$ (32 μs)	$2^{16}/f_{XP}$ (8.19 ms)	$2^8/f_{XP}$ (32 μs)
	$2^{10}/f_{XP}$ (128 μs)	$2^{18}/f_{XP}$ (32.7 ms)	$2^{10}/f_{XP}$ (128 μs)
	$2^{16}/f_{XP}$ (8.19 ms)	$2^{24}/f_{XP}$ (2.01 s)	$2^{16}/f_{XP}$ (8.19 ms)
f_{XP} = 10.0 MHz	$2^6/f_{XP}$ (6.4 μs)	$2^{14}/f_{XP}$ (1.64 ms)	$2^6/f_{XP}$ (6.4 μs)
	$2^8/f_{XP}$ (25.6 μs)	$2^{16}/f_{XP}$ (6.55 ms)	$2^8/f_{XP}$ (25.6 μs)
	$2^{10}/f_{XP}$ (102 μs)	$2^{18}/f_{XP}$ (26.2 ms)	$2^{10}/f_{XP}$ (102 μs)
	$2^{16}/f_{XP}$ (6.55 ms)	$2^{24}/f_{XP}$ (1.68 s)	$2^{16}/f_{XP}$ (6.55 ms)

Remark f_{XP}: Oscillation frequency of clock to peripheral hardware

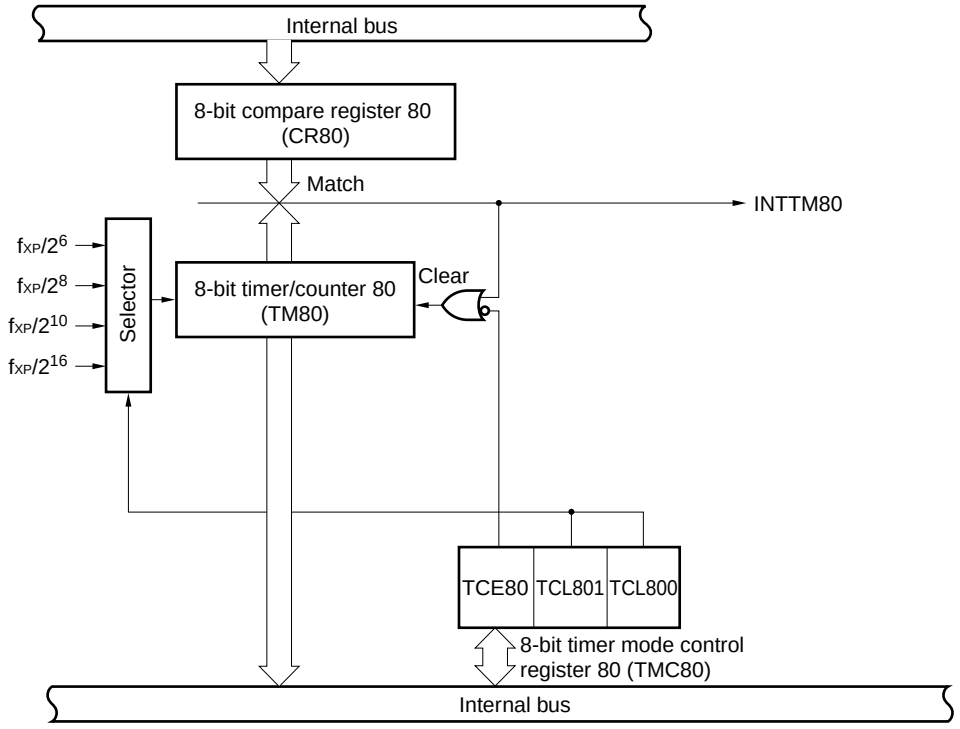
7.2 Configuration of 8-Bit Timer 80

8-bit timer 80 consists of the following hardware.

Table 7-2. Configuration of 8-Bit Timer 80

Item	Configuration
Timer counter	8-bit timer counter 80 (TM80)
Register	8-bit compare register 80 (CR80)
Control register	8-bit timer mode control register 80 (TMC80)

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



Remark f_{XP}: Oscillation frequency of clock to peripheral hardware

(1) 8-bit compare register 80 (CR80)

This 8-bit register always compares its set value with the count value of 8-bit timer/counter 80 (TM80). It generates an interrupt request signal (INTTM80) if the two values match.

CR80 is set by using an 8-bit memory manipulation instruction. A value of 00H to FFH can be set to this register.

Reset input makes the contents of this register undefined.

Figure 7-2. Format of 8-Bit Compare Register 80 (CR80)

Address: FFCDH After reset: Undefined W

Symbol	7	6	5	4	3	2	1	0
CR80								

Caution When changing the value of CR80, be sure to stop the timer operation. If the value of CR80 is changed with the timer operation enabled, a match interrupt request signal may be generated immediately.

(2) 8-bit timer/counter 80 (TM80)

This 8-bit register counts the count pulses.

The value of TM80 can be read by using an 8-bit memory manipulation instruction.

Reset input clears TM80 to 00H.

Figure 7-3. Format of 8-Bit Timer Counter 80 (TM80)

Address: FFCEH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
TM80								

7.3 Register Controlling 8-Bit Timer 80

8-bit timer 80 is controlled by 8-bit timer mode control register 80 (TMC80).

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

This register is used to enable or stop the operation of 8-bit timer/counter 80 (TM80), and to set the count clock of TM80.

This register is set by using a 1-bit or 8-bit memory manipulation instruction.

Reset input clears TMC80 to 00H.

Figure 7-4. Format of 8-Bit Timer Mode Control Register 80 (TMC80)

Address: FFCCH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
TMC80	TCE80	0	0	0	0	TCL801	TCL800	0

TCE80	Control of operation of TM80
0	Stop operation (clear TM80 to 00H).
1	Enable operation.

TCL801	TCL800	Selection of count clock of 8-bit timer 80		
			f_{XP} = 8.0 MHz	f_{XP} = 10.0 MHz
0	0	$f_{XP}/2^6$	125 kHz	156.3 kHz
0	1	$f_{XP}/2^8$	31.25 kHz	39.06 kHz
1	0	$f_{XP}/2^{10}$	7.81 kHz	9.77 kHz
1	1	$f_{XP}/2^{16}$	0.12 kHz	0.15 kHz

Caution Be sure to set TMC80 after stopping the timer operation.

Remark f_{XP}: Oscillation frequency of clock to peripheral hardware

7.4 Operation of 8-Bit Timer 80

7.4.1 Operation as interval timer

When 8-bit timer 80 operates as an interval timer, it can repeatedly generate an interrupt at intervals specified by the count value set in advance to 8-bit compare register 80 (CR80).

To use 8-bit timer 80 as an interval timer, make the following setting.

<1> Disable the operation of 8-bit timer/counter 80 (clear TCE80 (bit 7 of 8-bit timer mode control register 80 (TMC80)) to 0).
<2> Set the count clock of 8-bit timer 80 (refer to **Tables 7-3** and **7-4**).
<3> Set the count value to CR80.
<4> Enable the operation of TM80 (set TCE80 to 1).

When the count value of 8-bit timer/counter 80 (TM80) matches the set value of CR80, the value of TM80 is cleared to 00H and counting is continued. At the same time, an interrupt request signal (INTTM80) is generated.

Tables 7-3 and 7-4 show the interval time, and Figure 7-5 shows the timing of the interval timer operation.

Cautions 1. When changing the value of CR80, be sure to stop the timer operation. If the value of CR80 is changed with the timer operation enabled, a match interrupt request signal may be generated immediately.

2. If the count clock of TMC80 is set and the operation of TM80 is enabled at the same time by using an 8-bit memory manipulation instruction, the error of one cycle after the timer is started may be 1 clock or more. Therefore, be sure to follow the above sequence when using TM80 as an interval timer.

Table 7-3. Interval Time of 8-Bit Timer 80 (f_{XP} = 8.0 MHz)

TCL801	TCL800	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	$2^6/f_{XP}$ (8 μs)	$2^{14}/f_{XP}$ (2.05 ms)	$2^6/f_{XP}$ (8 μs)
0	1	$2^8/f_{XP}$ (32 μs)	$2^{16}/f_{XP}$ (8.19 ms)	$2^8/f_{XP}$ (32 μs)
1	0	$2^{10}/f_{XP}$ (128 μs)	$2^{18}/f_{XP}$ (32.7 ms)	$2^{10}/f_{XP}$ (128 μs)
1	1	$2^{16}/f_{XP}$ (8.19 ms)	$2^{24}/f_{XP}$ (2.01 s)	$2^{16}/f_{XP}$ (8.19 ms)

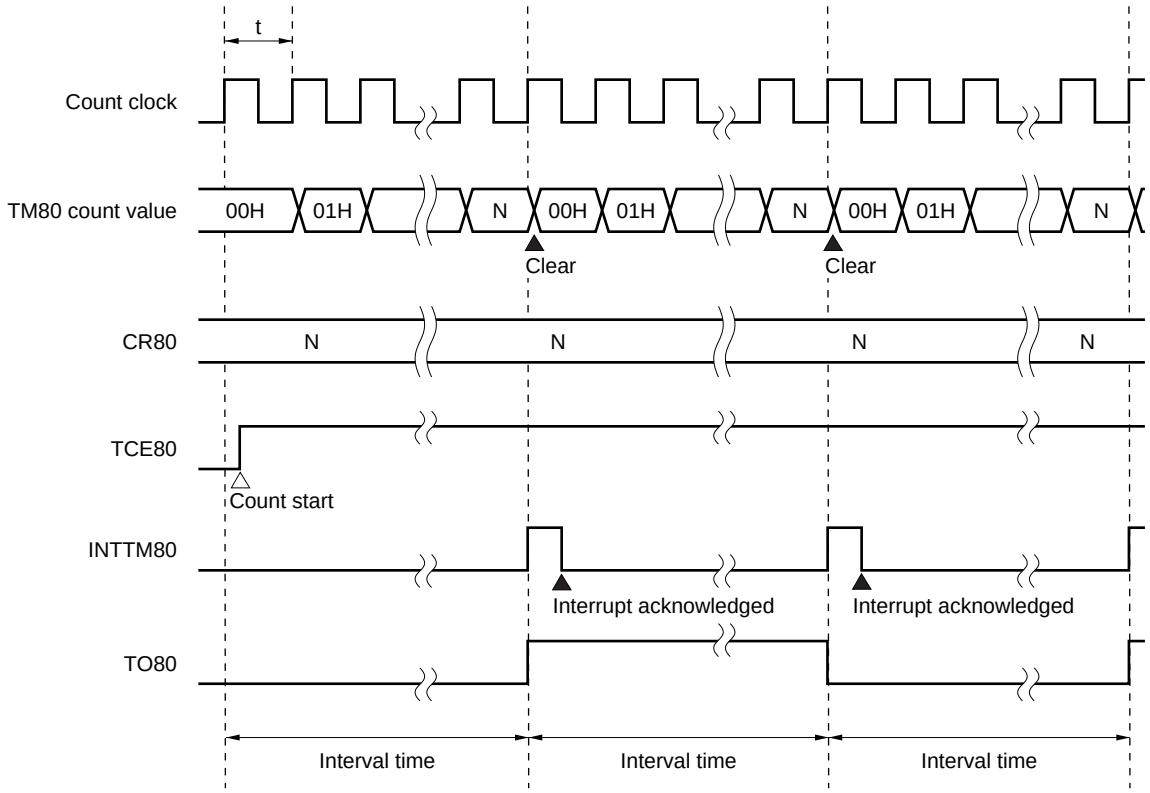
Remark f_{XP}: Oscillation frequency of clock to peripheral hardware

Table 7-4. Interval Time of 8-Bit Timer 80 (f_{XP} = 10.0 MHz)

TCL811	TCL810	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	$2^6/f_{XP}$ (6.4 μs)	$2^{14}/f_{XP}$ (1.64 ms)	$2^6/f_{XP}$ (6.4 μs)
0	1	$2^8/f_{XP}$ (25.6 μs)	$2^{16}/f_{XP}$ (6.55 ms)	$2^8/f_{XP}$ (25.6 μs)
1	0	$2^{10}/f_{XP}$ (102 μs)	$2^{18}/f_{XP}$ (26.2 ms)	$2^{10}/f_{XP}$ (102 μs)
1	1	$2^{16}/f_{XP}$ (6.55 ms)	$2^{24}/f_{XP}$ (1.68 s)	$2^{16}/f_{XP}$ (6.55 ms)

Remark f_{XP}: Oscillation frequency of clock to peripheral hardware

Figure 7-5. Timing of Interval Timer Operation



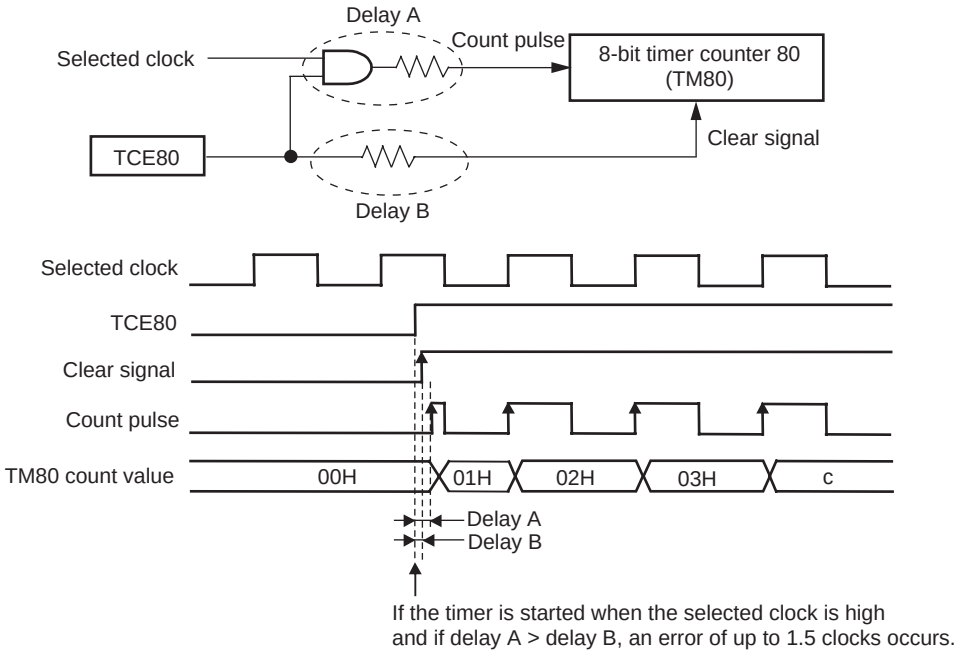
Remark Interval time = (N + 1) × t: N = 00H to FFH

7.5 Notes on 8-Bit Timer 80

(1) Error when timer starts

The time from starting the timer to generation of the match signal includes an error of up to 1.5 clocks. This is because, if the timer is started while the count clock is high, the rising edge may be immediately detected and the counter may be incremented (refer to **Figure 7-6**).

Figure 7-6. Case Where Error of 1.5 Clocks (Max.) Occurs



(2) Setting of 8-bit compare register 80

8-bit compare register 80 (CR80) can be set to 00H.

(3) Note on setting STOP mode

Before executing the STOP instruction, be sure to stop the timer operation (TCE80 = 0).

CHAPTER 8 8-BIT TIMER H1

8.1 Functions of 8-Bit Timer H1

8-bit timer H1 has the following functions.

- Interval timer
- PWM output mode
- Square-wave output

8.2 Configuration of 8-Bit Timer H1

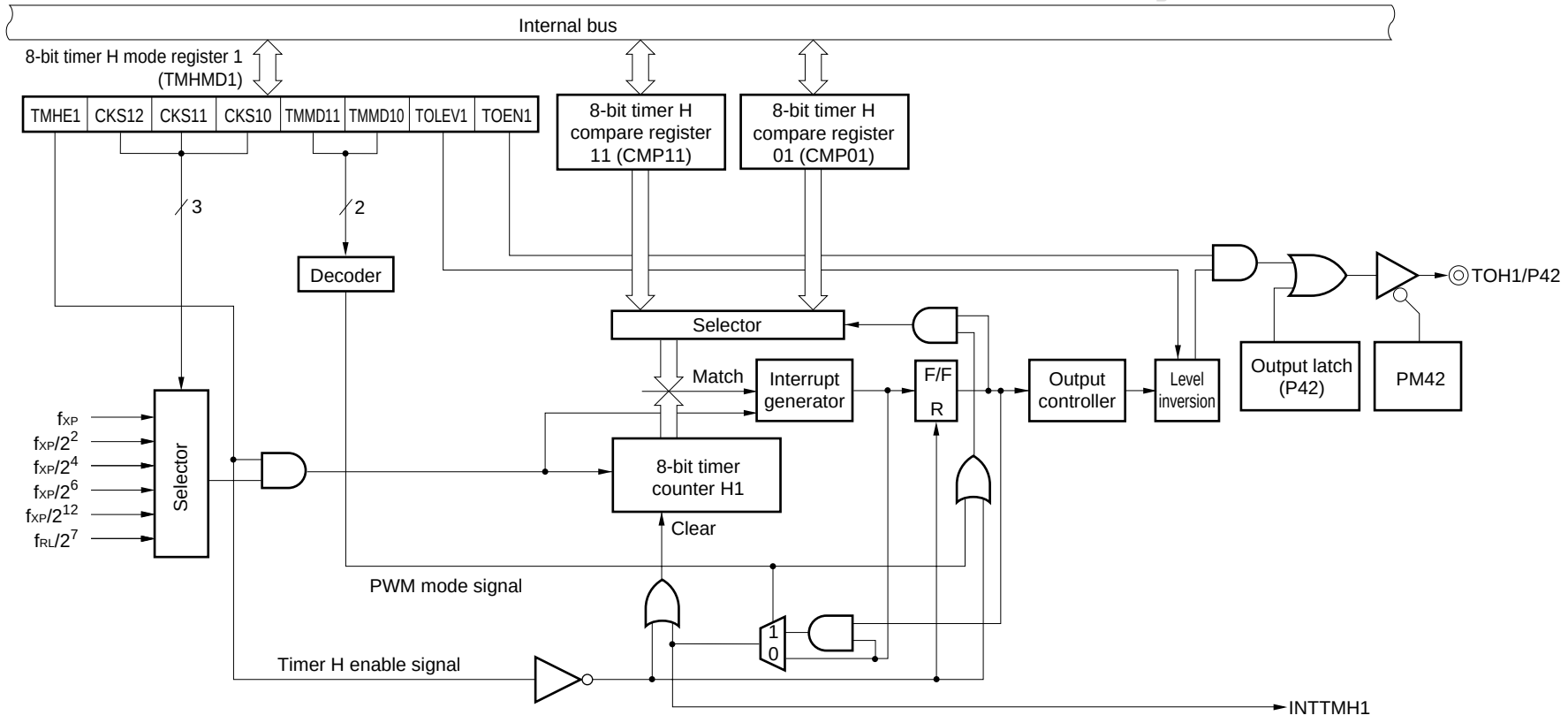
8-bit timer H1 consists of the following hardware.

Table 8-1. Configuration of 8-Bit Timer H1

Item	Configuration
Timer register	8-bit timer counter H1
Registers	8-bit timer H compare register 01 (CMP01) 8-bit timer H compare register 11 (CMP11)
Timer output	TOH1
Control registers	8-bit timer H mode register 1 (TMHMD1) Port mode register 4 (PM4) Port register 4 (P4)

Figure 8-1 shows a block diagram.

Figure 8-1. Block Diagram of 8-Bit Timer H1



(1) 8-bit timer H compare register 01 (CMP01)

This register can be read or written by an 8-bit memory manipulation instruction.

Reset input clears this register to 00H.

Figure 8-2. Format of 8-Bit Timer H Compare Register 01 (CMP01)

Address: FF0EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMP01								

Caution CMP01 cannot be rewritten during timer count operation.

(2) 8-bit timer H compare register 11 (CMP11)

This register can be read or written by an 8-bit memory manipulation instruction.

Reset input clears this register to 00H.

Figure 8-3. Format of 8-Bit Timer H Compare Register 11 (CMP11)

Address: FF0FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMP11								

CMP11 can be rewritten during timer count operation.

If the CMP11 value is rewritten during timer operation, transferring is performed at the timing at which the count value and CMP11 value match. If the transfer timing and writing from CPU to CMP11 conflict, transfer is not performed.

Caution In the PWM output mode, be sure to set CMP11 when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to CMP11).

8.3 Registers Controlling 8-Bit Timer H1

The following three registers are used to control 8-Bit Timer H1.

- 8-bit timer H mode register 1 (TMHMD1)
- Port mode register 4 (PM4)
- Port register 4 (P4)

(1) 8-bit timer H mode register 1 (TMHMD1)

This register controls the mode of timer H.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset input clears this register to 00H.

Figure 8-4. Format of 8-Bit Timer H Mode Register 1 (TMHMD1)

Address: FF70H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	<1>	<0>
TMHMD1	TMHE1	CKS12	CKS11	CKS10	TMMD11	TMMD10	TOLEV1	TOEN1

TMHE1	Timer operation enable
0	Stop timer count operation (counter is cleared to 0)
1	Enable timer count operation (count operation started by inputting clock)

CKS12	CKS11	CKS10	Count clock (f_{CNT}) selection
0	0	0	f_{XP} (10 MHz)
0	0	1	$f_{XP}/2^2$ (2.5 MHz)
0	1	0	$f_{XP}/2^4$ (625 kHz)
0	1	1	$f_{XP}/2^6$ (156.25 kHz)
1	0	0	$f_{XP}/2^{12}$ (2.44 kHz)
1	0	1	$f_{RL}/2^7$ (1.88 kHz (TYP.))
Other than above			Setting prohibited

TMMD11	TMMD10	Timer operation mode
0	0	Interval timer mode
1	0	PWM output mode
Other than above		Setting prohibited

TOLEV1	Timer output level control (in default mode)
0	Low level
1	High level

TOEN1	Timer output control
0	Disable output
1	Enable output

Cautions 1. When TMHE1 = 1, setting the other bits of the TMHMD1 register is prohibited.

2. In the PWM output mode, be sure to set 8-bit timer H compare register 11 (CMP11) when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to the CMP11 register).

Remarks 1. f_{XP}: Oscillation frequency of clock to peripheral hardware

2. f_{RL}: Low-speed Ring-OSC clock oscillation frequency

3. Figures in parentheses apply to operation at f_{XP} = 10 MHz, f_{RL} = 240 kHz (TYP.).

(2) Port mode register 4 (PM4)

This register sets port 4 input/output in 1-bit units.

When using the P42/TOH1 pin for timer output, clear PM42 and the output latch of P42 to 0.

PM4 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets this register to FFH.

Figure 8-5. Format of Port Mode Register 4 (PM4)

Address: FF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	1	1	PM45	PM44	PM43	PM42	PM41	PM40

PM4n	P4n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

8.4 Operation of 8-Bit Timer H1

8.4.1 Operation as interval timer/square-wave output

When 8-bit timer counter H1 and compare register 01 (CMP01) match, an interrupt request signal (INTTMH1) is generated and 8-bit timer counter H1 is cleared to 00H.

Compare register 11 (CMP11) is not used in interval timer mode. Since a match of 8-bit timer counter H1 and the CMP11 register is not detected even if the CMP11 register is set, timer output is not affected.

By setting bit 0 (TOEN1) of timer H mode register 1 (TMHMD1) to 1, a square wave of any frequency (duty = 50%) is output from TOH1.

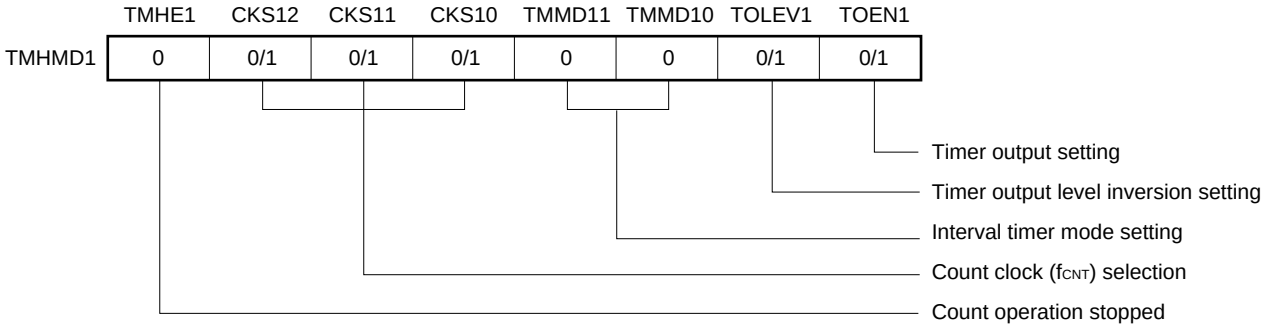
(1) Usage

Generates the INTTMH1 signal repeatedly at the same interval.

<1> Set each register.

Figure 8-6. Register Setting During Interval Timer/Square-Wave Output Operation

(i) Setting timer H mode register 1 (TMHMD1)



(ii) CMP01 register setting

- Compare value (N)

<2> Count operation starts when TMHE1 = 1.

<3> When the values of 8-bit timer counter H1 and the CMP01 register match, the INTTMH1 signal is generated and 8-bit timer counter H1 is cleared to 00H.

Interval time = (N +1)/f_{CNT}

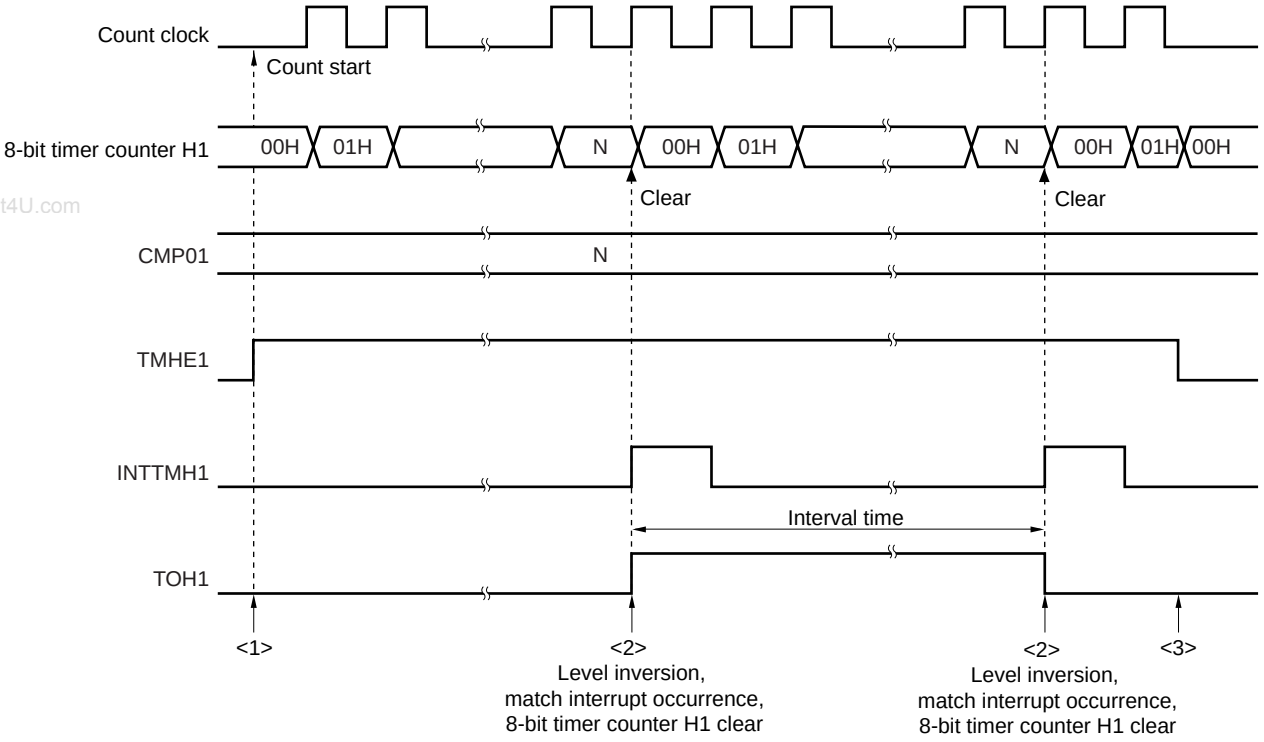
<4> Subsequently, the INTTMH1 signal is generated at the same interval. To stop the count operation, clear TMHE1 to 0.

(2) Timing chart

The timing of the interval timer/square-wave output operation is shown below.

Figure 8-7. Timing of Interval Timer/Square-Wave Output Operation (1/2)

(a) Basic operation



<1> The count operation is enabled by setting the TMHE1 bit to 1. The count clock starts counting no more than 1 clock after the operation is enabled.

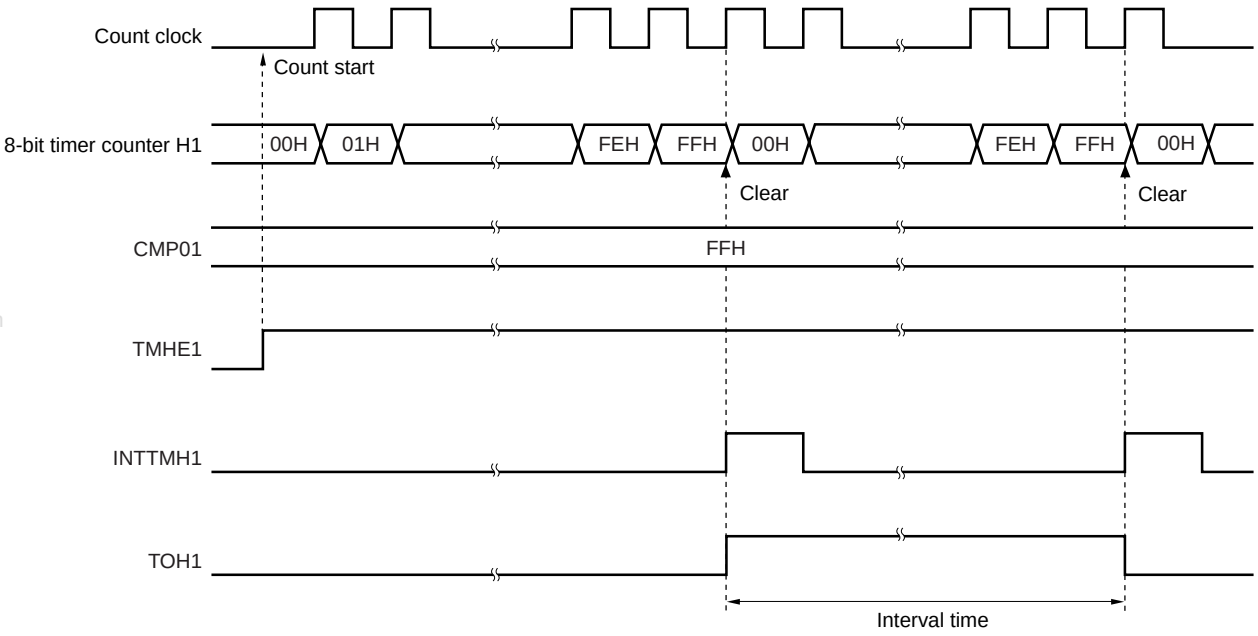
<2> When the values of 8-bit timer counter H1 and the CMP01 register match, the value of 8-bit timer counter H1 is cleared, the TOH1 output level is inverted, and the INTTMH1 signal is output.

<3> The INTTMH1 signal and TOH1 output become inactive by clearing the TMHE1 bit to 0 during timer H1 operation. If these are inactive from the first, the level is retained.

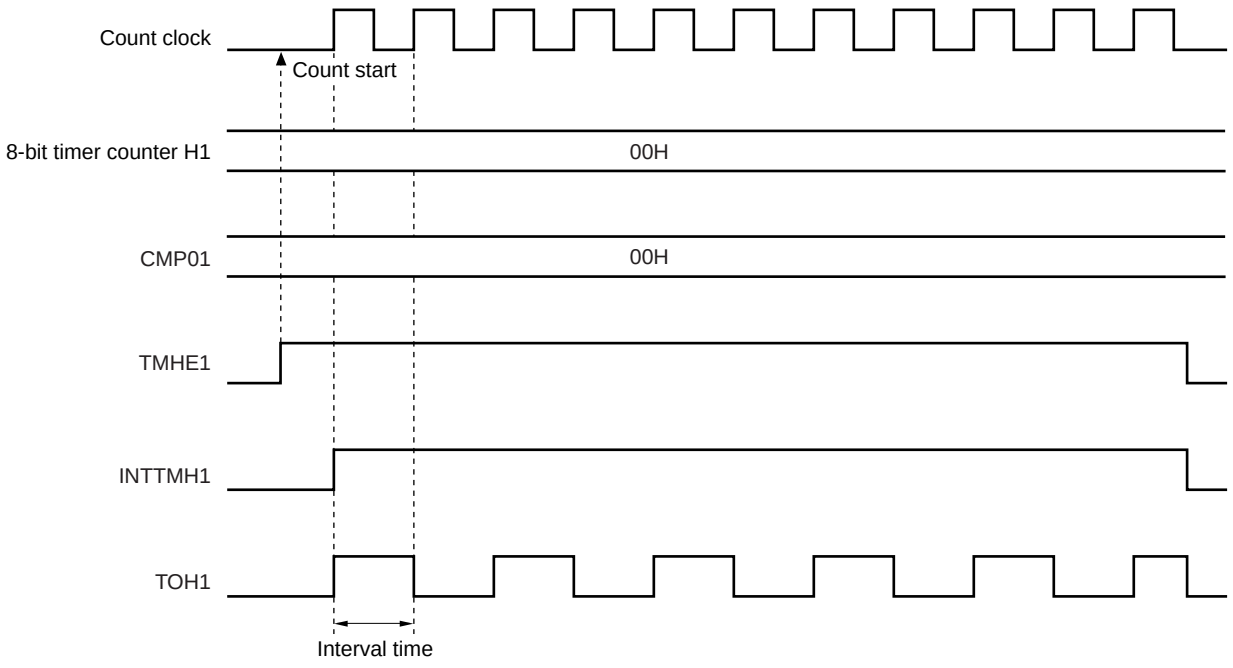
Remark N = 01H to FEH

Figure 8-7. Timing of Interval Timer/Square-Wave Output Operation (2/2)

(b) Operation when CMP01 = FFH



(c) Operation when CMP01 = 00H



8.4.2 Operation as PWM output mode

In PWM output mode, a pulse with an arbitrary duty and arbitrary cycle can be output.

8-bit timer compare register 01 (CMP01) controls the cycle of timer output (TOH1). Rewriting the CMP01 register during timer operation is prohibited.

8-bit timer compare register 11 (CMP11) controls the duty of timer output (TOH1). Rewriting the CMP11 register during timer operation is possible.

The operation in PWM output mode is as follows.

TOH1 output becomes active and 8-bit timer counter H1 is cleared to 0 when 8-bit timer counter H1 and the CMP01 register match after the timer count is started. TOH1 output becomes inactive when 8-bit timer counter H1 and the CMP11 register match.

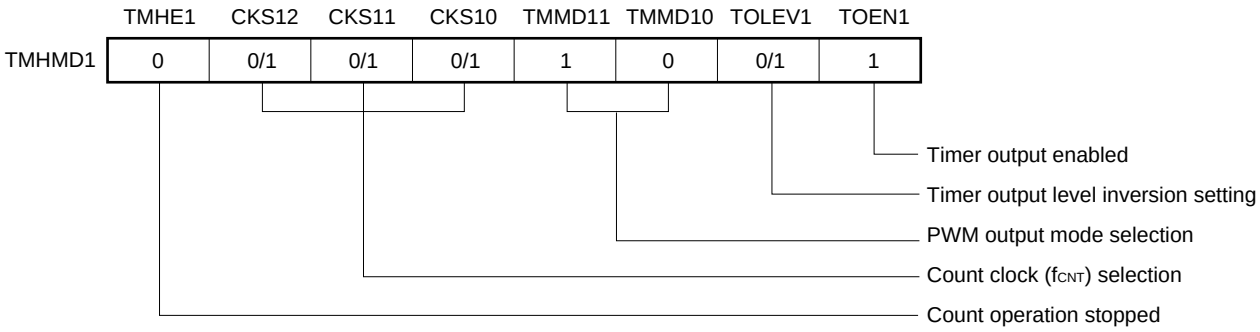
(1) Usage

In PWM output mode, a pulse for which an arbitrary duty and arbitrary cycle can be set is output.

<1> Set each register.

Figure 8-8. Register Setting in PWM Output Mode

(i) Setting timer H mode register 1 (TMHMD1)



(ii) Setting CMP01 register

- Compare value (N): Cycle setting

(iii) Setting CMP11 register

- Compare value (M): Duty setting

Remark 00H ≤ CMP11 (M) < CMP01 (N) ≤ FFH

<2> The count operation starts when TMHE1 = 1.

<3> The CMP01 register is the compare register that is to be compared first after count operation is enabled. When the values of 8-bit timer counter H1 and the CMP01 register match, 8-bit timer counter H1 is cleared, an interrupt request signal (INTTMH1) is generated, and TOH1 output becomes active. At the same time, the compare register to be compared with 8-bit timer counter H1 is changed from the CMP01 register to the CMP11 register.

<4> When 8-bit timer counter H1 and the CMP11 register match, TOH1 output becomes inactive and the compare register to be compared with 8-bit timer counter H1 is changed from the CMP11 register to the CMP01 register. At this time, 8-bit timer counter H1 is not cleared and the INTTMH1 signal is not generated.

<5> By performing procedures <3> and <4> repeatedly, a pulse with an arbitrary duty can be obtained.

<6> To stop the count operation, set TMHE1 = 0.

If the setting value of the CMP01 register is N, the setting value of the CMP11 register is M, and the count clock frequency is f_{CNT}, the PWM pulse output cycle and duty are as follows.

PWM pulse output cycle = $(N+1)/f_{CNT}$
Duty = Active width : Total width of PWM = (M + 1) : (N + 1)

Cautions 1. In PWM output mode, three operation clocks (signal selected using the CKS12 to CKS10 bits of the TMHMD1 register) are required to transfer the CMP11 register value after rewriting the register.

2. Be sure to set the CMP11 register when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to the CMP11 register).

(2) Timing chart

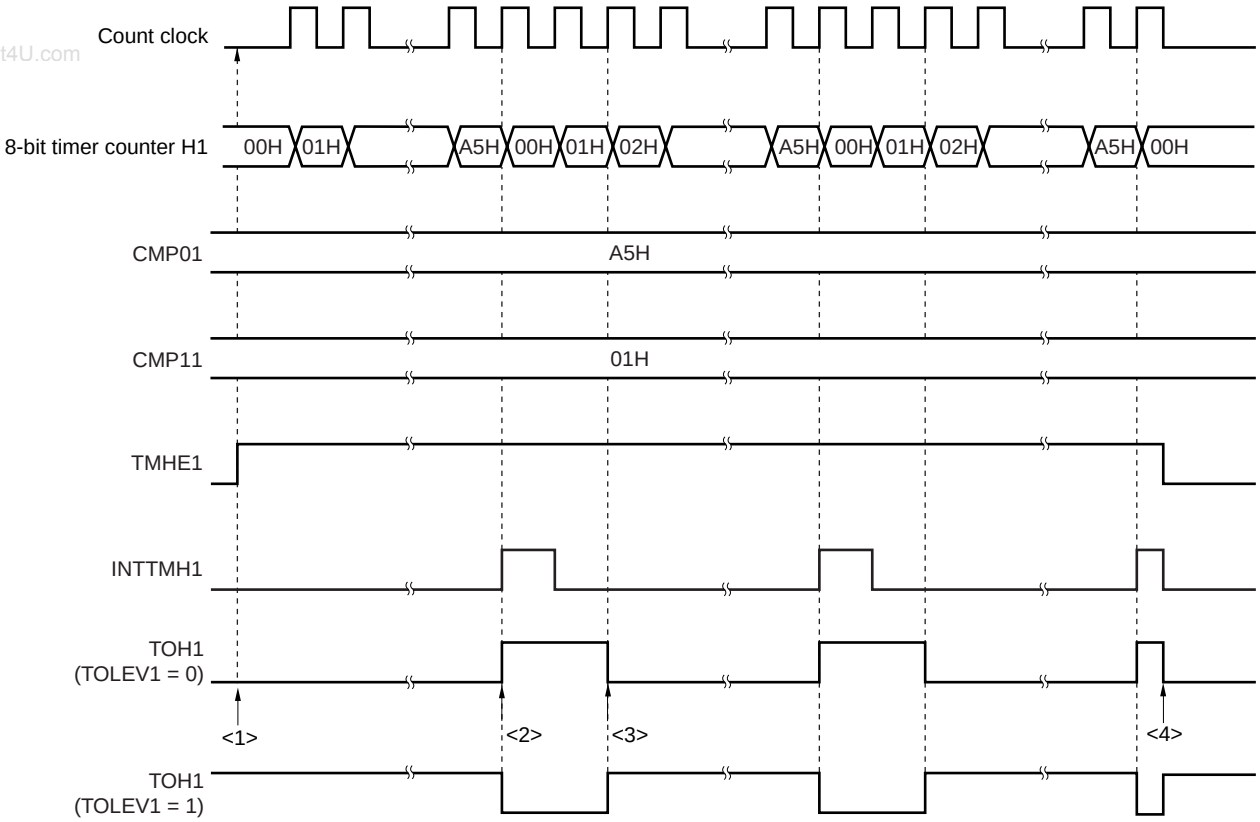
The operation timing in PWM output mode is shown below.

Caution Make sure that the CMP11 register setting value (M) and CMP01 register setting value (N) are within the following range.

00H ≤ CMP11 (M) < CMP01 (N) ≤ FFH

Figure 8-9. Operation Timing in PWM Output Mode (1/4)

(a) Basic operation



<1> The count operation is enabled by setting the TMHE1 bit to 1. Start 8-bit timer counter H1 by masking one count clock to count up. At this time, TOH1 output remains inactive (when TOLEV1 = 0).

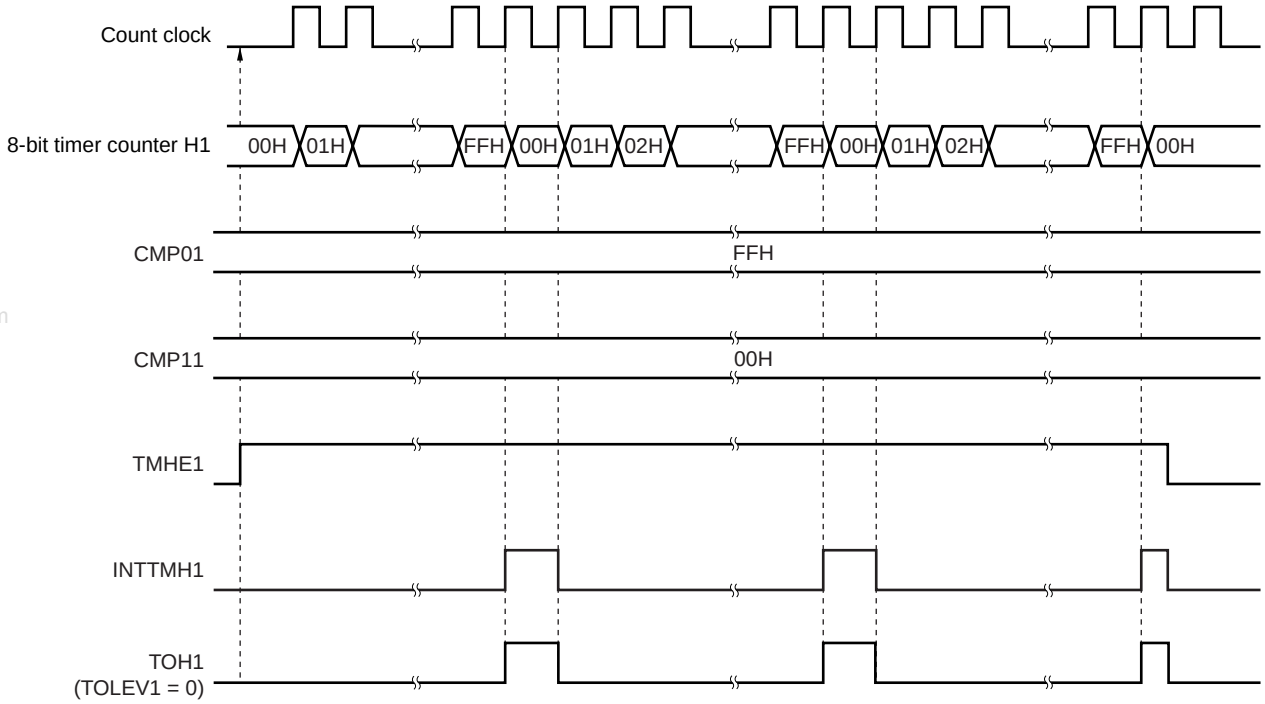
<2> When the values of 8-bit timer counter H1 and the CMP01 register match, the TOH1 output level is inverted, the value of 8-bit timer counter H1 is cleared, and the INTTMH1 signal is output.

<3> When the values of 8-bit timer counter H1 and the CMP11 register match, the level of the TOH1 output is returned. At this time, the 8-bit timer counter value is not cleared and the INTTMH1 signal is not output.

<4> Clearing the TMHE1 bit to 0 during timer H1 operation makes the INTTMH1 signal and TOH1 output inactive.

Figure 8-9. Operation Timing in PWM Output Mode (2/4)

(b) Operation when CMP01 = FFH, CMP11 = 00H



(c) Operation when CMP01 = FFH, CMP11 = FEH

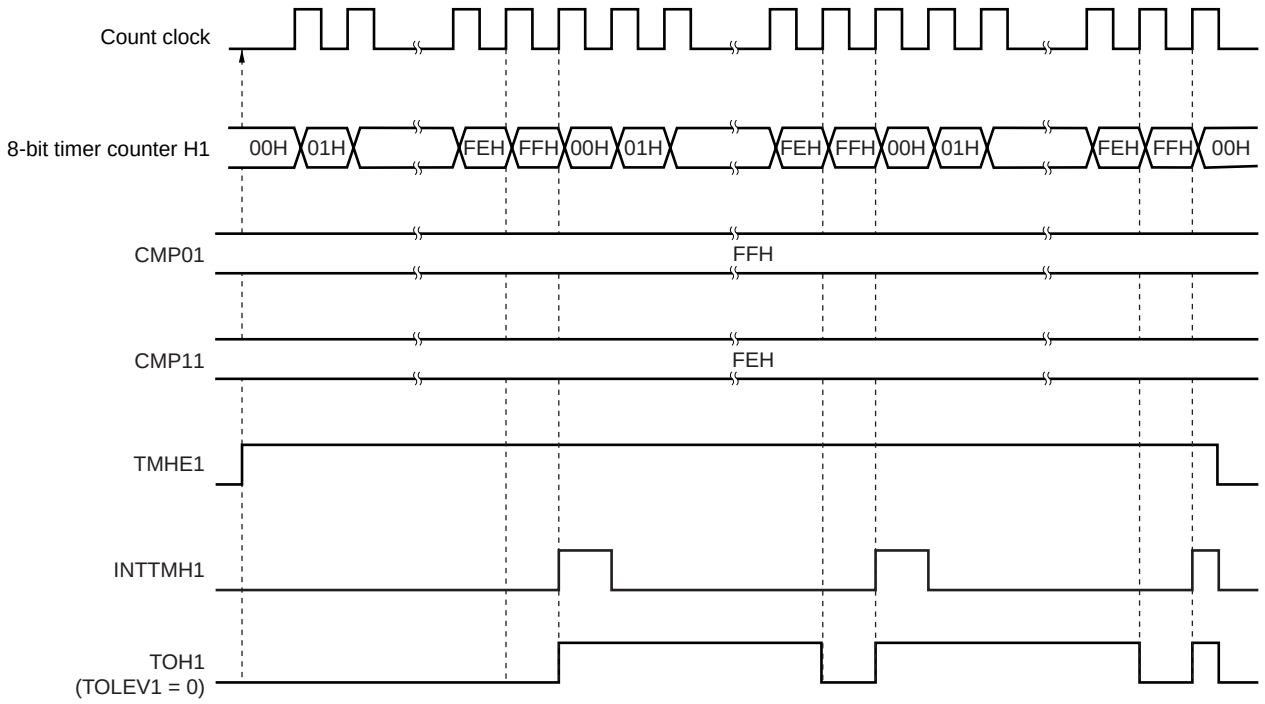


Figure 8-9. Operation Timing in PWM Output Mode (3/4)

(d) Operation when CMP01 = 01H, CMP11 = 00H

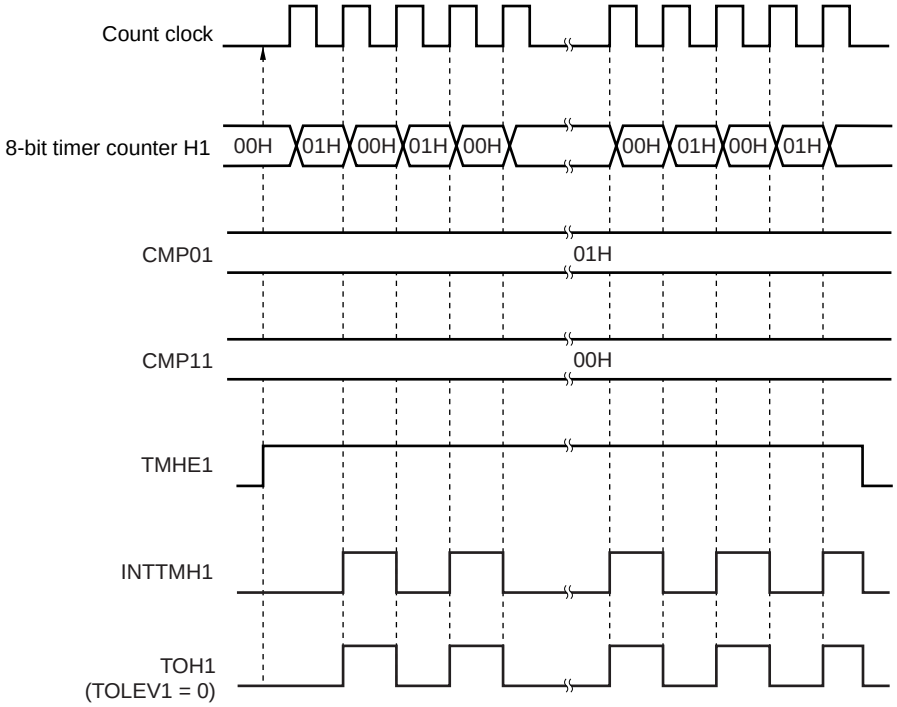
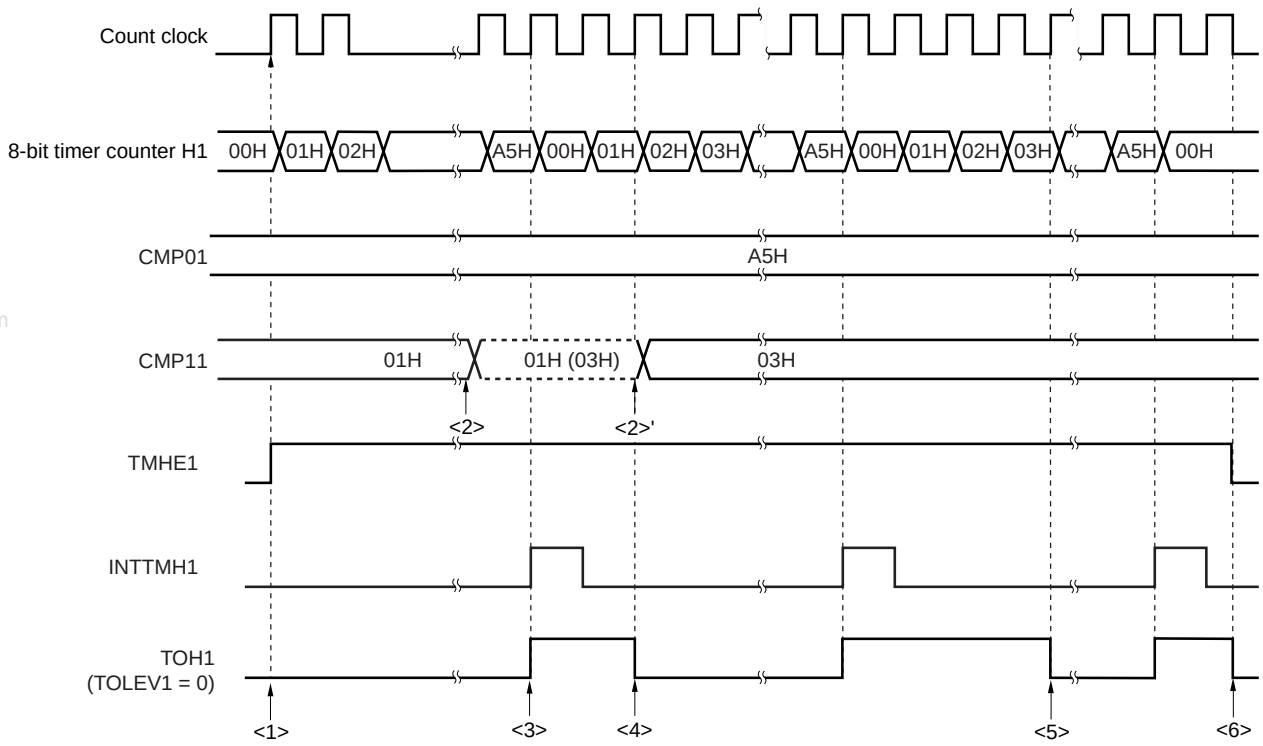


Figure 8-9. Operation Timing in PWM Output Mode (4/4)

(e) Operation by changing CMP11 (CMP11 = 01H → 03H, CMP01 = A5H)



<1> The count operation is enabled by setting TMHE1 = 1. Start 8-bit timer counter H1 by masking one count clock to count up. At this time, the TOH1 output remains inactive (when TOLEV1 = 0).

<2> The CMP11 register value can be changed during timer counter operation. This operation is asynchronous to the count clock.

<3> When the values of 8-bit timer counter H1 and the CMP01 register match, the value of 8-bit timer counter H1 is cleared, the TOH1 output becomes active, and the INTTMH1 signal is output.

<4> If the CMP11 register value is changed, the value is latched and not transferred to the register. When the values of 8-bit timer counter H1 and the CMP11 register before the change match, the value is transferred to the CMP11 register and the CMP11 register value is changed (<2>').

However, three count clocks or more are required from when the CMP11 register value is changed to when the value is transferred to the register. If a match signal is generated within three count clocks, the changed value cannot be transferred to the register.

<5> When the values of 8-bit timer counter H1 and the CMP11 register after the change match, the TOH1 output becomes inactive. 8-bit timer counter H1 is not cleared and the INTTMH1 signal is not generated.

<6> Clearing the TMHE1 bit to 0 during timer H1 operation makes the INTTMH1 signal and TOH1 output inactive.

CHAPTER 9 WATCHDOG TIMER

9.1 Functions of Watchdog Timer

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1. For details of RESF, see **CHAPTER 14 RESET FUNCTION**.

Table 9-1. Loop Detection Time of Watchdog Timer

Loop Detection Time	
During Low-Speed Ring-OSC Clock Operation	During Operation of Clock to Peripheral Hardware
$f_{RL}/2^{11}$ (8.53 ms)	$f_{XP}/2^{13}$ (819.2 μs)
$f_{RL}/2^{12}$ (17.07 ms)	$f_{XP}/2^{14}$ (1.64 ms)
$f_{RL}/2^{13}$ (34.13 ms)	$f_{XP}/2^{15}$ (3.28 ms)
$f_{RL}/2^{14}$ (68.27 ms)	$f_{XP}/2^{16}$ (6.55 ms)
$f_{RL}/2^{15}$ (136.53 ms)	$f_{XP}/2^{17}$ (13.11 ms)
$f_{RL}/2^{16}$ (273.07 ms)	$f_{XP}/2^{18}$ (26.21 ms)
$f_{RL}/2^{17}$ (546.13 ms)	$f_{XP}/2^{19}$ (52.43 ms)
$f_{RL}/2^{18}$ (1.09 s)	$f_{XP}/2^{20}$ (104.86 ms)

Remarks 1. f_{RL}: Low-speed Ring-OSC clock oscillation frequency

2. f_{XP}: Oscillation frequency of clock to peripheral hardware

3. Figures in parentheses apply to operation at f_{RL} = 240 kHz (TYP.), f_{XP} = 10 MHz.

The operation mode of the watchdog timer (WDT) is switched according to the option byte setting of the on-chip low-speed Ring-OSC oscillator as shown in Table 9-2.

Table 9-2. Option Byte Setting and Watchdog Timer Operation Mode

	Option Byte Setting	
	Low-Speed Ring-OSC Cannot Be Stopped	Low-Speed Ring-OSC Can Be Stopped by Software
Watchdog timer clock source	Fixed to f_{RL} Note 1.	• Selectable by software (f_{XP}, f_{RL} or stopped) • When reset is released: f_{RL}
Operation after reset	Operation starts with the maximum interval ($f_{RL}/2^{18}$).	Operation starts with the maximum interval ($f_{RL}/2^{18}$).
Operation mode selection	The interval can be changed only once.	The clock selection/interval can be changed only once.
Features	The watchdog timer cannot be stopped.	The watchdog timer can be stopped Note 2.

Notes 1. As long as power is being supplied, low-speed Ring-OSC oscillation cannot be stopped (except in the reset period).

2. The conditions under which clock supply to the watchdog timer is stopped differ depending on the clock source of the watchdog timer.

<1> If the clock source is f_{XP}, clock supply to the watchdog timer is stopped under the following conditions.

- When f_{XP} is stopped
- In HALT/STOP mode
- During oscillation stabilization time

<2> If the clock source is f_{RL}, clock supply to the watchdog timer is stopped under the following conditions.

- If the CPU clock is f_{XP} and if f_{RL} is stopped by software before execution of the STOP instruction
- In HALT/STOP mode

Remarks 1. f_{RL}: Low-speed Ring-OSC clock oscillation frequency

2. f_{XP}: Oscillation frequency of clock to peripheral hardware

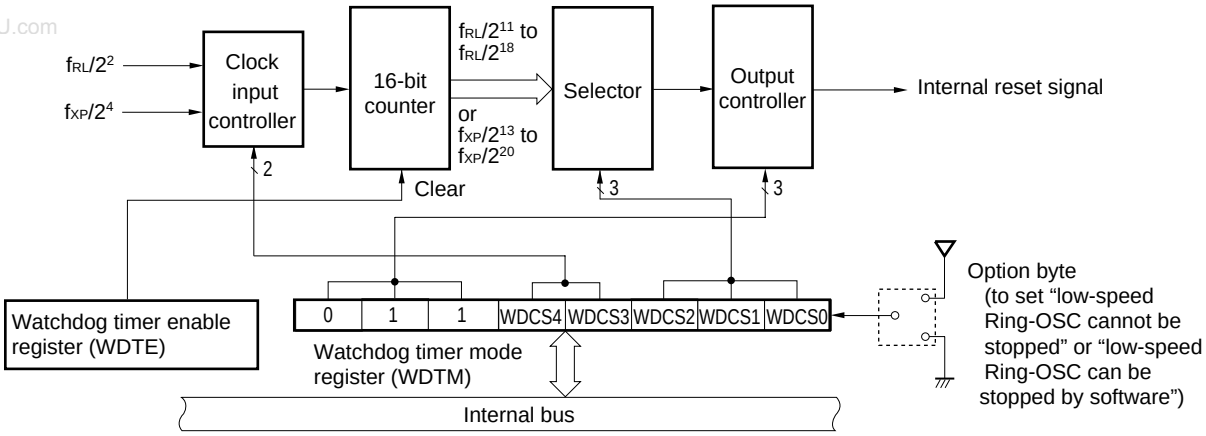
9.2 Configuration of Watchdog Timer

The watchdog timer consists of the following hardware.

Table 9-3. Configuration of Watchdog Timer

Item	Configuration
Control registers	Watchdog timer mode register (WDTM) Watchdog timer enable register (WDTE)

Figure 9-1. Block Diagram of Watchdog Timer



9.3 Registers Controlling Watchdog Timer

The watchdog timer is controlled by the following two registers.

- Watchdog timer mode register (WDTM)
- Watchdog timer enable register (WDTE)

(1) Watchdog timer mode register (WDTM)

This register sets the overflow time and operation clock of the watchdog timer.

This register can be set by an 8-bit memory manipulation instruction and can be read many times, but can be written only once after reset is released.

Reset input sets this register to 67H.

Figure 9-2. Format of Watchdog Timer Mode Register (WDTM)

Address: FF48H After reset: 67H R/W

Symbol	7	6	5	4	3	2	1	0
WDTM	0	1	1	WDCS4	WDCS3	WDCS2	WDCS1	WDCS0

WDCS4[Note 1]	WDCS3[Note 1]	Operation clock selection
0	0	Low-speed Ring-OSC clock (f_{RL})
0	1	Clock to peripheral hardware (f_{XP})
1	×	Watchdog timer operation stopped

WDCS2[Note 2]	WDCS1[Note 2]	WDCS0[Note 2]	Overflow time setting	
			During low-speed Ring-OSC clock operation	During operation of clock to peripheral hardware
0	0	0	$f_{RL}/2^{11}$ (8.53 ms)	$f_{XP}/2^{13}$ (819.2 μs)
0	0	1	$f_{RL}/2^{12}$ (17.07 ms)	$f_{XP}/2^{14}$ (1.64 ms)
0	1	0	$f_{RL}/2^{13}$ (34.13 ms)	$f_{XP}/2^{15}$ (3.28 ms)
0	1	1	$f_{RL}/2^{14}$ (68.27 ms)	$f_{XP}/2^{16}$ (6.55 ms)
1	0	0	$f_{RL}/2^{15}$ (136.53 ms)	$f_{XP}/2^{17}$ (13.11 ms)
1	0	1	$f_{RL}/2^{16}$ (273.07 ms)	$f_{XP}/2^{18}$ (26.21 ms)
1	1	0	$f_{RL}/2^{17}$ (546.13 ms)	$f_{XP}/2^{19}$ (52.43 ms)
1	1	1	$f_{RL}/2^{18}$ (1.09 s)	$f_{XP}/2^{20}$ (104.86 ms)

Notes 1. If “low-speed Ring-OSC cannot be stopped” is specified by the option byte, this cannot be set. The low-speed Ring-OSC clock will be selected no matter what value is written.

2. Reset is released at the maximum cycle (WDCS2, 1, 0 = 1, 1, 1).

Cautions 1. Set bits 7, 6, and 5 to 0, 1, and 1, respectively (when "low-speed Ring-OSC cannot be stopped" is selected by the option byte, other values are ignored).

2. After reset is released, WDTM can be written only once by an 8-bit memory manipulation instruction. If writing is attempted a second time, an internal reset signal is generated.

3. WDTM cannot be set by a 1-bit memory manipulation instruction.

Remarks 1. f_{RL}: Low-speed Ring-OSC clock oscillation frequency

2. f_{XP}: Oscillation frequency of clock to peripheral hardware

3. ×: Don't care

4. Figures in parentheses apply to operation at f_{RL} = 240 kHz (TYP.), f_{XP} = 10 MHz.

(2) Watchdog timer enable register (WDTE)

Writing ACH to WDTE clears the watchdog timer counter and starts counting again.

This register can be set by an 8-bit memory manipulation instruction.

Reset input sets this register to 9AH.

Figure 9-3. Format of Watchdog Timer Enable Register (WDTE)

Address: FF49H After reset: 9AH R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Cautions 1. If a value other than ACH is written to WDTE, an internal reset signal is generated.

2. If a 1-bit memory manipulation instruction is executed for WDTE, an internal reset signal is generated.

3. The value read from WDTE is 9AH (this differs from the written value (ACH)).

9.4 Operation of Watchdog Timer

9.4.1 Watchdog timer operation when "low-speed Ring-OSC cannot be stopped" is selected by option byte

The operation clock of watchdog timer is fixed to low-speed Ring-OSC.

After reset is released, operation is started at the maximum cycle (bits 2, 1, and 0 (WDCS2, WDCS1, WDCS0) of the watchdog timer mode register (WDTM) = 1, 1, 1). The watchdog timer operation cannot be stopped.

The following shows the watchdog timer operation after reset release.

1. The status after reset release is as follows.
   - Operation clock: Low-speed Ring-OSC clock
   - Cycle: $f_{RL}/2^{18}$ (1.09 seconds: At operation with f_{RL} = 240 kHz (TYP.))
   - Counting starts
2. The following should be set in the watchdog timer mode register (WDTM) by an 8-bit memory manipulation instruction[Notes 1, 2].
   - Cycle: Set using bits 2 to 0 (WDCS2 to WDCS0)
3. After the above procedures are executed, writing ACH to WDTE clears the count to 0, enabling recounting.

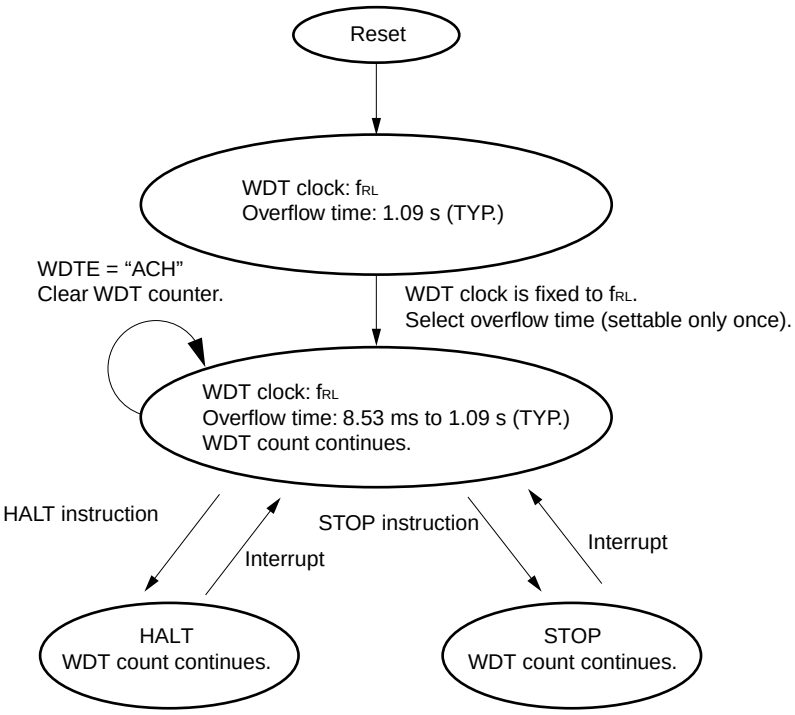
Notes 1. The operation clock (low-speed Ring-OSC clock) cannot be changed. If any value is written to bits 3 and 4 (WDCS3, WDCS4) of WDTM, it is ignored.

2. As soon as WDTM is written, the counter of the watchdog timer is cleared.

Caution In this mode, operation of the watchdog timer cannot be stopped even during STOP instruction execution. For 8-bit timer H1 (TMH1), a division of the low-speed Ring-OSC clock can be selected as the count source, so clear the watchdog timer using the interrupt request of TMH1 before the watchdog timer overflows after STOP instruction execution. If this processing is not performed, an internal reset signal is generated when the watchdog timer overflows after STOP instruction execution.

A status transition diagram is shown below

Figure 9-4. Status Transition Diagram When "Low-Speed Ring-OSC Cannot Be Stopped" Is Selected by Option Byte



9.4.2 Watchdog timer operation when "low-speed Ring-OSC can be stopped by software" is selected by option byte

The operation clock of the watchdog timer can be selected as either the low-speed Ring-OSC clock or the clock to peripheral hardware.

After reset is released, operation is started at the maximum cycle of the low-speed Ring-OSC clock (bits 2, 1, and 0 (WDCS2, WDCS1, WDCS0) of the watchdog timer mode register (WDTM) = 1, 1, 1).

The following shows the watchdog timer operation after reset release.

1. The status after reset release is as follows.
   - Operation clock: Low-speed Ring-OSC clock
   - Cycle: $f_{RL}/2^{18}$ (1.09 seconds: At operation with f_{RL} = 240 kHz (TYP.))
   - Counting starts
2. The following should be set in the watchdog timer mode register (WDTM) by an 8-bit memory manipulation instruction[Notes 1, 2, 3].
   - Operation clock: Any of the following can be selected using bits 3 and 4 (WDCS3 and WDCS4).
     Low-speed Ring-OSC clock (f_{RL})
     Clock to peripheral hardware (f_{XP})
     Watchdog timer operation stopped
   - Cycle: Set using bits 2 to 0 (WDCS2 to WDCS0)
3. After the above procedures are executed, writing ACH to WDTE clears the count to 0, enabling recounting.

Notes 1. As soon as WDTM is written, the counter of the watchdog timer is cleared.

2. Set bits 7, 6, and 5 to 0, 1, 1, respectively. Do not set the other values.

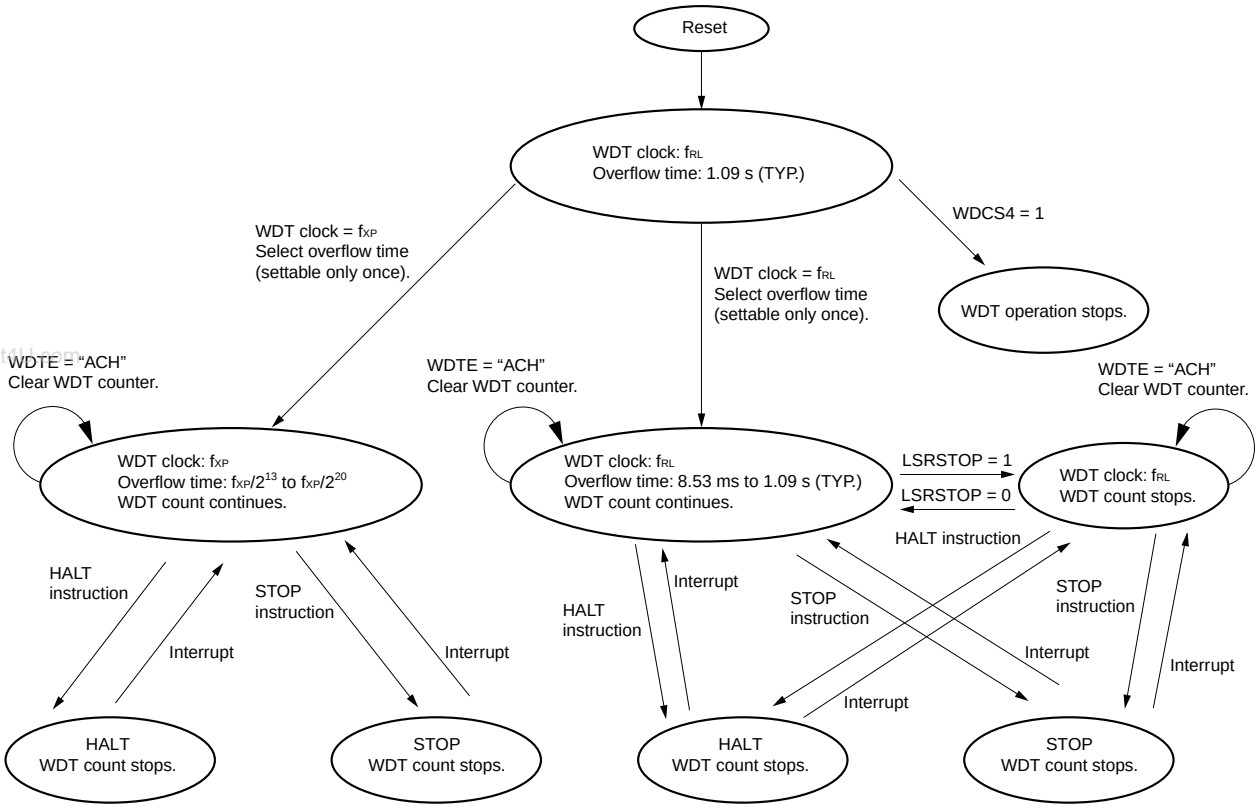
3. If the watchdog timer is stopped by setting WDCS4 and WDCS3 to 1 and ×, respectively, an internal reset signal is not generated even if the following processing is performed.
   - WDTM is written a second time.
   - A 1-bit memory manipulation instruction is executed to WDTE.
   - A value other than ACH is written to WDTE.

Caution In this mode, watchdog timer operation is stopped during HALT/STOP instruction execution. After HALT/STOP mode is released, counting is started again using the operation clock of the watchdog timer set before HALT/STOP instruction execution by WDTM. At this time, the counter is not cleared to 0 but holds its value.

For the watchdog timer operation during STOP mode and HALT mode in each status, see **9.4.3 Watchdog timer operation in STOP mode** and **9.4.4 Watchdog timer operation in HALT mode**.

A status transition diagram is shown below.

Figure 9-5. Status Transition Diagram When "Low-Speed Ring-OSC Can Be Stopped by Software" Is Selected by Option Byte



9.4.3 Watchdog timer operation in STOP mode (when "low-speed Ring-OSC can be stopped by software" is selected by option byte)

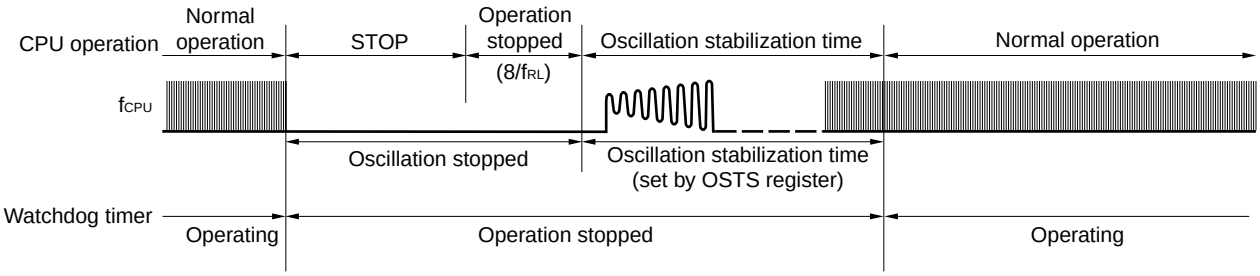
The watchdog timer stops counting during STOP instruction execution regardless of whether the clock to peripheral hardware or low-speed Ring-OSC clock is being used.

(1) When the watchdog timer operation clock is the clock to peripheral hardware (f_{XP}) when the STOP instruction is executed

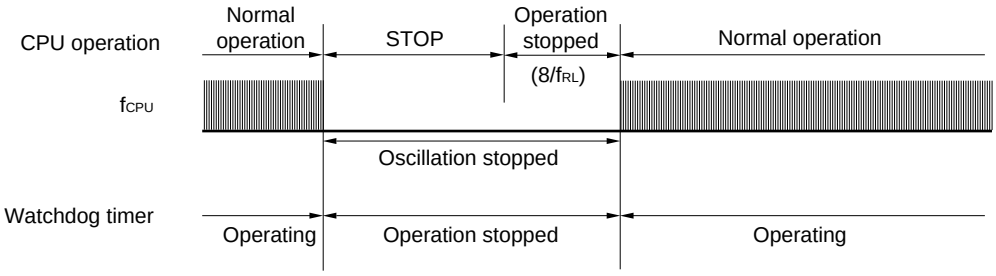
When STOP instruction is executed, operation of the watchdog timer is stopped. After STOP mode is released, operation stops for 8 clocks of the low-speed Ring-OSC clock (after waiting for the oscillation stabilization time set by the oscillation stabilization time select register (OSTS) after operation stops in the case of crystal/ceramic oscillation) and then counting is started again using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0 but holds its value.

Figure 9-6. Operation in STOP Mode (WDT Operation Clock: Clock to Peripheral Hardware)

<1> CPU clock: Crystal/ceramic oscillation clock



<2> CPU clock: High-speed Ring-OSC clock or external clock input

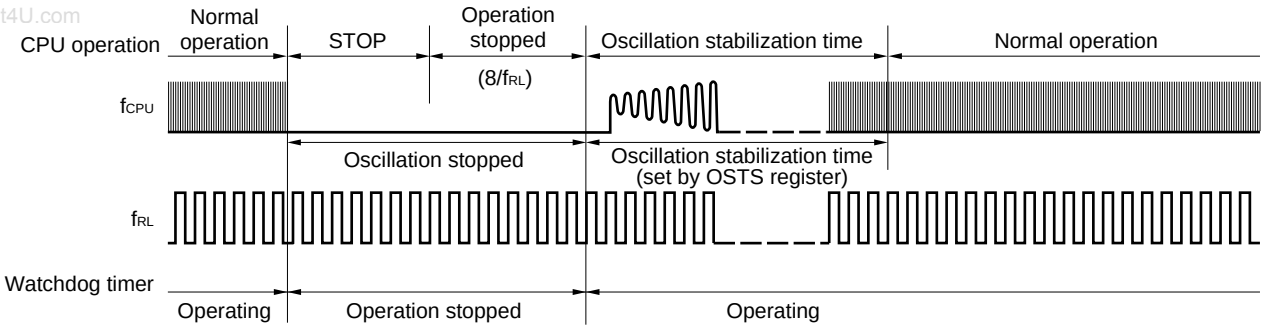


(2) When the watchdog timer operation clock is the low-speed Ring-OSC clock (f_{RL}) when the STOP instruction is executed

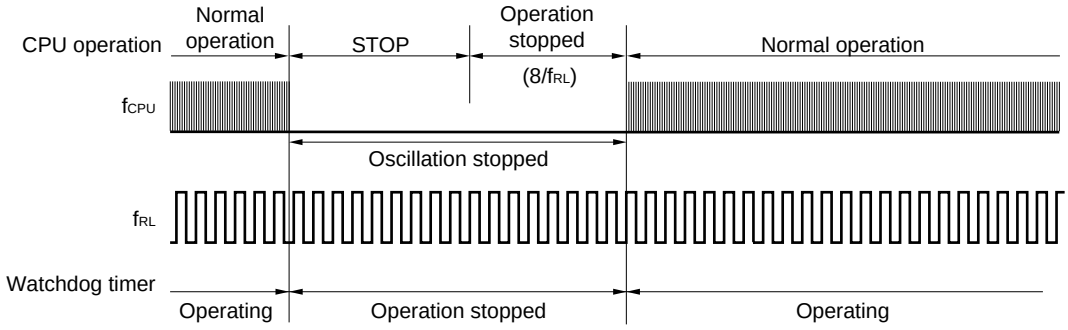
When the STOP instruction is executed, operation of the watchdog timer is stopped. After STOP mode is released, operation stops for 8 clocks of the low-speed Ring-OSC clock and then counting is started again using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0 but holds its value.

Figure 9-7. Operation in STOP Mode (WDT Operation Clock: Low-Speed Ring-OSC Clock)

<1> CPU clock: Crystal/ceramic oscillation clock



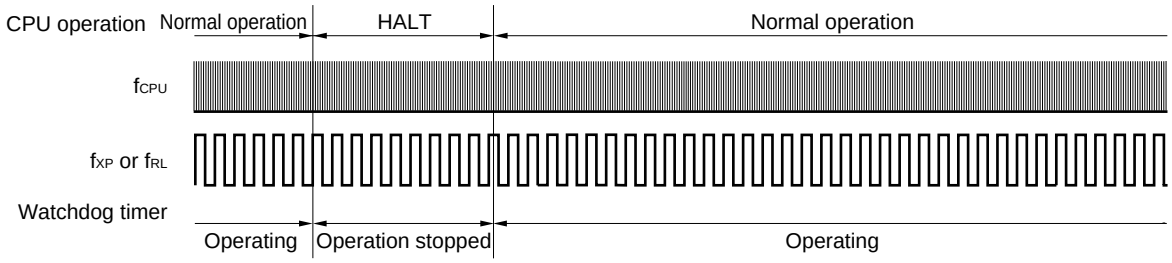
<2> CPU clock: High-speed Ring-OSC clock or external clock input



9.4.4 Watchdog timer operation in HALT mode (when “low-speed Ring-OSC can be stopped by software” is selected by option byte)

The watchdog timer stops counting during HALT instruction execution regardless of whether the operation clock of the watchdog timer is the clock to peripheral hardware (f_{XP}) or low-speed Ring-OSC clock (f_{RL}). After HALT mode is released, counting is started again using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0 but holds its value.

Figure 9-8. Operation in HALT Mode



CHAPTER 10 A/D CONVERTER

10.1 Functions of A/D Converter

The A/D converter converts an analog input signal into a digital value, and consists of up to four channels (ANI0 to ANI3) with a resolution of 10 bits.

The A/D converter has the following function.

- **10-bit resolution A/D conversion**
  10-bit resolution A/D conversion is carried out repeatedly for one channel selected from analog inputs ANI0 to ANI3. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

Figure 10-1 shows the timing of sampling and A/D conversion, and Table 10-1 shows the sampling time and A/D conversion time.

Figure 10-1. Timing of A/D Converter Sampling and A/D Conversion

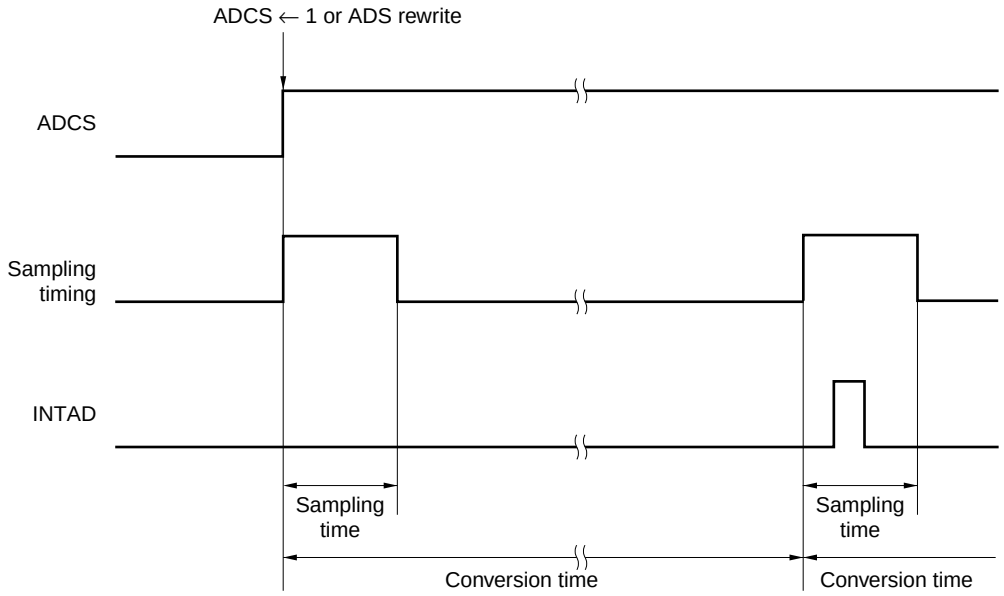


Table 10-1. Sampling Time and A/D Conversion Time

FR2	FR1	FR0	Sampling Time[Note 1]	Conversion Time[Note 2]	f_{XP} = 8 MHz		f_{XP} = 10 MHz	
					Sampling Time[Note 1]	Conversion Time[Note 2]	Sampling Time[Note 1]	Conversion Time[Note 2]
0	0	0	$12/f_{XP}$	$36/f_{XP}$	1.5 μs	4.5 μs	1.2 μs	3.6 μs
0	0	1	$24/f_{XP}$	$48/f_{XP}$	3.0 μs	6.0 μs	2.4 μs	4.8 μs
0	1	0	$48/f_{XP}$	$72/f_{XP}$	6.0 μs	9.0 μs	4.8 μs	7.2 μs
0	1	1	$88/f_{XP}$	$112/f_{XP}$	11.0 μs	14.0 μs	8.8 μs	11.2 μs
1	0	0	$24/f_{XP}$	$72/f_{XP}$	3.0 μs	9.0 μs	2.4 μs	7.2 μs
1	0	1	$48/f_{XP}$	$96/f_{XP}$	6.0 μs	12.0 μs	4.8 μs	9.6 μs
1	1	0	$96/f_{XP}$	$144/f_{XP}$	12.0 μs	18.0 μs	9.6 μs	14.4 μs
1	1	1	$176/f_{XP}$	$224/f_{XP}$	22.0 μs	28.0 μs	17.2 μs	22.4 μs

Notes 1. Set the sampling time as follows.

- $AV_{REF} \geq 4.5$ V: 1.0 μs or more
- $AV_{REF} \geq 4.0$ V: 2.4 μs or more
- $AV_{REF} \geq 2.85$ V: 3.0 μs or more
- $AV_{REF} \geq 2.7$ V: 11.0 μs or more

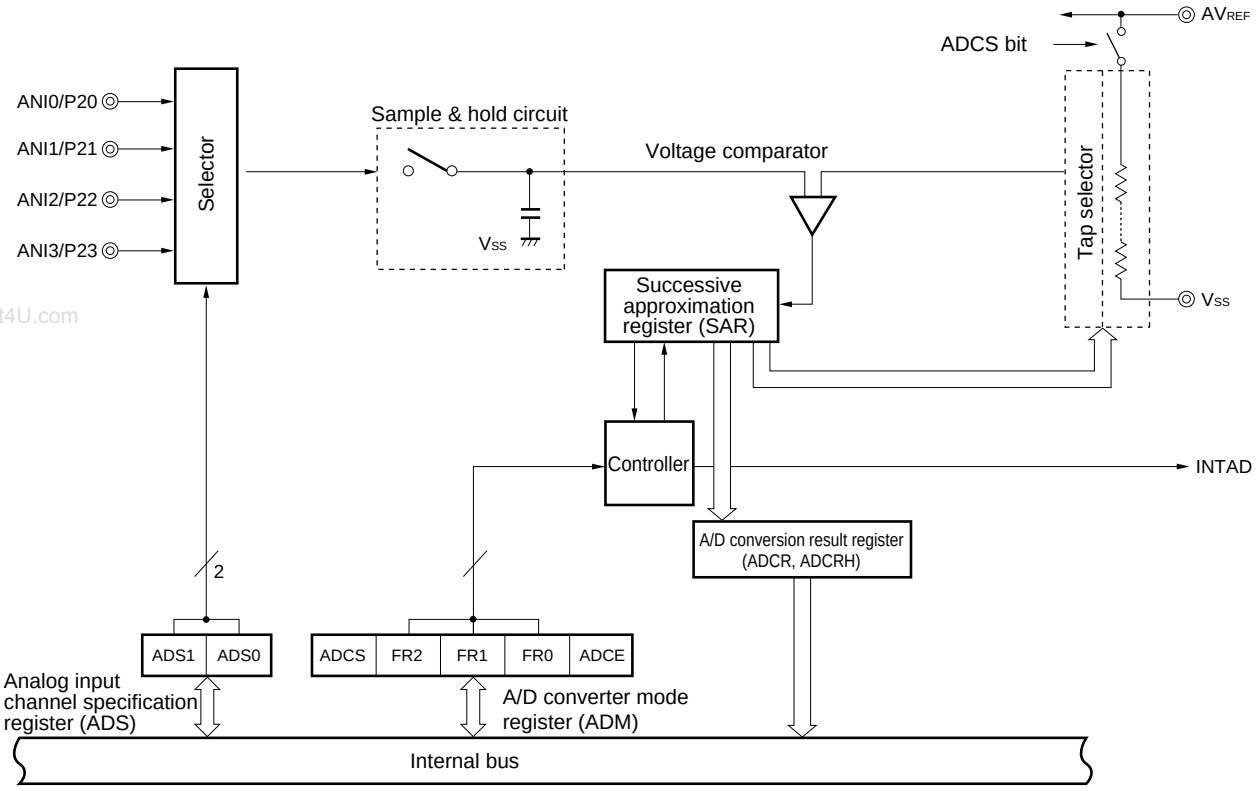
2. Set the A/D conversion time as follows.

- $AV_{REF} \geq 4.5$ V: 3.0 μs or more and less than 100 μs
- $AV_{REF} \geq 4.0$ V: 4.8 μs or more and less than 100 μs
- $AV_{REF} \geq 2.85$ V: 6.0 μs or more and less than 100 μs
- $AV_{REF} \geq 2.7$ V: 14.0 μs or more and less than 100 μs

Remark f_{XP}: Oscillation frequency of clock to peripheral hardware

Figure 10-2 shows the block diagram of A/D converter.

Figure 10-2. Block Diagram of A/D Converter



Caution In the 78K0S/KA1+, V_{SS} and AV_{SS} are internally connected. Be sure to connect V_{SS} to a stable GND, and stabilize V_{SS} via GND (= 0 V).

10.2 Configuration of A/D Converter

The A/D converter consists of the following hardware.

Table 10-2. Registers of A/D Converter Used on Software

Item	Configuration
Registers	Successive approximation register (SAR) 10-bit A/D conversion result register (ADCR) 8-bit A/D conversion result register (ADCRH) A/D converter mode register (ADM) Analog input channel specification register (ADS) Port mode control register 2 (PMC2) Port mode register 2 (PM2)

(1) ANI0 to ANI3 pins

These are the analog input pins of the 4-channel A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin by the analog input channel specification register (ADS) can be used as input port pins.

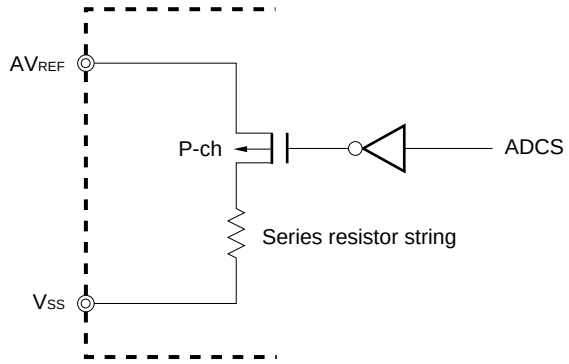
(2) Sample & hold circuit

The sample & hold circuit samples the input signal of the analog input pin selected by the selector when A/D conversion is started, and holds the sampled analog input voltage value during A/D conversion.

(3) Series resistor string

The series resistor string is connected between AV_{REF} and V_{SS}, and generates a voltage to be compared with the analog input signal.

Figure 10-3. Circuit Configuration of Series Resistor String



(4) Voltage comparator

The voltage comparator compares the sampled analog input voltage and the output voltage of the series resistor string.

(5) Successive approximation register (SAR)

This register compares the sampled analog voltage and the voltage of the series resistor string, and converts the result, starting from the most significant bit (MSB).

When the voltage value is converted into a digital value down to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register are transferred to the A/D conversion result register (ADCR).

(6) 10-bit A/D conversion result register (ADCR)

The result of A/D conversion is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the result of A/D conversion in its lower 10 bits (the higher 6 bits are fixed to 0).

(7) 8-bit A/D conversion result register (ADCRH)

The result of A/D conversion is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register holds the result of A/D conversion in its higher 8 bits.

(8) Controller

When A/D conversion has been completed, INTAD is generated.

(9) AV_{REF} pin

This pin inputs an analog power/reference voltage to the A/D converter. When the A/D converter is not used, connect this pin to V_{DD}.

The signal input to ANI0 to ANI3 is converted into a digital signal, based on the voltage applied across AV_{REF} and V_{SS}.

In the standby mode, the current flowing through the series resistor string can be reduced by lowering the voltage input to the AV_{REF} pin to the V_{SS} level.

(10) V_{SS} pin

This is the ground potential pin.

Caution In the 78K0S/KA1+, V_{SS} and AV_{SS} are internally connected. Be sure to connect V_{SS} to a stable GND, and stabilize V_{SS} via GND (= 0 V).

(11) A/D converter mode register (ADM)

This register is used to set the conversion time of the analog input signal to be converted, and to start or stop the conversion operation.

(12) Analog input channel specification register (ADS)

This register is used to specify the port that inputs the analog voltage to be converted into a digital signal.

(13) Port mode control register 2 (PMC2)

This register is used when the P20/ANI0 to P23/ANI3 pins are used as the analog input pins of the A/D converter.

10.3 Registers Used by A/D Converter

The A/D converter uses the following six registers.

- A/D converter mode register (ADM)
- Analog input channel specification register (ADS)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)
- Port mode control register 2 (PMC2)
- Port mode register 2 (PM2)

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

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.

ADM can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset input clears this register to 00H.

Figure 10-4. Format of A/D Converter Mode Register (ADM)

Address: FF80H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	0	FR2	FR1	FR0	0	0	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation
1	Enables conversion operation

FR2	FR1	FR0	Sampling Time[Note 1]	Conversion Time[Note 2]	f_{XP} = 8 MHz		f_{XP} = 10 MHz	
					Sampling Time[Note 1]	Conversion Time[Note 2]	Sampling Time[Note 1]	Conversion Time[Note 2]
0	0	0	$12/f_{XP}$	$36/f_{XP}$	1.5 μs	4.5 μs	1.2 μs	3.6 μs
0	0	1	$24/f_{XP}$	$48/f_{XP}$	3.0 μs	6.0 μs	2.4 μs	4.8 μs
0	1	0	$48/f_{XP}$	$72/f_{XP}$	6.0 μs	9.0 μs	4.8 μs	7.2 μs
0	1	1	$88/f_{XP}$	$112/f_{XP}$	11.0 μs	14.0 μs	8.8 μs	11.2 μs
1	0	0	$24/f_{XP}$	$72/f_{XP}$	3.0 μs	9.0 μs	2.4 μs	7.2 μs
1	0	1	$48/f_{XP}$	$96/f_{XP}$	6.0 μs	12.0 μs	4.8 μs	9.6 μs
1	1	0	$96/f_{XP}$	$144/f_{XP}$	12.0 μs	18.0 μs	9.6 μs	14.4 μs
1	1	1	$176/f_{XP}$	$224/f_{XP}$	22.0 μs	28.0 μs	17.2 μs	22.4 μs

ADCE	Boost reference voltage generator operation control[Note 3]
0	Stops operation of reference voltage generator
1	Enables operation of reference voltage generator

Notes 1. Set the sampling time as follows.

- $AV_{REF} \geq 4.5$ V: 1.0 μs or more
- $AV_{REF} \geq 4.0$ V: 2.4 μs or more
- $AV_{REF} \geq 2.85$ V: 3.0 μs or more
- $AV_{REF} \geq 2.7$ V: 11.0 μs or more

2. Set the A/D conversion time as follows.

- $AV_{REF} \geq 4.5$ V: 3.0 μs or more and less than 100 μs
- $AV_{REF} \geq 4.0$ V: 4.8 μs or more and less than 100 μs
- $AV_{REF} \geq 2.85$ V: 6.0 μs or more and less than 100 μs
- $AV_{REF} \geq 2.7$ V: 14.0 μs or more and less than 100 μs

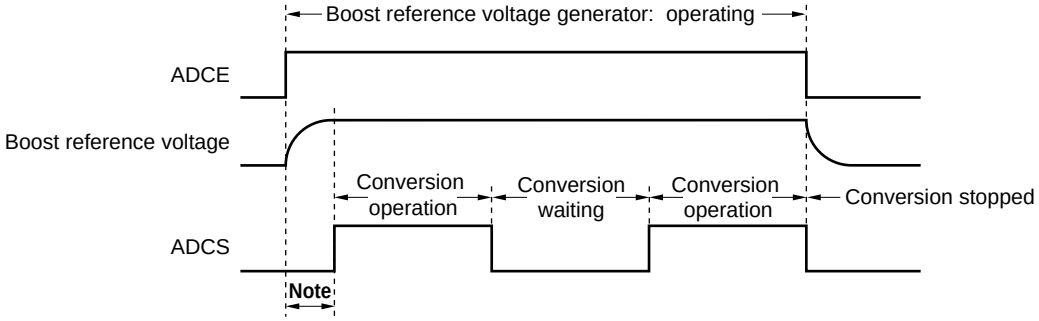
3. A booster circuit is incorporated to realize low-voltage operation. The operation of the circuit that generates the reference voltage for boosting is controlled by ADCE, and it takes 1 μs from operation start to operation stabilization. Therefore, when ADCS is set to 1 after 1 μs or more has elapsed from the time ADCE is set to 1, the conversion result at that time has priority over the first conversion result.

Table 10-3. Settings of ADCS and ADCE

ADCS	ADCE	A/D Conversion Operation
0	0	Stop status (DC power consumption path does not exist)
0	1	Conversion waiting mode (only reference voltage generator consumes power)
1	0	Conversion mode (reference voltage generator operation stopped[Note])
1	1	Conversion mode (reference voltage generator operates)

Note Data of first conversion cannot be used.

Figure 10-5. Timing Chart When Boost Reference Voltage Generator Is Used



Note The time from the rising of the ADCE bit to the rising of the ADCS bit must be 1 μs or longer to stabilize the reference voltage.

Caution A/D conversion must be stopped before rewriting bits FR0 to FR2.

Remark f_{XP}: Oscillation frequency of clock to peripheral hardware

(2) Analog input channel specification register (ADS)

This register specifies the input port of the analog voltage to be A/D converted.

ADS can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset input clears this register to 00H.

Figure 10-6. Format of Analog Input Channel Specification Register (ADS)

Address: FF81H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	0	0	0	ADS1	ADS0

ADS1	ADS0	Analog input channel specification
0	0	ANI0
0	1	ANI1
1	0	ANI2
1	1	ANI3

Caution Be sure to clear bits 2 to 7 of ADS to 0.

(3) 10-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result. The higher six bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register, and is stored in ADCR in order starting from bit 1 of FF19H. FF19H indicates the higher 2 bits of the conversion result, and FF18H indicates the lower 8 bits of the conversion result.

ADCR can be read by a 16-bit memory manipulation instruction.

Reset input makes ADCR undefined.

Figure 10-7. Format of 10-Bit A/D Conversion Result Register (ADCR)

Address: FF18H, FF19H After reset: Undefined R

Symbol	FF19H								FF18H							
ADCR	0	0	0	0	0	0										

Caution When writing to the A/D converter mode register (ADM) and analog input channel specification register (ADS), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM and ADS. Using timing other than the above may cause an incorrect conversion result to be read.

(4) 8-bit A/D conversion result register (ADCRH)

This register is an 8-bit register that stores the A/D conversion result. It stores the higher 8 bits of a 10-bit resolution result.

ADCRH can be read by an 8-bit memory manipulation instruction.

Reset input makes ADCRH undefined.

Figure 10-8. Format of 8-Bit A/D Conversion Result Register (ADCRH)

Address: FF1AH After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
ADCRH								

(5) Port mode control register 2 (PMC2) and port mode register 2 (PM2)

When using the P20/ANI0 to P23/ANI3 pins for analog input, set PMC20 to PMC23 and PM20 to PM23 to 1. At this time, the output latches of P20 to P23 may be 0 or 1.

PMC2 and PM2 are set by a 1-bit or 8-bit memory manipulation instruction.

Reset input clears PMC2 to 00H and sets PM2 to FFH.

Figure 10-9. Format of Port Mode Control Register 2 (PMC2)

Address: FF84H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PMC2	0	0	0	0	PMC23	PMC22	PMC21	PMC20

PMC2n	Operation mode specification (n = 0 to 3)
0	Port mode
1	Alternate-function mode (A/D converter)

Figure 10-10. Format of Port Mode Register 2 (PM2)

Address: FF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	1	1	1	1	PM23	PM22	PM21	PM20

PM2n	Pmn pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution When PMC20 to PMC23 are set to 1, the P20/ANI0 to P23/ANI3 pins cannot be used as port pins.

10.4 A/D Converter Operations

10.4.1 Basic operations of A/D converter

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

<2> Set ADCE to 1 and wait for 1 μs or longer.

<3> Set ADCS to 1 and start the conversion operation.
(<4> to <10> are operations performed by hardware.)

<4> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<5> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the input analog voltage is held until the A/D conversion operation has ended.

<6> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF} by the tap selector.

<7> The voltage difference between the series resistor string voltage tap and analog input is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{REF}, the MSB of SAR remains set to 1. If the analog input is smaller than (1/2) AV_{REF}, the MSB is reset to 0.

<8> Next, bit 8 of SAR is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.

- Bit 9 = 1: (3/4) AV_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

The voltage tap and analog input voltage are compared and bit 8 of SAR is manipulated as follows.

- Analog input voltage ≥ Voltage tap: Bit 8 = 1
- Analog input voltage < Voltage tap: Bit 8 = 0

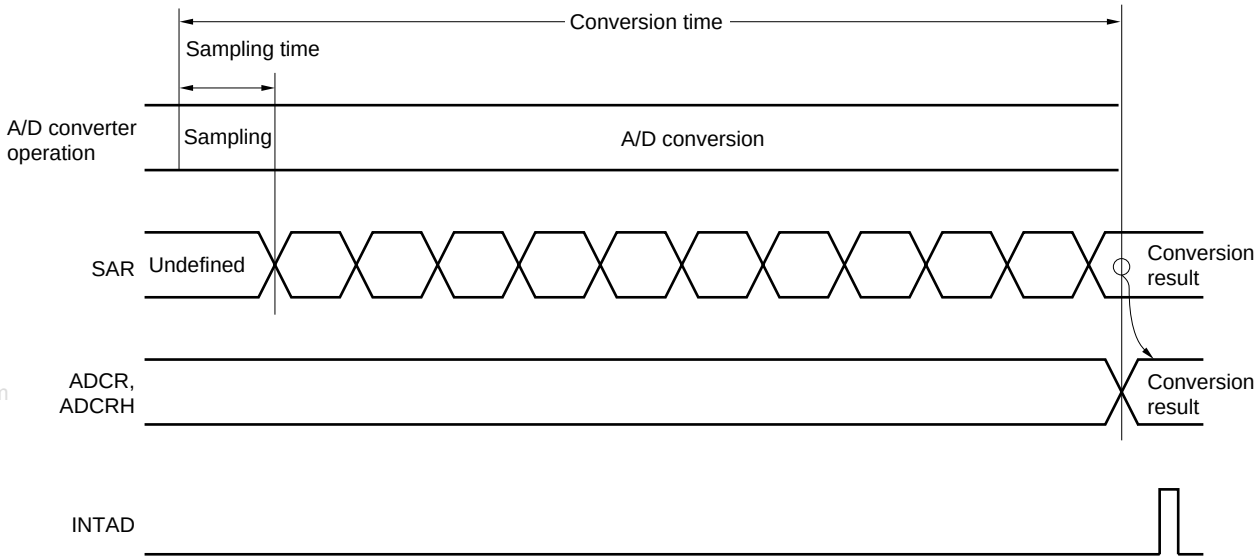
<9> Comparison is continued in this way up to bit 0 of SAR.

<10> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched.
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

<11> Repeat steps <4> to <10>, until ADCS is cleared to 0.
To stop the A/D converter, clear ADCS to 0.
To restart A/D conversion from the status of ADCE = 1, start from <3>. To restart A/D conversion from the status of ADCE = 0, however, start from <2>.

Remark The following two types of A/D conversion result registers can be used.
<1> ADCR (16 bits): Stores a 10-bit A/D conversion value.
<2> ADCRH (8 bits): Stores an 8-bit A/D conversion value.

Figure 10-11. Basic Operation of A/D Converter



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

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

Reset input makes the A/D conversion result register (ADCR, ADCRH) undefined.

10.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI3) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$ADCR = SAR \times 64$$

or

$$(ADCR - 0.5) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < (ADCR + 0.5) \times \frac{AV_{REF}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

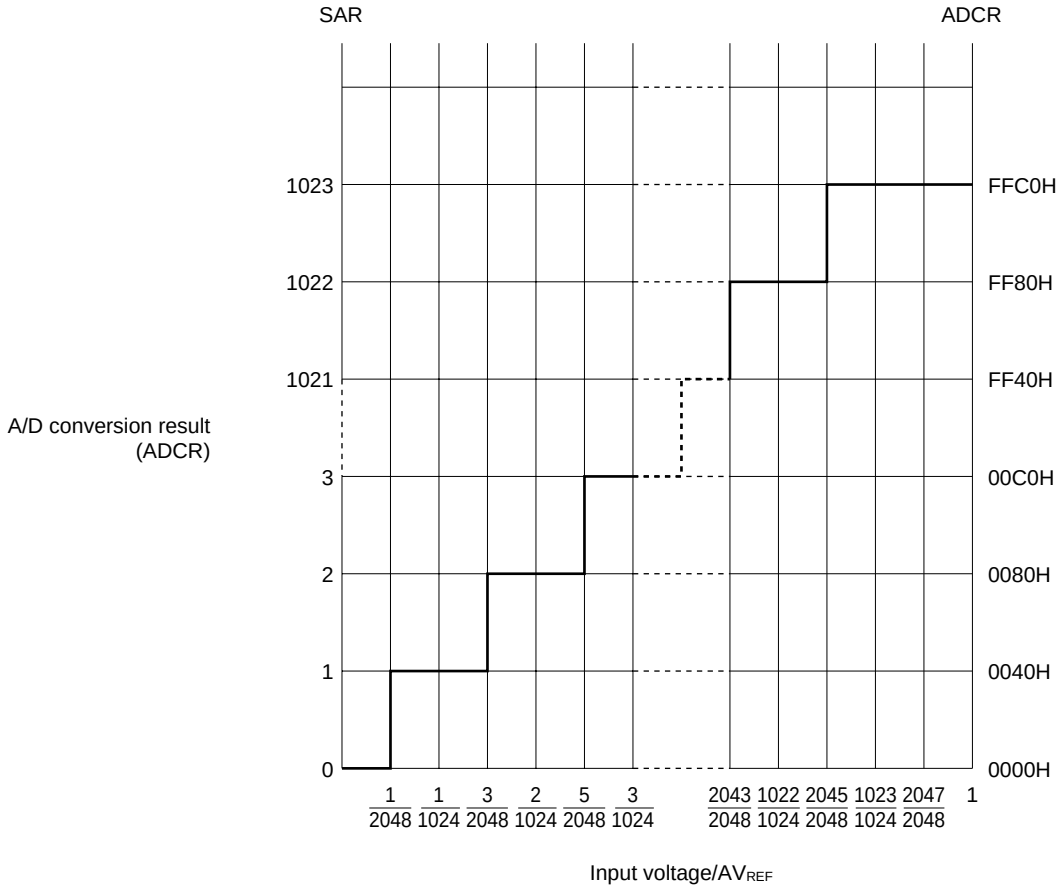
AV_{REF}: AV_{REF} pin voltage

ADCR: 10-bit A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 10-12 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 10-12. Relationship Between Analog Input Voltage and A/D Conversion Result



10.4.3 A/D converter operation mode

The operation mode of the A/D converter is the select mode. One channel of analog input is selected from ANI0 to ANI3 by the analog input channel specification register (ADS) and A/D conversion is executed.

(1) A/D conversion operation

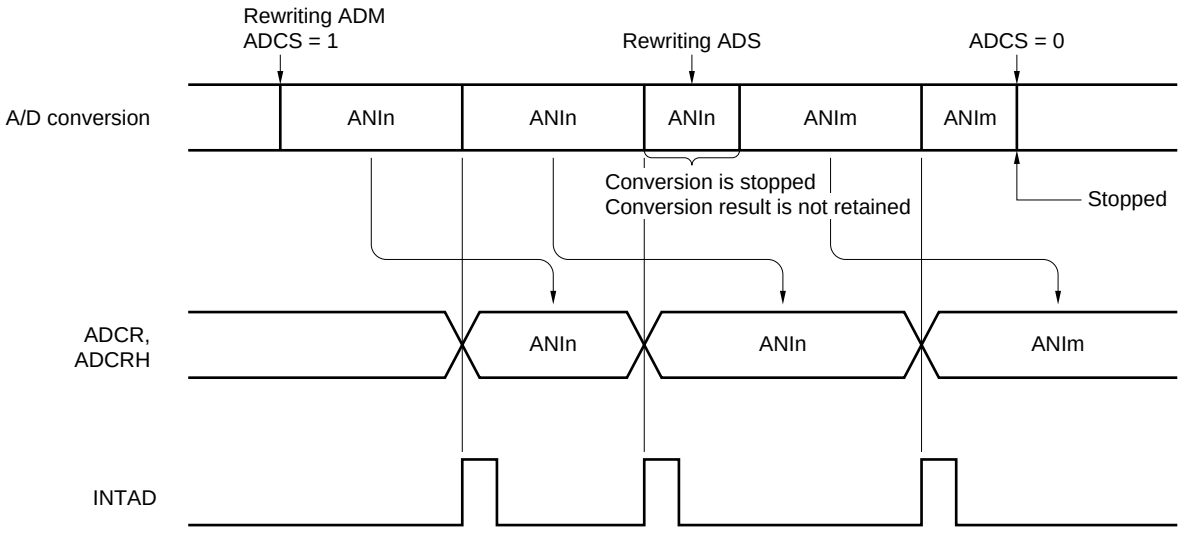
By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1, the A/D conversion operation of the voltage, which is applied to the analog input pin specified by the analog input channel specification register (ADS), is started.

When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR, ADCRH), and an interrupt request signal (INTAD) is generated. Once the A/D conversion has started and when one A/D conversion has been completed, the next A/D conversion operation is immediately started. The A/D conversion operations are repeated until new data is written to ADS.

If ADM or ADS is written during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning.

If 0 is written to ADCS during A/D conversion, A/D conversion is immediately stopped. At this time, the conversion result is undefined.

Figure 10-13. A/D Conversion Operation



Remarks 1. n = 0 to 3

2. m = 0 to 3

The setting method is described below.

<1> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1.
<2> Select the channel and conversion time using bits 1 and 0 (ADS1, ADS0) of the analog input channel specification register (ADS) and bits 5 to 3 (FR2 to FR0) of ADM.
<3> Set bit 7 (ADCS) of ADM to 1.
<4> An interrupt request signal (INTAD) is generated.
<5> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Change the channel>

<6> Change the channel using bits 1 and 0 (ADS1, ADS0) of ADS.
<7> An interrupt request signal (INTAD) is generated.
<8> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Complete A/D conversion>

<9> Clear ADCS to 0.
<10> Clear ADCE to 0.

Cautions 1. Make sure the period of <1> to <3> is 1 μs or more.
2. It is no problem if the order of <1> and <2> is reversed.
3. <1> can be omitted. However, do not use the first conversion result after <3> in this case.
4. The period from <4> to <7> differs from the conversion time set using bits 5 to 3 (FR2 to FR0) of ADM. The period from <6> to <7> is the conversion time set using FR2 to FR0.

10.5 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

$1LSB = 1/2^{10} = 1/1024$
$= 0.098\%FSR$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.

Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.

Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.

Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 10-14. Overall Error

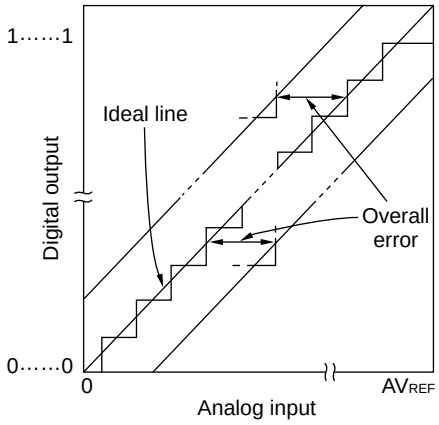
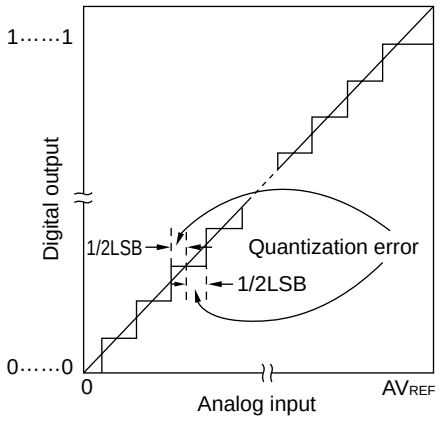


Figure 10-15. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0.....000 to 0.....001.

If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale – 3/2LSB) when the digital output changes from 1.....110 to 1.....111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

(7) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

Figure 10-16. Zero-Scale Error

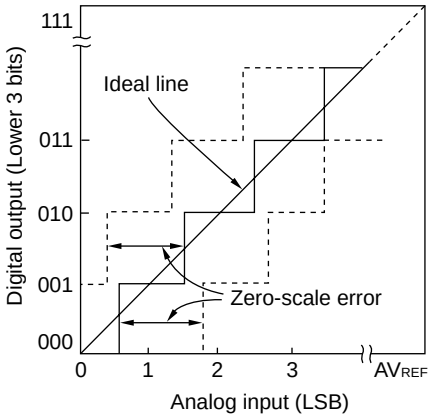


Figure 10-17. Full-Scale Error

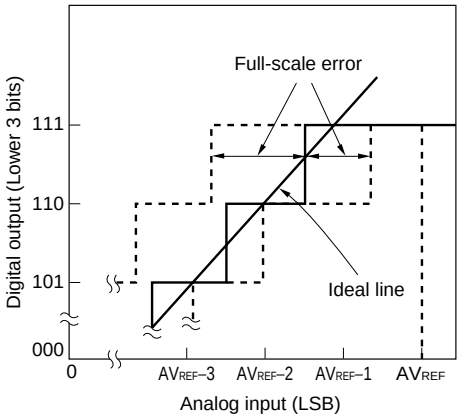


Figure 10-18. Integral Linearity Error

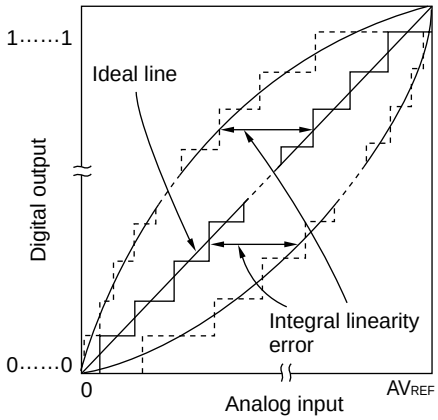
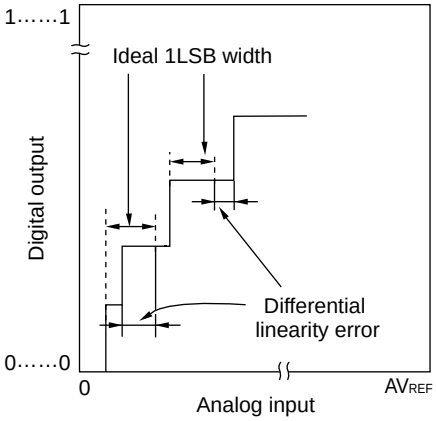


Figure 10-19. Differential Linearity Error



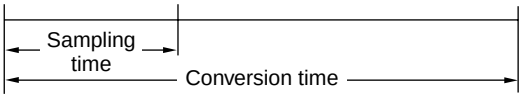
(8) Conversion time

This expresses the time from the start of sampling to when the digital output is obtained.

The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



10.6 Cautions for A/D Converter

(1) Operating current in standby mode

The A/D converter stops operating in the standby mode. At this time, the operating current can be reduced by clearing bit 7 (ADCS) of the A/D converter mode register (ADM) to 0 (see **Figure 10-3**).

(2) Input range of ANI0 to ANI3

Observe the rated range of the ANI0 to ANI3 input voltage. If a voltage of AV_{REF} or higher and V_{SS} or lower (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

(3) Conflicting operations

<1> Conflict between A/D conversion result register (ADCR, ADCRH) write and ADCR, ADCRH read by instruction upon the end of conversion

ADCR, ADCRH read has priority. After the read operation, the new conversion result is written to ADCR, ADCRH.

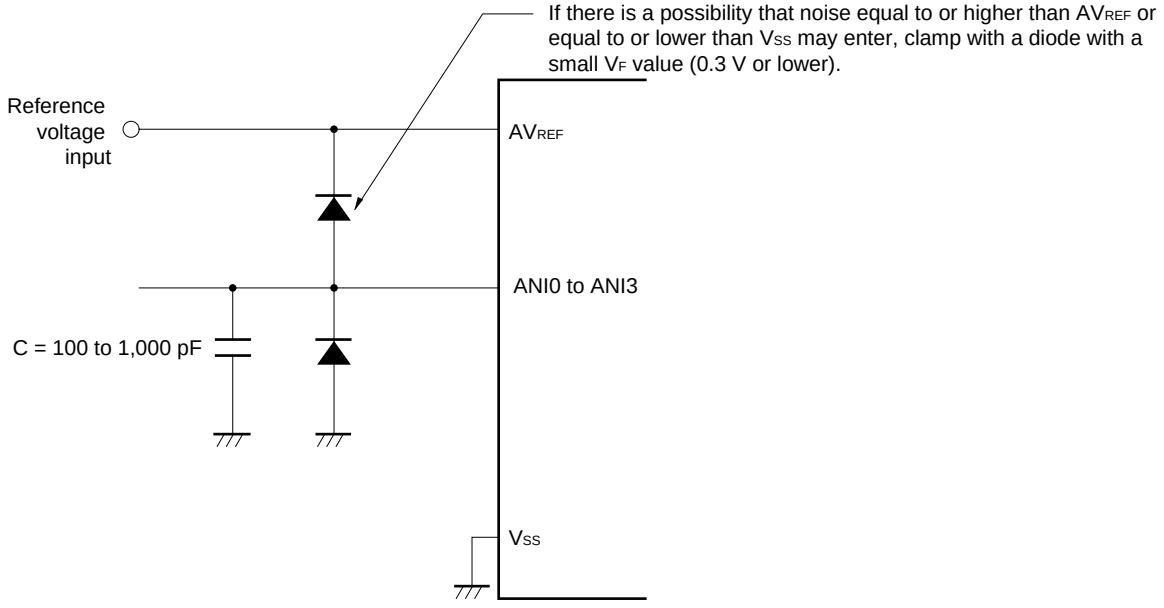
<2> Conflict between ADCR, ADCRH write and A/D converter mode register (ADM) write or analog input channel specification register (ADS) write upon the end of conversion

ADM or ADS write has priority. ADCR, ADCRH write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

(4) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the AV_{REF} pin and pins ANI0 to ANI3. Because the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally, as shown in Figure 10-20, to reduce noise.

Figure 10-20. Analog Input Pin Connection



(5) ANI0/P20 to ANI3/P23

<1> The analog input pins (ANI0 to ANI3) are also used as input port pins (P20 to P23).
When A/D conversion is performed with any of ANI0 to ANI3 selected, do not access port 2 while conversion is in progress; otherwise the conversion resolution may be degraded.

<2> If a digital pulse is applied to the pins adjacent to the pins currently used for A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, do not apply a pulse to the pins adjacent to the pin undergoing A/D conversion.

(6) Input impedance of ANI0 to ANI3 pins

In this A/D converter, the internal sampling capacitor is charged and sampling is performed for approx. one sixth of the conversion time.

Since only the leakage current flows other than during sampling and the current for charging the capacitor also flows during sampling, the input impedance fluctuates and has no meaning.

To perform sufficient sampling, however, it is recommended to make the output impedance of the analog input source 10 kΩ or lower, or attach a capacitor of around 100 pF to the ANI0 to ANI3 pins (see **Figure 10-20**).

(7) AV_{REF} pin input impedance

A series resistor string of several tens of 10 kΩ is connected between the AV_{REF} and V_{SS} pins.

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

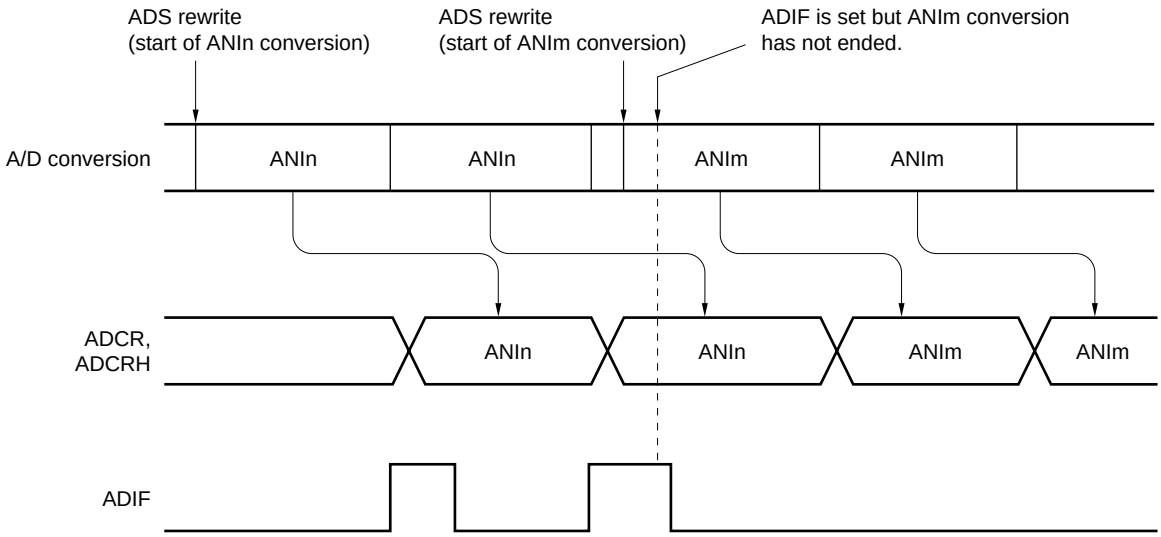
(8) Interrupt request flag (ADIF)

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

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF for the pre-change analog input may be set just before the ADS rewrite. Caution is therefore required since, at this time, when ADIF is read immediately after the ADS rewrite, ADIF is set despite the fact A/D conversion for the post-change analog input has not ended.

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

Figure 10-21. Timing of A/D Conversion End Interrupt Request Generation



Remarks 1. n = 0 to 3

2. m = 0 to 3

(9) Conversion results just after A/D conversion start

The first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μs after the ADCE bit was set to 1, or if the ADCS bit is set to 1 with the ADCE bit = 0. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(10) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to the A/D converter mode register (ADM) and analog input channel specification register (ADS), the contents of ADCR and ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM and ADS. Using a timing other than the above may cause an incorrect conversion result to be read.

(11) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 10-22. Internal Equivalent Circuit of ANIn Pin

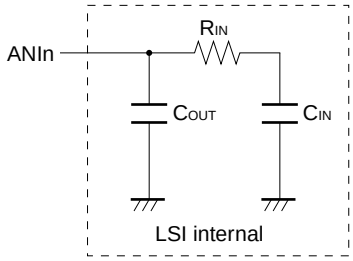


Table 10-4. Resistance and Capacitance Values of Equivalent Circuit

AV_{REF}	R_{IN}	C_{OUT}	C_{IN}
2.7 V	T.B.D.	T.B.D.	T.B.D.
4.5 V	T.B.D.	T.B.D.	T.B.D.

Remarks 1. The resistance and capacitance values shown in Table 10-4 are not guaranteed values.

2. n = 0 to 3

R_{IN}: Input equivalent resistance

C_{IN}: Input equivalent capacitance

C_{OUT}: Package pin capacitance

CHAPTER 11 SERIAL INTERFACE UART6

11.1 Functions of Serial Interface UART6

Serial interface UART6 has the following two modes.

(1) Operation stop mode

This mode is used when serial communication is not executed and can enable a reduction in the power consumption.

For details, see **11.4.1 Operation stop mode**.

(2) Asynchronous serial interface (UART) mode

This mode supports the LIN (Local Interconnect Network)-bus. The functions of this mode are outlined below.

For details, see **11.4.2 Asynchronous serial interface (UART) mode** and **11.4.3 Dedicated baud rate generator**.

- Two-pin configuration TxD6: Transmit data output pin
  RxB6: Receive data input pin
- Data length of communication data can be selected from 7 or 8 bits.
- Dedicated internal 8-bit baud rate generator allowing any baud rate to be set
- Transmission and reception can be performed independently.
- Twelve operating clock inputs selectable
- MSB- or LSB-first communication selectable
- Inverted transmission operation
- Synchronous break field transmission from 13 to 20 bits
- More than 11 bits can be identified for synchronous break field reception (SBF reception flag provided).

Cautions 1. The TxD6 output inversion function inverts only the transmission side and not the reception side. To use this function, the reception side must be ready for reception of inverted data.

2. If clock supply to serial interface UART6 is not stopped (e.g., in the HALT mode), normal operation continues. If clock supply to serial interface UART6 is stopped (e.g., in the STOP mode), each register stops operating, and holds the value immediately before clock supply was stopped. The TxD6 pin also holds the value immediately before clock supply was stopped and outputs it. However, the operation is not guaranteed after clock supply is resumed. Therefore, reset the circuit so that POWER6 = 0, RXE6 = 0, and TXE6 = 0.

3. If data is continuously transmitted, the communication timing from the stop bit to the next start bit is extended two operating clocks of the macro. However, this does not affect the result of communication because the reception side initializes the timing when it has detected a start bit. Do not use the continuous transmission function if the interface is incorporated in LIN.

Remark LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol intended to aid the cost reduction of an automotive network.

LIN communication is single-master communication, and up to 15 slaves can be connected to one master.

The LIN slaves are used to control the switches, actuators, and sensors, and these are connected to the LIN master via the LIN network.

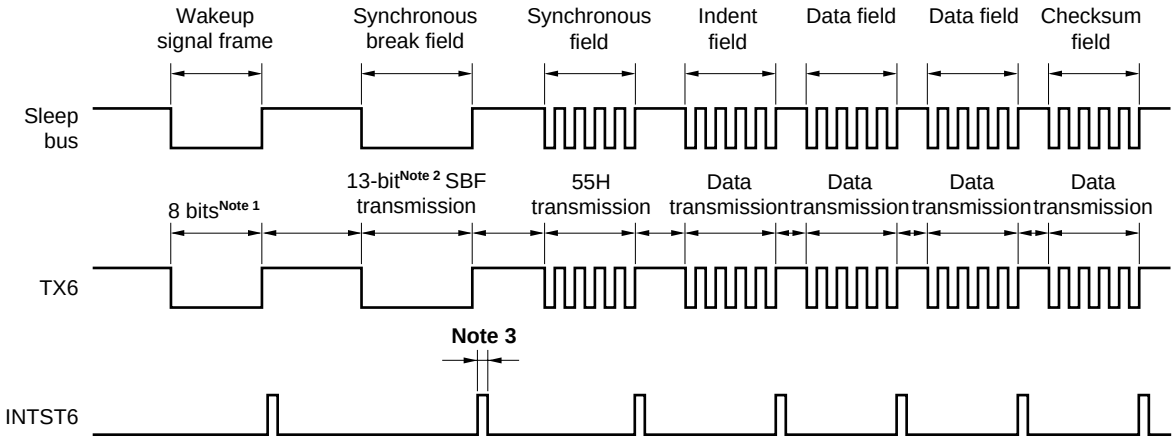
Normally, the LIN master is connected to a network such as CAN (Controller Area Network).

In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.

In the LIN protocol, the master transmits a frame with baud rate information and the slave receives it and corrects the baud rate error. Therefore, communication is possible when the baud rate error in the slave is ±15% or less.

Figures 11-1 and 11-2 outline the transmission and reception operations of LIN.

Figure 11-1. LIN Transmission Operation



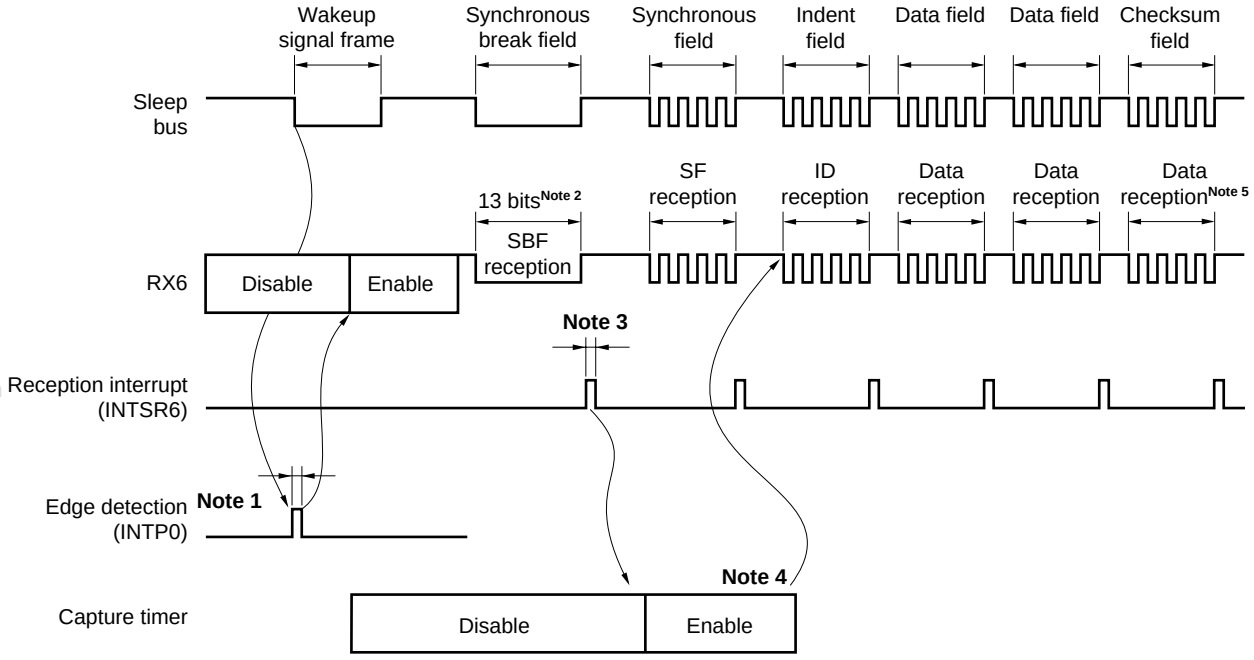
Notes 1. The wakeup signal frame is substituted by 80H transmission in the 8-bit mode.

2. The synchronous break field is output by hardware. The output width is equal to the bit length set by bits 4 to 2 (SBL62 to SBL60) of asynchronous serial interface control register 6 (ASICL6). If the output width needs to be adjusted more accurately, use baud rate generator control register 6 (BRGC6) (see **11.4.2 (h) SBF transmission**).

3. INTST6 is output on completion of each transmission. It is also output when SBF is transmitted.

Remark The interval between each field is controlled by software.

Figure 11-2. LIN Reception Operation



Notes 1. The wakeup signal is detected at the edge of the pin, and enables UART6 and sets the SBF reception mode.

2. Reception continues until the STOP bit is detected. When an SBF with low-level data of 11 bits or more has been detected, it is assumed that SBF reception has been completed correctly, and an interrupt signal is output. If an SBF with low-level data of less than 11 bits has been detected, it is assumed that an SBF reception error has occurred. The interrupt signal is not output and the SBF reception mode is restored.

3. If SBF reception has been completed correctly, an interrupt signal is output. This SBF reception completion interrupt enables the capture timer. Detection of errors OVE6, PE6, and FE6 is suppressed, and error detection processing of UART communication and data transfer of the shift register and RXB6 is not performed. The shift register holds the reset value FFH.

4. Calculate the baud rate error from the bit length of the synchronous field, disable UART6 after SF reception, and then re-set baud rate generator control register 6 (BRGC6).

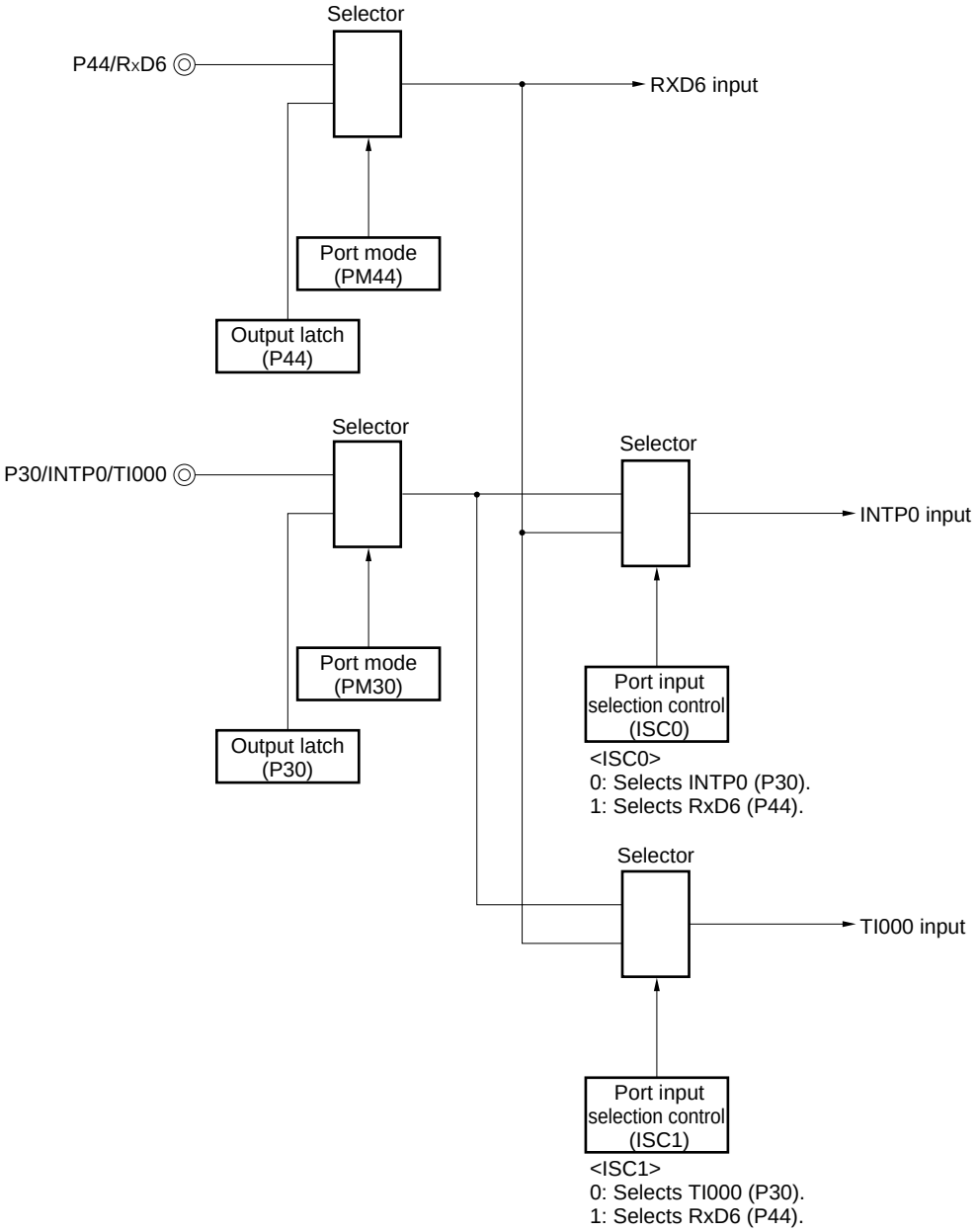
5. Distinguish the checksum field by software. Also perform processing by software to initialize UART6 after reception of the checksum field and to set the SBF reception mode again.

To perform a LIN receive operation, use a configuration like the one shown in Figure 11-3.

The wakeup signal transmitted from the LIN master is received by detecting the edge of the external interrupt (INTP0). The length of the synchronous field transmitted from the LIN master can be measured using the external event capture operation of 16-bit timer/event counter 00, and the baud rate error can be calculated.

The input signal of the reception port input (RxD6) can be input to the external interrupt (INTP0) and 16-bit timer/event counter 00 by port input switch control (ISC0/ISC1), without connecting RxD6 and INTP0/TI000 externally.

Figure 11-3. Port Configuration for LIN Reception Operation



Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (see **Figure 11-11**)

The peripheral functions used in the LIN communication operation are shown below.

<Peripheral functions used>

- External interrupt (INTP0); wakeup signal detection
  Use: Detects the wakeup signal edges and detects start of communication.
- 16-bit timer/event counter 00 (TI000); baud rate error detection
  Use: Detects the baud rate error (measures the TI000 input edge interval in the capture mode) by detecting the synchronous break field (SBF) length and divides it by the number of bits.
- Serial interface UART6

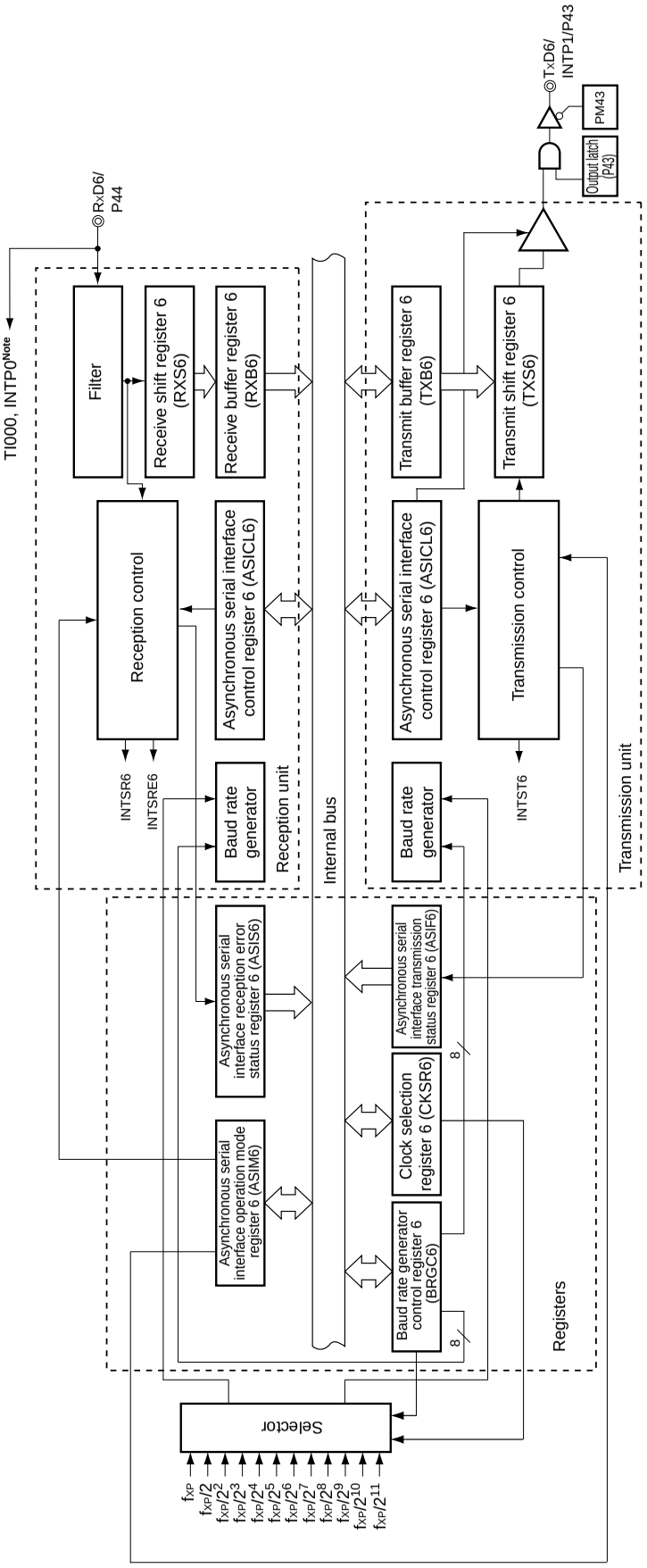
11.2 Configuration of Serial Interface UART6

Serial interface UART6 consists of the following hardware.

Table 11-1. Configuration of Serial Interface UART6

Item	Configuration
Registers	Receive buffer register 6 (RXB6) Receive shift register 6 (RXS6) Transmit buffer register 6 (TXB6) Transmit shift register 6 (TXS6)
Control registers	Asynchronous serial interface operation mode register 6 (ASIM6) Asynchronous serial interface reception error status register 6 (ASIS6) Asynchronous serial interface transmission status register 6 (ASIF6) Clock selection register 6 (CKSR6) Baud rate generator control register 6 (BRGC6) Asynchronous serial interface control register 6 (ASICL6) Input switch control register (ISC) Port mode register 4 (PM4) Port register 4 (P4)

Figure 11-4. Block Diagram of Serial Interface UART6



Note Selectable with input switch control register (ISC).

(1) Receive buffer register 6 (RXB6)

This 8-bit register stores parallel data converted by receive shift register 6 (RXS6).

Each time 1 byte of data has been received, new receive data is transferred to this register from receive shift register 6 (RXS6). If the data length is set to 7 bits, data is transferred as follows.

- In LSB-first reception, the receive data is transferred to bits 0 to 6 of RXB6 and the MSB of RXB6 is always 0.
- In MSB-first reception, the receive data is transferred to bits 1 to 7 of RXB6 and the LSB of RXB6 is always 0.

If an overrun error (OVE6) occurs, the receive data is not transferred to RXB6.

RXB6 can be read by an 8-bit memory manipulation instruction. No data can be written to this register.

Reset input sets this register to FFH.

(2) Receive shift register 6 (RXS6)

This register converts the serial data input to the RxD6 pin into parallel data.

RXS6 cannot be directly manipulated by a program.

(3) Transmit buffer register 6 (TXB6)

This buffer register is used to set transmit data. Transmission is started when data is written to TXB6.

This register can be read or written by an 8-bit memory manipulation instruction.

Reset input sets this register to FFH.

Cautions 1. Do not write data to TXB6 when bit 1 (TXBF6) of asynchronous serial interface transmission status register 6 (ASIF6) is 1.

2. Do not refresh (write the same value to) TXB6 by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of asynchronous serial interface operation mode register 6 (ASIM6) are 1 or when bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 are 1).

(4) Transmit shift register 6 (TXS6)

This register transmits the data transferred from TXB6 from the TxD6 pin as serial data. Data is transferred from TXB6 immediately after TXB6 is written for the first transmission, or immediately before INTST6 occurs after one frame was transmitted for continuous transmission. Data is transferred from TXB6 and transmitted from the TxD6 pin at the falling edge of the base clock.

TXS6 cannot be directly manipulated by a program.

11.3 Registers Controlling Serial Interface UART6

Serial interface UART6 is controlled by the following nine registers.

- Asynchronous serial interface operation mode register 6 (ASIM6)
- Asynchronous serial interface reception error status register 6 (ASIS6)
- Asynchronous serial interface transmission status register 6 (ASIF6)
- Clock selection register 6 (CKSR6)
- Baud rate generator control register 6 (BRGC6)
- Asynchronous serial interface control register 6 (ASICL6)
- Input switch control register (ISC)
- Port mode register 4 (PM4)
- Port register 4 (P4)

(1) Asynchronous serial interface operation mode register 6 (ASIM6)

This 8-bit register controls the serial communication operations of serial interface UART6.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets this register to 01H.

Remark ASIM6 can be refreshed (the same value is written) by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1 or bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1).

Figure 11-5. Format of Asynchronous Serial Interface Operation Mode Register 6 (ASIM6) (1/2)

Address: FF90H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6

POWER6	Enabling/disabling operation of internal operation clock
0[Note 1]	Disable operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit[Note 2].
1[Note 3]	Enable operation of the internal operation clock

TXE6	Enabling/disabling transmission
0	Disable transmission (synchronously reset the transmission circuit).
1	Enable transmission

Notes 1. The output of the TxD6 pin goes high and the input from the RxD6 pin is fixed to the high level when POWER6 = 0.

2. Asynchronous serial interface reception error status register 6 (ASIS6), asynchronous serial interface transmission status register 6 (ASIF6), bit 7 (SBRF6) and bit 6 (SBRT6) of asynchronous serial interface control register 6 (ASICL6), and receive buffer register 6 (RXB6) are reset.

3. Operation of the 8-bit counter output is enabled at the second base clock after 1 is written to the POWER6 bit.

Figure 11-5. Format of Asynchronous Serial Interface Operation Mode Register 6 (ASIM6) (2/2)

RXE6	Enabling/disabling reception
0	Disable reception (synchronously reset the reception circuit).
1	Enable reception

PS61	PS60	Transmission operation	Reception operation
0	0	Parity bit not output.	Reception without parity
0	1	Output 0 parity.	Reception as 0 parity[Note]
1	0	Output odd parity.	Judge as odd parity.
1	1	Output even parity.	Judge as even parity.

CL6	Specification of character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL6	Specification of number of stop bits of transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

ISRM6	Enabling/disabling occurrence of reception completion interrupt in case of error
0	“INTSRE6” occurs in case of error (at this time, INTSR6 does not occur).
1	“INTSR6” occurs in case of error (at this time, INTSRE6 does not occur).

Note If “reception as 0 parity” is selected, the parity is not judged. Therefore, bit 2 (PE6) of asynchronous serial interface reception error status register 6 (ASIS6) is not set and the error interrupt does not occur.

Cautions
1. **At startup, set POWER6 to 1 and then set TXE6 to 1. To stop the operation, clear TXE6 to 0, and then clear POWER6 to 0.**
2. **At startup, set POWER6 to 1 and then set RXE6 to 1. To stop the operation, clear RXE6 to 0, and then clear POWER6 to 0.**
3. **Set POWER6 to 1 and then set RXE6 to 1 while a high level is input to the RxD6 pin. If POWER6 is set to 1 and RXE6 is set to 1 while a low level is input, reception is started.**
4. **Clear the TXE6 and RXE6 bits to 0 before rewriting the PS61, PS60, and CL6 bits.**
5. **Fix the PS61 and PS60 bits to 0 when mounting the device on LIN.**
6. **Make sure that TXE6 = 0 when rewriting the SL6 bit. Reception is always performed with “the number of stop bits = 1”, and therefore, is not affected by the set value of the SL6 bit.**
7. **Make sure that RXE6 = 0 when rewriting the ISRM6 bit.**

(2) Asynchronous serial interface reception error status register 6 (ASIS6)

This register indicates an error status on completion of reception by serial interface UART6. It includes three error flag bits (PE6, FE6, OVE6).

This register is read-only by an 8-bit memory manipulation instruction.

Reset input clears this register to 00H if bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 0. 00H is read when this register is read.

Figure 11-6. Format of Asynchronous Serial Interface Reception Error Status Register 6 (ASIS6)

Address: FF93H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS6	0	0	0	0	0	PE6	FE6	OVE6

PE6	Status flag indicating parity error
0	If POWER6 = 0 and RXE6 = 0, or if ASIS6 register is read
1	If the parity of transmit data does not match the parity bit on completion of reception

FE6	Status flag indicating framing error
0	If POWER6 = 0 and RXE6 = 0, or if ASIS6 register is read
1	If the stop bit is not detected on completion of reception

OVE6	Status flag indicating overrun error
0	If POWER6 = 0 and RXE6 = 0, or if ASIS6 register is read
1	If receive data is set to the RXB register and the next reception operation is completed before the data is read.

Cautions 1. The operation of the PE6 bit differs depending on the set values of the PS61 and PS60 bits of asynchronous serial interface operation mode register 6 (ASIM6).

2. The first bit of the receive data is checked as the stop bit, regardless of the number of stop bits.

3. If an overrun error occurs, the next receive data is not written to receive buffer register 6 (RXB6) but discarded.

(3) Asynchronous serial interface transmission status register 6 (ASIF6)

This register indicates the status of transmission by serial interface UART6. It includes two status flag bits (TXBF6 and TXSF6).

Transmission can be continued without disruption even during an interrupt period, by writing the next data to the TXB6 register after data has been transferred from the TXB6 register to the TXS6 register.

This register is read-only by an 8-bit memory manipulation instruction.

Reset input clears this register to 00H if bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 0.

Figure 11-7. Format of Asynchronous Serial Interface Transmission Status Register 6 (ASIF6)

Address: FF95H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIF6	0	0	0	0	0	0	TXBF6	TXSF6

TXBF6	Transmit buffer data flag
0	If POWER6 = 0 or TXE6 = 0, or if data is transferred to transmit shift register 6 (TXS6)
1	If data is written to transmit buffer register 6 (TXB6) (if data exists in TXB6)

TXSF6	Transmit shift register data flag
0	If POWER6 = 0 or TXE6 = 0, or if the next data is not transferred from transmit buffer register 6 (TXB6) after completion of transfer
1	If data is transferred from transmit buffer register 6 (TXB6) (if data transmission is in progress)

Cautions 1. To transmit data continuously, write the first transmit data (first byte) to the TXB6 register. Be sure to check that the TXBF6 flag is "0". If so, write the next transmit data (second byte) to the TXB6 register. If data is written to the TXB6 register while the TXBF6 flag is "1", the transmit data cannot be guaranteed.

2. To initialize the transmission unit upon completion of continuous transmission, be sure to check that the TXSF6 flag is "0" after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF6 flag is "1", the transmit data cannot be guaranteed.

(4) Clock selection register 6 (CKSR6)

This register selects the base clock of serial interface UART6.

CKSR6 can be set by an 8-bit memory manipulation instruction.

Reset input clears this register to 00H.

Remark CKSR6 can be refreshed (the same value is written) by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1 or bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1).

Figure 11-8. Format of Clock Selection Register 6 (CKSR6)

Address: FF96H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKSR6	0	0	0	0	TPS63	TPS62	TPS61	TPS60

TPS63	TPS62	TPS61	TPS60	Base clock (f_{XCLK6}) selection
0	0	0	0	f_{XP} (10 MHz)
0	0	0	1	$f_{XP}/2$ (5 MHz)
0	0	1	0	$f_{XP}/2^2$ (2.5 MHz)
0	0	1	1	$f_{XP}/2^3$ (1.25 MHz)
0	1	0	0	$f_{XP}/2^4$ (625 kHz)
0	1	0	1	$f_{XP}/2^5$ (312.5 kHz)
0	1	1	0	$f_{XP}/2^6$ (156.25 kHz)
0	1	1	1	$f_{XP}/2^7$ (78.13 kHz)
1	0	0	0	$f_{XP}/2^8$ (39.06 kHz)
1	0	0	1	$f_{XP}/2^9$ (19.53 kHz)
1	0	1	0	$f_{XP}/2^{10}$ (9.77 kHz)
1	0	1	1	$f_{XP}/2^{11}$ (4.89 kHz)
Other than above				Setting prohibited

Caution Make sure POWER6 = 0 when rewriting TPS63 to TPS60.

Remarks 1. Figures in parentheses are for operation with f_{XP} = 10 MHz

2. f_{XP}: Oscillation frequency of clock to peripheral hardware

(5) Baud rate generator control register 6 (BRGC6)

This register sets the division value of the 8-bit counter of serial interface UART6.

BRGC6 can be set by an 8-bit memory manipulation instruction.

Reset input sets this register to FFH.

Remark BRGC6 can be refreshed (the same value is written) by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1 or bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1).

Figure 11-9. Format of Baud Rate Generator Control Register 6 (BRGC6)

Address: FF97H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC6	MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60

MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60	k	Output clock selection of 8-bit counter
0	0	0	0	0	×	×	×	×	Setting prohibited
0	0	0	0	1	0	0	0	8	$f_{XCLK6}/8$
0	0	0	0	1	0	0	1	9	$f_{XCLK6}/9$
0	0	0	0	1	0	1	0	10	$f_{XCLK6}/10$
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	0	252	$f_{XCLK6}/252$
1	1	1	1	1	1	0	1	253	$f_{XCLK6}/253$
1	1	1	1	1	1	1	0	254	$f_{XCLK6}/254$
1	1	1	1	1	1	1	1	255	$f_{XCLK6}/255$

Cautions 1. Make sure that bit 6 (TXE6) and bit 5 (RXE6) of the ASIM6 register = 0 when rewriting the MDL67 to MDL60 bits.

2. The baud rate is the output clock of the 8-bit counter divided by 2.

Remarks 1. f_{XCLK6}: Frequency of base clock selected by the TPS63 to TPS60 bits of CKSR6 register

2. k: Value set by MDL67 to MDL60 bits (k = 8, 9, 10, ..., 255)

3. ×: Don't care

(6) Asynchronous serial interface control register 6 (ASICL6)

This register controls the serial communication operations of serial interface UART6.

ASICL6 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets this register to 16H.

Caution ASICL6 can be refreshed (the same value is written) by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1 or bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1). Note, however, that communication is started by the refresh operation because bit 6 (SBRT6) of ASICL6 is cleared to 0 when communication is completed (when an interrupt signal is generated).

Figure 11-10. Format of Asynchronous Serial Interface Control Register 6 (ASICL6) (1/2)

Address: FF98H After reset: 16H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	1	0
ASICL6	SBRF6	SBRT6	SBTT6	SBL62	SBL61	SBL60	DIR6	TXDLV6

SBRF6	SBF reception status flag
0	If POWER6 = 0 and RXE6 = 0 or if SBF reception has been completed correctly
1	SBF reception in progress

SBRT6	SBF reception trigger
0	–
1	SBF reception trigger

SBTT6	SBF transmission trigger
0	–
1	SBF transmission trigger

Note Bit 7 is read-only.

Figure 11-10. Format of Asynchronous Serial Interface Control Register 6 (ASICL6) (2/2)

SBL62	SBL61	SBL60	SBF transmission output width control
1	0	1	SBF is output with 13-bit length.
1	1	0	SBF is output with 14-bit length.
1	1	1	SBF is output with 15-bit length.
0	0	0	SBF is output with 16-bit length.
0	0	1	SBF is output with 17-bit length.
0	1	0	SBF is output with 18-bit length.
0	1	1	SBF is output with 19-bit length.
1	0	0	SBF is output with 20-bit length.

DIR6	Specification of first bit
0	MSB
1	LSB

TXDLV6	Enabling/disabling inverting TxD6 output
0	Normal output of TxD6
1	Inverted output of TxD6

Cautions 1. In the case of an SBF reception error, return the mode to the SBF reception mode and hold the status of the SBRF6 flag.

2. Before setting the SBRT6 bit, make sure that bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1.

3. The read value of the SBRT6 bit is always 0. SBRT6 is automatically cleared to 0 after SBF reception has been correctly completed.

4. Before setting the SBTT6 bit to 1, make sure that bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1.

5. The read value of the SBTT6 bit is always 0. SBTT6 is automatically cleared to 0 at the end of SBF transmission.

6. Before rewriting the DIR6 and TXDLV6 bits, clear the TXE6 and RXE6 bits to 0.

(7) Input switch control register (ISC)

The input switch control register (ISC) is used to receive a status signal transmitted from the master during LIN (Local Interconnect Network) reception. The input signal is switched by setting ISC.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset input clears this register to 00H.

Figure 11-11. Format of Input Switch Control Register (ISC)

Address: FF8CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	TI000 input source selection
0	TI000 (P30)
1	RxD6 (P44)

ISC0	INTP0 input source selection
0	INTP0 (P30)
1	RxD6 (P44)

(8) Port mode register 4 (PM4)

This register sets port 4 input/output in 1-bit units.

When using the P43/TxD6/INTP1 pin for serial interface data output, clear PM43 to 0 and set the output latch of P43 to 1.

When using the P44/RxD6 pin for serial interface data input, set PM44 to 1. The output latch of P44 at this time may be 0 or 1.

PM4 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets this register to FFH.

Figure 11-12. Format of Port Mode Register 4 (PM4)

Address: FF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	1	1	PM45	PM44	PM43	PM42	PM41	PM40

PM4n	P4n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

11.4 Operation of Serial Interface UART6

Serial interface UART6 has the following two modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode

11.4.1 Operation stop mode

In this mode, serial communication cannot be executed; therefore, the power consumption can be reduced. In addition, the pins can be used as ordinary port pins in this mode. To set the operation stop mode, clear bits 7, 6, and 5 (POWER6, TXE6, and RXE6) of ASIM6 to 0.

(1) Register used

The operation stop mode is set by asynchronous serial interface operation mode register 6 (ASIM6).

ASIM6 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets this register to 01H.

Address: FF90H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6

POWER6	Enabling/disabling operation of internal operation clock
0[Note 1]	Disable operation of the internal operation clock (fix the clock to low level) and asynchronously reset the internal circuit[Note 2].

TXE6	Enabling/disabling transmission
0	Disable transmission operation (synchronously reset the transmission circuit).

RXE6	Enabling/disabling reception
0	Disable reception (synchronously reset the reception circuit).

Notes 1. The output of the TxD6 pin goes high and the input from the RxD6 pin is fixed to high level when POWER6 = 0.

2. Asynchronous serial interface reception error status register 6 (ASIS6), asynchronous serial interface transmission status register 6 (ASIF6), bit 7 (SBRF6) and bit 6 (SBRT6) of asynchronous serial interface control register 6 (ASICL6), and receive buffer register 6 (RXB6) are reset.

Caution Clear POWER6 to 0 after clearing TXE6 and RXE6 to 0 to set the operation stop mode. To start the operation, set POWER6 to 1, and then set TXE6 and RXE6 to 1.

Remark To use the RxD6/P44 and TxD6/INTP1/P43 pins as general-purpose port pins, see **CHAPTER 4 PORT FUNCTIONS**.

11.4.2 Asynchronous serial interface (UART) mode

In this mode, data of 1 byte is transmitted/received following a start bit, and a full-duplex operation can be performed.

A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

(1) Registers used

- Asynchronous serial interface operation mode register 6 (ASIM6)
- Asynchronous serial interface reception error status register 6 (ASIS6)
- Asynchronous serial interface transmission status register 6 (ASIF6)
- Clock selection register 6 (CKSR6)
- Baud rate generator control register 6 (BRGC6)
- Asynchronous serial interface control register 6 (ASICL6)
- Input switch control register (ISC)
- Port mode register 4 (PM4)
- Port register 4 (P4)

The basic procedure of setting an operation in the UART mode is as follows.

<1> Set the CKSR6 register (see **Figure 11-8**).
<2> Set the BRGC6 register (see **Figure 11-9**).
<3> Set bits 0 to 4 (ISRM6, SL6, CL6, PS60, PS61) of the ASIM6 register (see **Figure 11-5**).
<4> Set bits 0 and 1 (TXDLV6, DIR6) of the ASICL6 register (see **Figure 11-10**).
<5> Set bit 7 (POWER6) of the ASIM6 register to 1.
<6> Set bit 6 (TXE6) of the ASIM6 register to 1. → Transmission is enabled.
Set bit 5 (RXE6) of the ASIM6 register to 1. → Reception is enabled.
<7> Write data to transmit buffer register 6 (TXB6). → Data transmission is started.

Caution Take relationship with the other party of communication into consideration when setting the port mode register and port register.

The relationship between the register settings and pins is shown below.

Table 11-2. Relationship Between Register Settings and Pins

POWER6	TXE6	RXE6	PM43	P43	PM44	P44	UART6 Operation	Pin Function	
								TxD6/INTP1/P43	RxD6/P44
0	0	0	×Note	×Note	×Note	×Note	Stop	P43	P44
1	0	1	×Note	×Note	1	×	Reception	P43	RxD6
	1	0	0	1	×Note	×Note	Transmission	TxD6	P44
	1	1	0	1	1	×	Transmission/ reception	TxD6	RxD6

Note Can be set as port function.

Remark ×: don't care
POWER6: Bit 7 of asynchronous serial interface operation mode register 6 (ASIM6)
TXE6: Bit 6 of ASIM6
RXE6: Bit 5 of ASIM6
PM4×: Port mode register
P4×: Port output latch

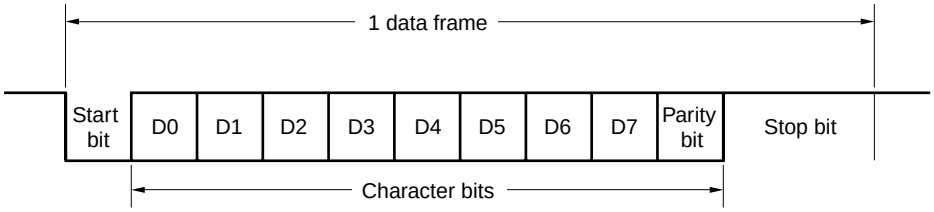
(2) Communication operation

(a) Format and waveform example of normal transmit/receive data

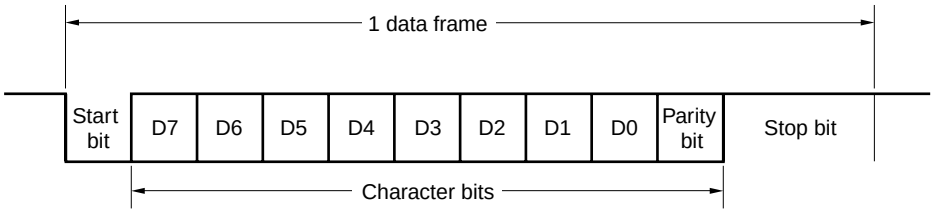
Figures 11-13 and 11-14 show the format and waveform example of the normal transmit/receive data.

Figure 11-13. Format of Normal UART Transmit/Receive Data

1. LSB-first transmission/reception



2. MSB-first transmission/reception



One data frame consists of the following bits.

- Start bit ... 1 bit
- Character bits ... 7 or 8 bits
- Parity bit ... Even parity, odd parity, 0 parity, or no parity
- Stop bit ... 1 or 2 bits

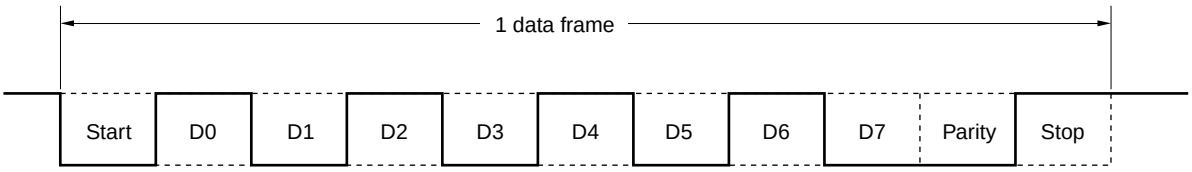
The character bit length, parity, and stop bit length in one data frame are specified by asynchronous serial interface operation mode register 6 (ASIM6).

Whether data is communicated with the LSB or MSB first is specified by bit 1 (DIR6) of asynchronous serial interface control register 6 (ASICL6).

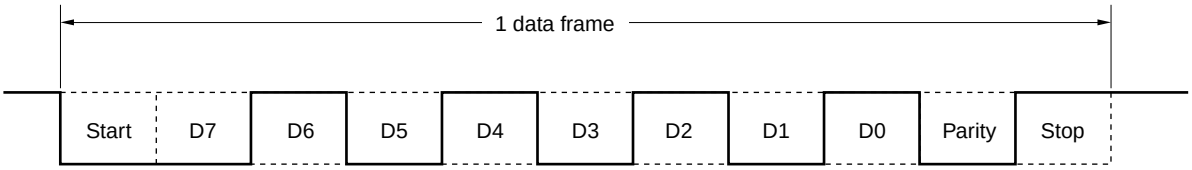
Whether the TxD6 pin outputs normal or inverted data is specified by bit 0 (TXDLV6) of ASICL6.

Figure 11-14. Example of Normal UART Transmit/Receive Data Waveform

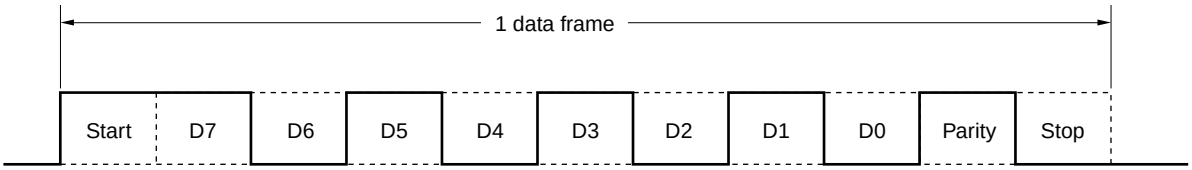
1. Data length: 8 bits, LSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H



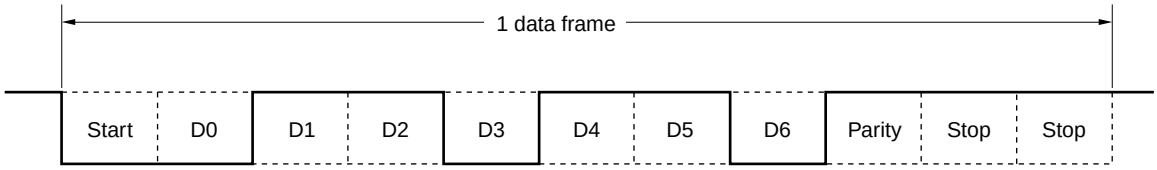
2. Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H



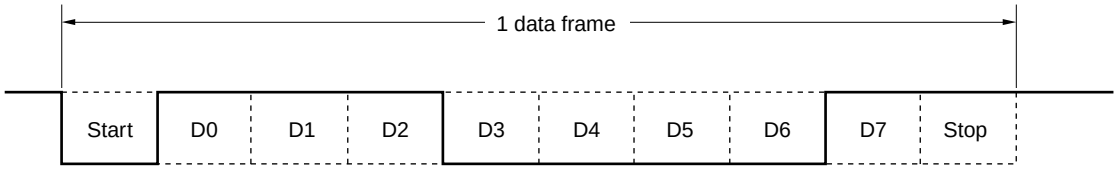
3. Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H, TxD6 pin inverted output



4. Data length: 7 bits, LSB first, Parity: Odd parity, Stop bit: 2 bits, Communication data: 36H



5. Data length: 8 bits, LSB first, Parity: None, Stop bit: 1 bit, Communication data: 87H



(b) Parity types and operation

The parity bit is used to detect a bit error in communication data. Usually, the same type of parity bit is used on both the transmission and reception sides. With even parity and odd parity, a 1-bit (odd number) error can be detected. With zero parity and no parity, an error cannot be detected.

Caution Fix the PS61 and PS60 bits to 0 when the device is incorporated in LIN.

(i) Even parity

- Transmission

  Transmit data, including the parity bit, is controlled so that the number of bits that are “1” is even.
  The value of the parity bit is as follows.

  If transmit data has an odd number of bits that are “1”: 1
  If transmit data has an even number of bits that are “1”: 0

- Reception

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

(ii) Odd parity

- Transmission

  Unlike even parity, transmit data, including the parity bit, is controlled so that the number of bits that are “1” is odd.

  If transmit data has an odd number of bits that are “1”: 0
  If transmit data has an even number of bits that are “1”: 1

- Reception

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

(iii) 0 parity

The parity bit is cleared to 0 when data is transmitted, regardless of the transmit data.
The parity bit is not detected when the data is received. Therefore, a parity error does not occur regardless of whether the parity bit is “0” or “1”.

(iv) No parity

No parity bit is appended to the transmit data.
Reception is performed assuming that there is no parity bit when data is received. Because there is no parity bit, a parity error does not occur.

(c) Normal transmission

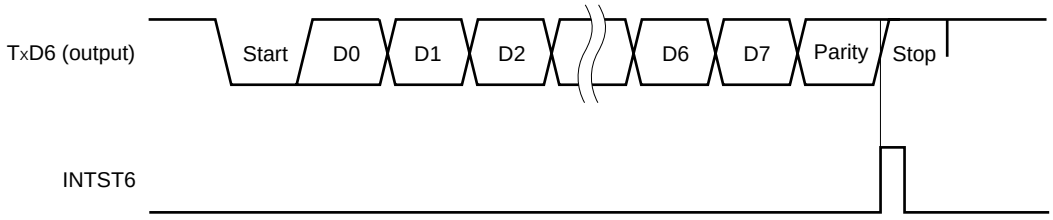
The TxD6 pin outputs a high level when bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is set to 1. If bit 6 (TXE6) of ASIM6 is then set to 1, transmission is enabled. Transmission can be started by writing transmit data to transmit buffer register 6 (TXB6). The start bit, parity bit, and stop bit are automatically appended to the data.

When transmission is started, the data in TXB6 is transferred to transmit shift register 6 (TXS6). After that, the data is sequentially output from TXS6 to the TxD6 pin. When transmission is completed, the parity and stop bits set by ASIM6 are appended and a transmission completion interrupt request (INTST6) is generated. Transmission is stopped until the data to be transmitted next is written to TXB6.

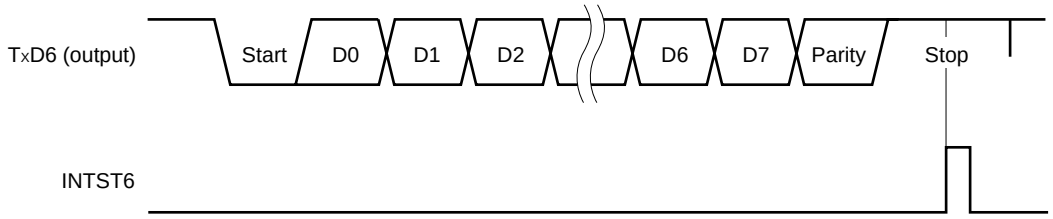
Figure 11-15 shows the timing of the transmission completion interrupt request (INTST6). This interrupt occurs as soon as the last stop bit has been output.

Figure 11-15. Normal Transmission Completion Interrupt Request Timing

1. Stop bit length: 1



2. Stop bit length: 2



(d) Continuous transmission

The next transmit data can be written to transmit buffer register 6 (TXB6) as soon as transmit shift register 6 (TXS6) has started its shift operation. Consequently, even while the INTST6 interrupt is being serviced after transmission of one data frame, data can be continuously transmitted and an efficient communication rate can be realized. In addition, the TXB6 register can be efficiently written twice (2 bytes) without having to wait for the transmission time of one data frame, by reading bit 0 (TXSF6) of asynchronous serial interface transmission status register 6 (ASIF6) when the transmission completion interrupt has occurred.

To transmit data continuously, be sure to reference the ASIF6 register to check the transmission status and whether the TXB6 register can be written, and then write the data.

Cautions 1. The TXBF6 and TXSF6 flags of the ASIS register change from "10" to "11", and to "01" during continuous transmission. To check the status, therefore, do not use a combination of the TXBF6 and TXSF6 flags for judgment. Read only the TXBF6 flag when executing continuous transmission.

2. When the device is incorporated in a LIN, the continuous transmission function cannot be used. Make sure that asynchronous serial interface transmission status register 6 (ASIF6) is 00H before writing transmit data to transmit buffer register 6 (TXB6).

TXBF6	Writing to TXB6 Register
0	Writing enabled
1	Writing disabled

Caution To transmit data continuously, write the first transmit data (first byte) to the TXB6 register. Be sure to check that the TXBF6 flag is "0". If so, write the next transmit data (second byte) to the TXB6 register. If data is written to the TXB6 register while the TXBF6 flag is "1", the transmit data cannot be guaranteed.

The communication status can be checked using the TXSF6 flag.

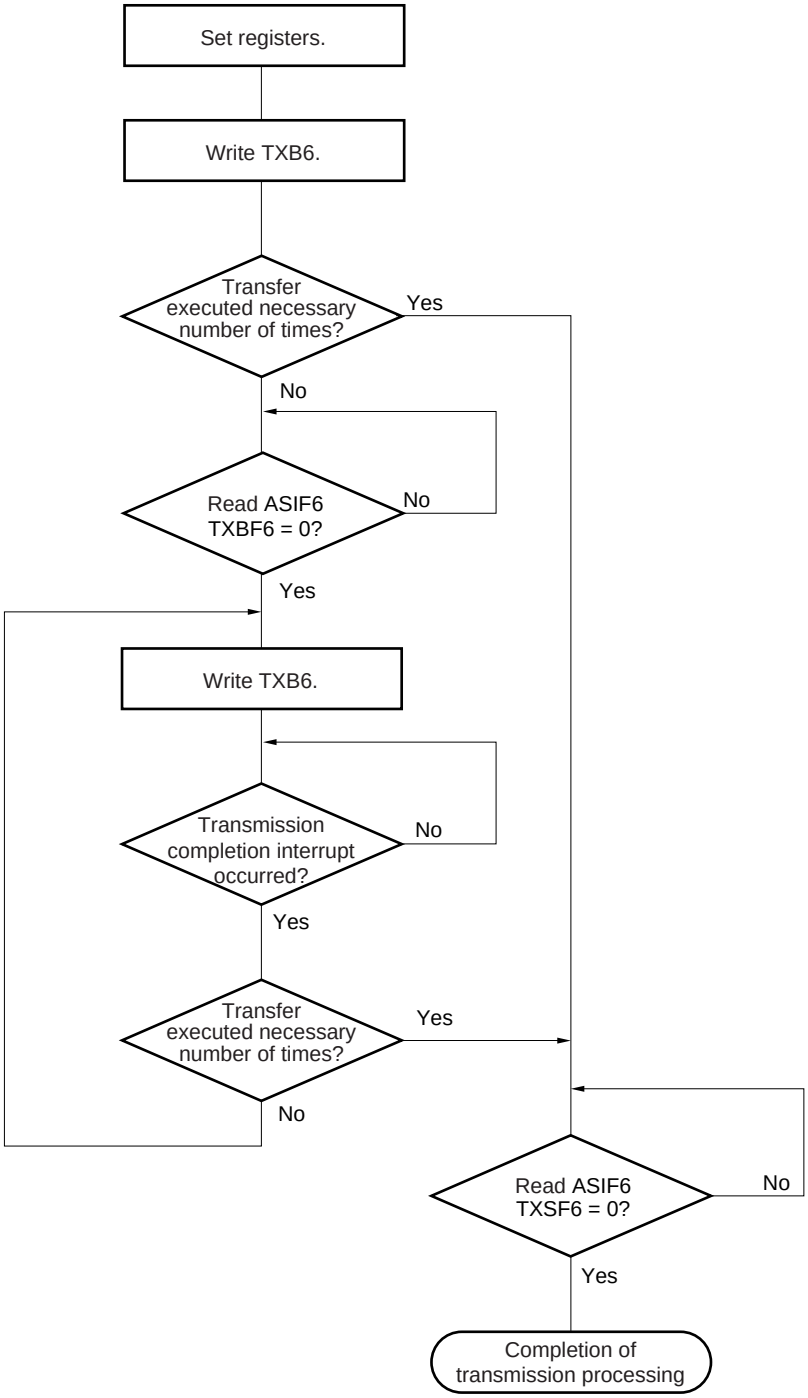
TXSF6	Transmission Status
0	Transmission is completed.
1	Transmission is in progress.

Cautions 1. To initialize the transmission unit upon completion of continuous transmission, be sure to check that the TXSF6 flag is "0" after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF6 flag is "1", the transmit data cannot be guaranteed.

2. During continuous transmission, an overrun error may occur, which means that the next transmission was completed before execution of INTST6 interrupt servicing after transmission of one data frame. An overrun error can be detected by developing a program that can count the number of transmit data and by referencing the TXSF6 flag.

Figure 11-16 shows an example of the continuous transmission processing flow.

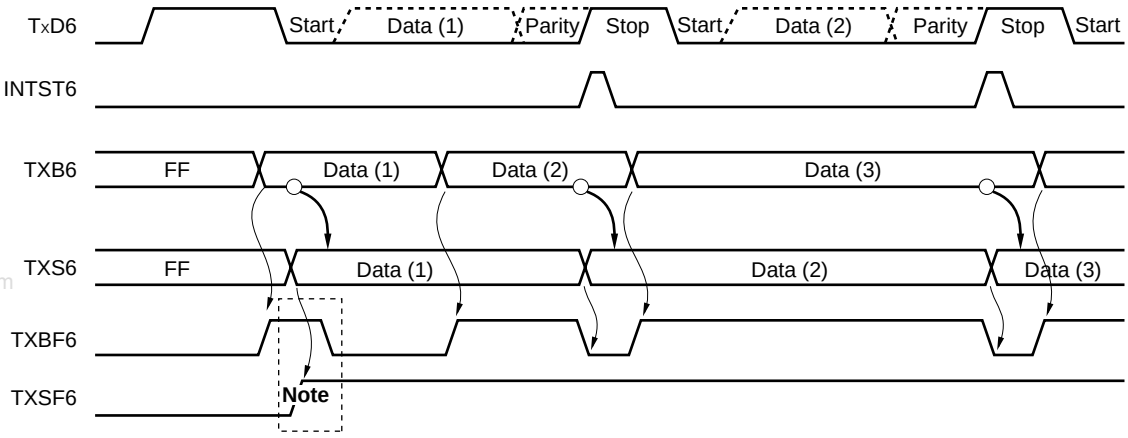
Figure 11-16. Example of Continuous Transmission Processing Flow



Remark TXB6: Transmit buffer register 6
ASIF6: Asynchronous serial interface transmission status register 6
TXBF6: Bit 1 of ASIF6 (transmit buffer data flag)
TXSF6: Bit 0 of ASIF6 (transmit shift register data flag)

Figure 11-17 shows the timing of starting continuous transmission, and Figure 11-18 shows the timing of ending continuous transmission.

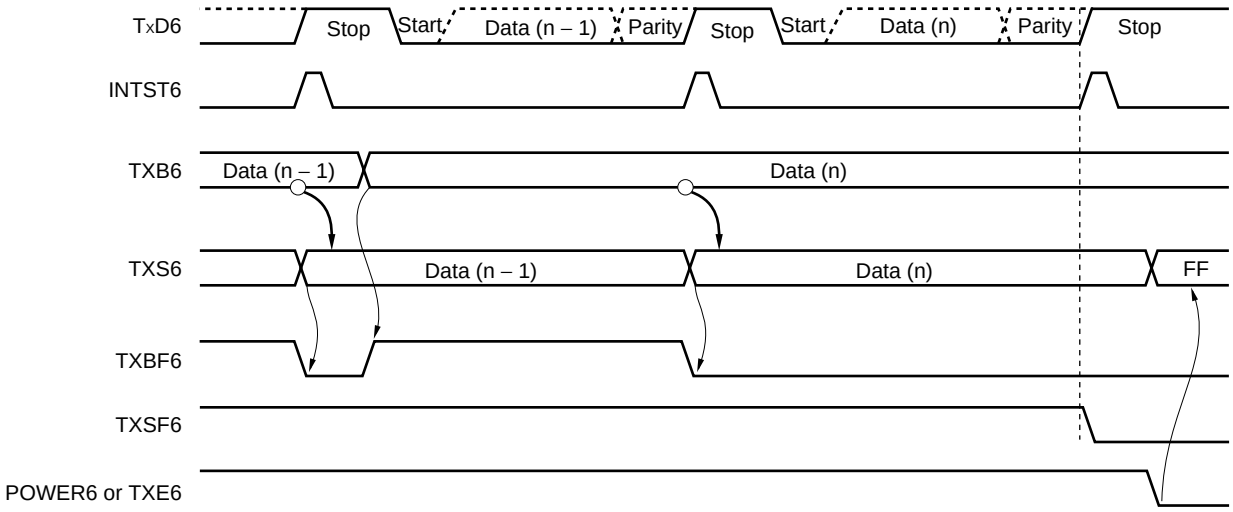
Figure 11-17. Timing of Starting Continuous Transmission



Note When ASIF6 is read, there is a period in which TXBF6 and TXSF6 = 1, 1. Therefore, judge whether writing is enabled using only the TXBF6 bit.

Remark TxD6: TxD6 pin (output)
INTST6: Interrupt request signal
TXB6: Transmit buffer register 6
TXS6: Transmit shift register 6
ASIF6: Asynchronous serial interface transmission status register 6
TXBF6: Bit 1 of ASIF6
TXSF6: Bit 0 of ASIF6

Figure 11-18. Timing of Ending Continuous Transmission



Remark TxD6: TxD6 pin (output)
INTST6: Interrupt request signal
TXB6: Transmit buffer register 6
TXS6: Transmit shift register 6
ASIF6: Asynchronous serial interface transmission status register 6
TXBF6: Bit 1 of ASIF6
TXSF6: Bit 0 of ASIF6
POWER6: Bit 7 of asynchronous serial interface operation mode register (ASIM6)
TXE6: Bit 6 of asynchronous serial interface operation mode register (ASIM6)

(e) Normal reception

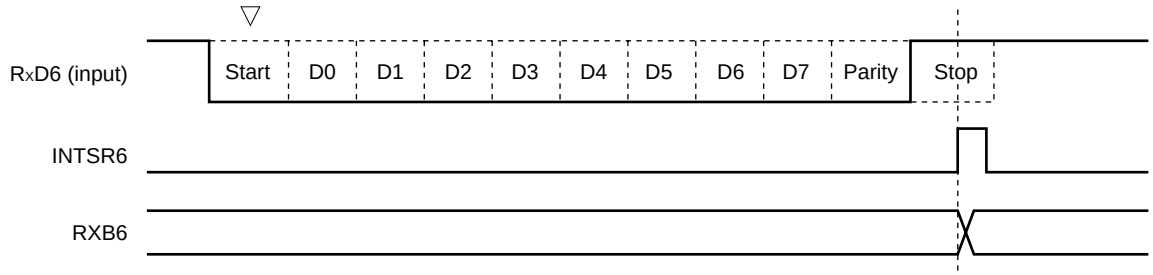
Reception is enabled and the RxD6 pin input is sampled when bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is set to 1 and then bit 5 (RXE6) of ASIM6 is set to 1.

The 8-bit counter of the baud rate generator starts counting when the falling edge of the RxD6 pin input is detected. When the set value of baud rate generator control register 6 (BRGC6) has been counted, the RxD6 pin input is sampled again (▽ in Figure 11-19). If the RxD6 pin is low level at this time, it is recognized as a start bit.

When the start bit is detected, reception is started, and serial data is sequentially stored in the receive shift register (RXS6) at the set baud rate. When the stop bit has been received, the reception completion interrupt (INTSR6) is generated and the data of RXS6 is written to receive buffer register 6 (RXB6). If an overrun error (OVE6) occurs, however, the receive data is not written to RXB6.

Even if a parity error (PE6) occurs while reception is in progress, reception continues to the reception position of the stop bit, and an error interrupt (INTSR6/INTSRE6) is generated on completion of reception.

Figure 11-19. Reception Completion Interrupt Request Timing



Cautions 1. Be sure to read receive buffer register 6 (RXB6) even if a reception error occurs. Otherwise, an overrun error will occur when the next data is received, and the reception error status will persist.

2. Reception is always performed with the "number of stop bits = 1". The second stop bit is ignored.

3. Be sure to read asynchronous serial interface reception error status register 6 (ASIS6) before reading RXB6.

(f) Reception error

Three types of errors may occur during reception: a parity error, framing error, or overrun error. If the error flag of asynchronous serial interface reception error status register 6 (ASIS6) is set as a result of data reception, a reception error interrupt request (INTSR6/INTSRE6) is generated.

Which error has occurred during reception can be identified by reading the contents of ASIS6 in the reception error interrupt servicing (INTSR6/INTSRE6) (see **Figure 11-6**).

The contents of ASIS6 are reset to 0 when ASIS6 is read.

Table 11-3. Cause of Reception Error

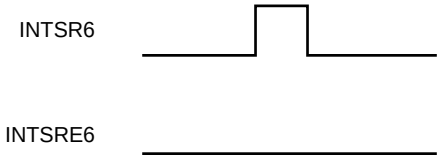
Reception Error	Cause
Parity error	The parity specified for transmission does not match the parity of the receive data.
Framing error	Stop bit is not detected.
Overrun error	Reception of the next data is completed before data is read from receive buffer register 6 (RXB6).

The error interrupt can be separated into reception completion interrupt (INTSR6) and error interrupt (INTSRE6) by clearing bit 0 (ISRM6) of asynchronous serial interface operation mode register 6 (ASIM6) to 0.

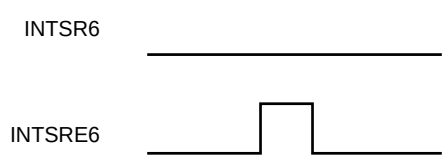
Figure 11-20. Reception Error Interrupt

1. If ISRM6 is cleared to 0 (reception completion interrupt (INTSR6) and error interrupt (INTSRE6) are separated)

(a) No error during reception

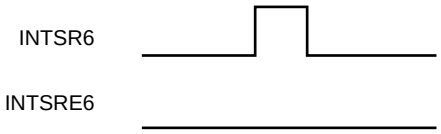


(b) Error during reception

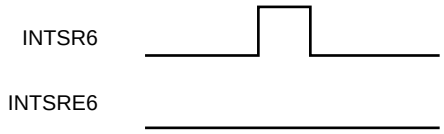


2. If ISRM6 is set to 1 (error interrupt is included in INTSR6)

(a) No error during reception



(b) Error during reception



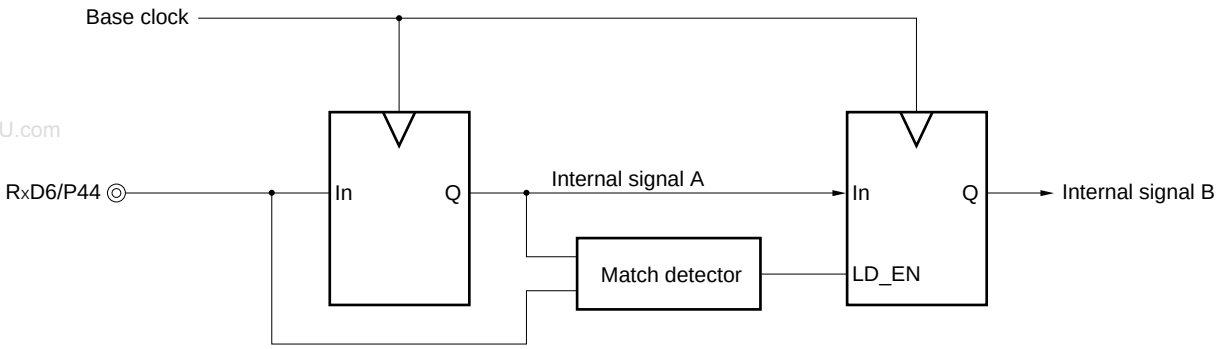
(g) Noise filter of receive data

The RXD6 signal is sampled with the base clock output by the prescaler block.

If two sampled values are the same, the output of the match detector changes, and the data is sampled as input data.

Because the circuit is configured as shown in Figure 11-21, the internal processing of the reception operation is delayed by two clocks from the external signal status.

Figure 11-21. Noise Filter Circuit



(h) SBF transmission

When the device is incorporated in LIN, the SBF (Synchronous Break Field) transmission control function is used for transmission. For the transmission operation of LIN, see **Figure 11-1 LIN Transmission Operation**.

An SBF length that is a low-level width of 13 bits or more is set by bits 4 to 2 (SBL62 to SBL60) of asynchronous serial interface control register 6 (ASICL6). If the output width needs to be adjusted more accurately, use the baud rate value of the normal UART transmission function.

[Setting method]

Transmit 00H by setting the number of character bits of the data to 8 bits and the parity bit to 0 parity or even parity. This enables a low-level transmission of a data frame consisting of 10 bits (1 bit (start bit) + 8 bits (character bits) + 1 bit (parity bit)).

Adjust the baud rate value to adjust this 10-bit low level to the targeted 13-bit SBF length (SBL62, SBL61, SBL60 = 1, 0, 1).

Example If LIN is to be transmitted under the following conditions

- Base clock of UART6 = 5 MHz (set by clock selection register 6 (CKSR6))
- Target baud rate value = 19200 bps

To realize the above baud rate value, the length of a 13-bit SBF is as follows if the baud rate generator control register 6 (BRGC6) is set to 130.

- 13-bit SBF length = 0.2 μs $\times$ 130 $\times$ 2 $\times$ 13 = 676 μs

  To realize a 13-bit SBF length in 10 bits, set a value 1.3 times the targeted baud rate to BRGC6. In this example, set 169 to BRGC6. The transmission length of a 10-bit low level in this case is as follows, and matches the 13-bit SBF length.

- 10-bit low-level transmission length = 0.2 μs $\times$ 169 $\times$ 2 $\times$ 10 = 676 μs

  If the number of bits set by BRGC6 runs short, adjust the number of bits by setting the base clock of UART6.

Figure 11-22. Example of Setting Procedure of SBF Transmission (Flowchart)

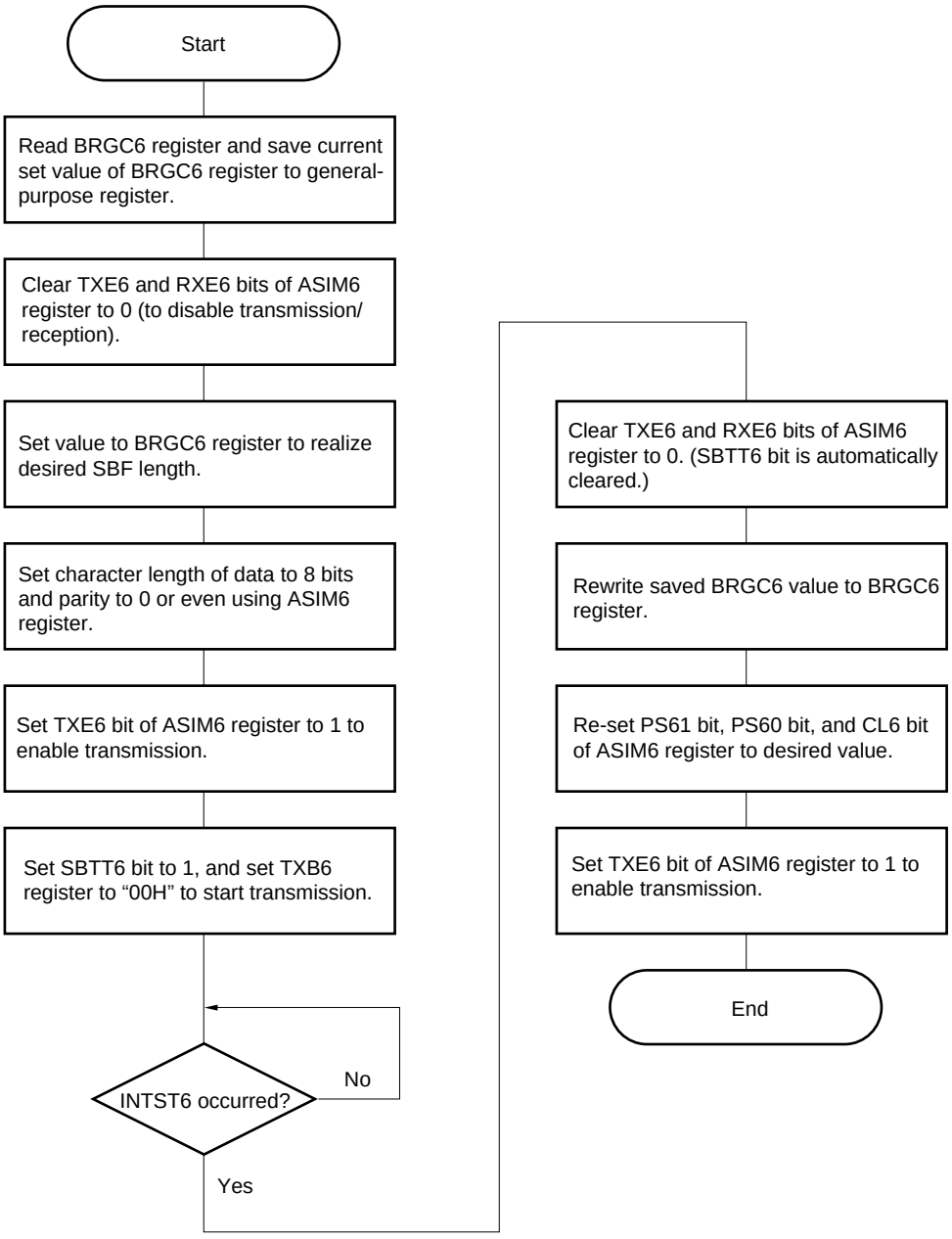
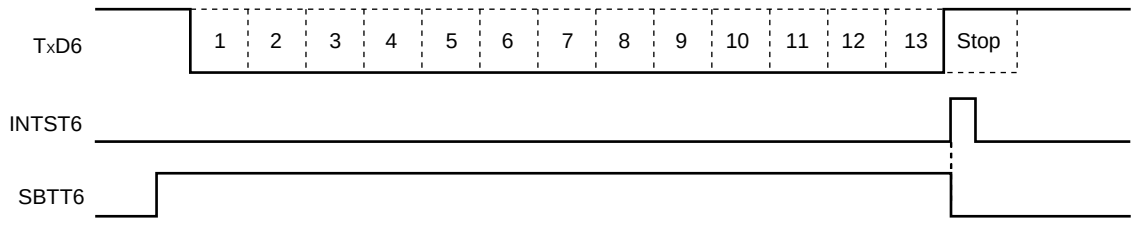


Figure 11-23. SBF Transmission



Remark TxD6: TxD6 pin (output)
INTST6: Transmission completion interrupt request
SBTT6: Bit 5 of asynchronous serial interface control register 6 (ASICL6)

(i) SBF reception

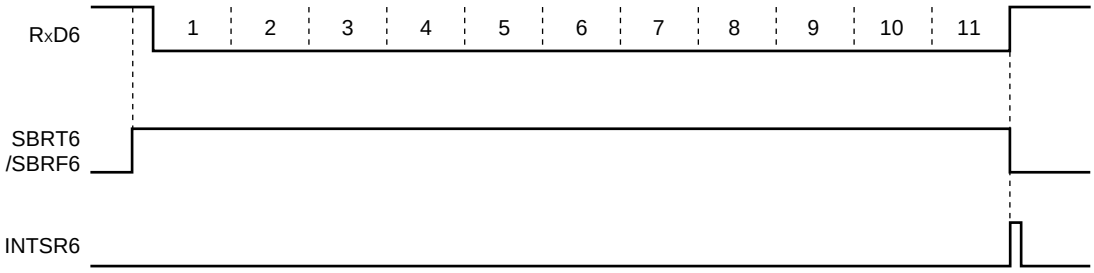
When the device is incorporated in LIN, the SBF (Synchronous Break Field) reception control function is used for reception. For the reception operation of LIN, see **Figure 11-2 LIN Reception Operation**.

Reception is enabled when bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is set to 1 and then bit 5 (RXE6) of ASIM6 is set to 1. SBF reception is enabled when bit 6 (SBRT6) of asynchronous serial interface control register 6 (ASICL6) is set to 1. In the SBF reception enabled status, the RxD6 pin is sampled and the start bit is detected in the same manner as the normal reception enable status.

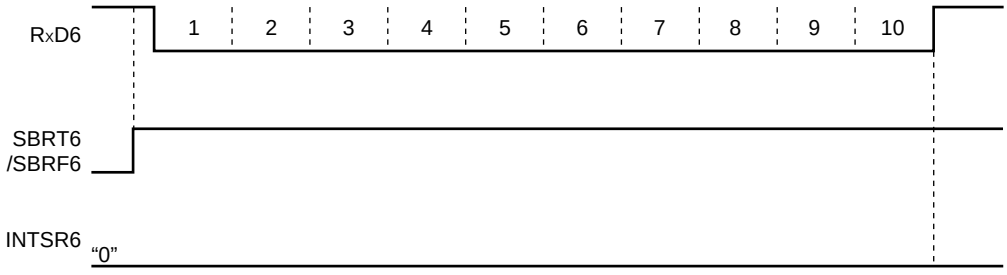
When the start bit has been detected, reception is started, and serial data is sequentially stored in the receive shift register 6 (RXS6) at the set baud rate. When the stop bit is received and if the width of SBF is 11 bits or more, a reception completion interrupt request (INTSR6) is generated as normal processing. At this time, the SBRF6 and SBRT6 bits are automatically cleared, and SBF reception ends. Detection of errors, such as OVE6, PE6, and FE6 (bits 0 to 2 of asynchronous serial interface reception error status register 6 (ASIS6)) is suppressed, and error detection processing of UART communication is not performed. In addition, data transfer between receive shift register 6 (RXS6) and receive buffer register 6 (RXB6) is not performed, and the reset value of FFH is retained. If the width of SBF is 10 bits or less, an interrupt does not occur as error processing after the stop bit has been received, and the SBF reception mode is restored. In this case, the SBRF6 and SBRT6 bits are not cleared.

Figure 11-24. SBF Reception

1. Normal SBF reception (stop bit is detected with a width of more than 10.5 bits)



2. SBF reception error (stop bit is detected with a width of 10.5 bits or less)



Remark RxD6: RxD6 pin (input)
SBRT6: Bit 6 of asynchronous serial interface control register 6 (ASICL6)
SBRF6: Bit 7 of ASICL6
INTSR6: Reception completion interrupt request

11.4.3 Dedicated baud rate generator

The dedicated baud rate generator consists of a source clock selector and an 8-bit programmable counter, and generates a serial clock for transmission/reception of UART6.

Separate 8-bit counters are provided for transmission and reception.

(1) Configuration of baud rate generator

- Base clock

  The clock selected by bits 3 to 0 (TPS63 to TPS60) of clock selection register 6 (CKSR6) is supplied to each module when bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is 1. This clock is called the base clock and its frequency is called f_{XCLK6}. The base clock is fixed to low level when POWER6 = 0.

- Transmission counter

  This counter stops operation, cleared to 0, when bit 7 (POWER6) or bit 6 (TXE6) of asynchronous serial interface operation mode register 6 (ASIM6) is 0.

  It starts counting when POWER6 = 1 and TXE6 = 1.

  The counter is cleared to 0 when the first data transmitted is written to transmit buffer register 6 (TXB6).

  If data are continuously transmitted, the counter is cleared to 0 again when one frame of data has been completely transmitted. If there is no data to be transmitted next, the counter is not cleared to 0 and continues counting until POWER6 or TXE6 is cleared to 0.

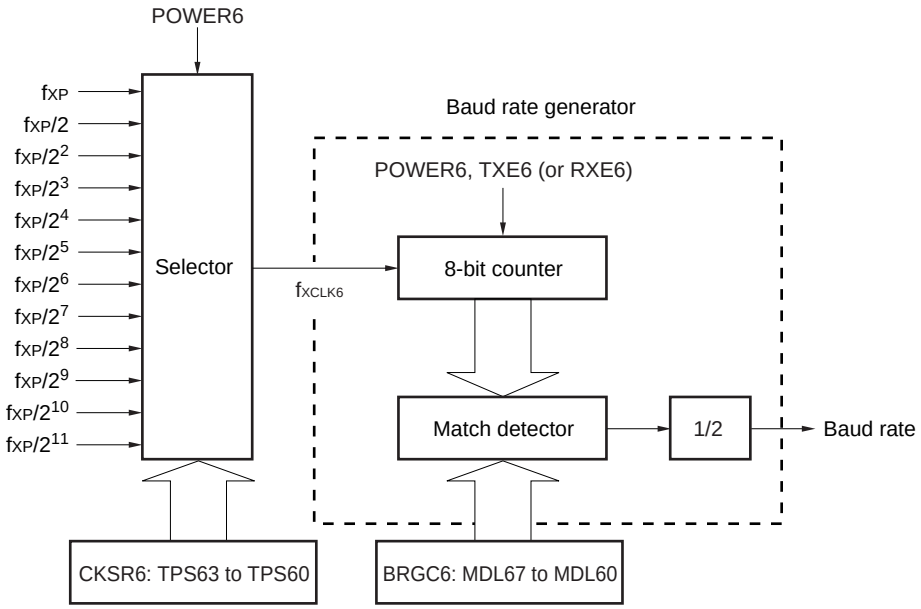
- Reception counter

  This counter stops operation, cleared to 0, when bit 7 (POWER6) or bit 5 (RXE6) of asynchronous serial interface operation mode register 6 (ASIM6) is 0.

  It starts counting when the start bit has been detected.

  The counter stops operation after one frame has been received, until the next start bit is detected.

Figure 11-25. Configuration of Baud Rate Generator



Remark POWER6: Bit 7 of asynchronous serial interface operation mode register 6 (ASIM6)
TXE6: Bit 6 of ASIM6
RXE6: Bit 5 of ASIM6
CKSR6: Clock selection register 6
BRGC6: Baud rate generator control register 6

(2) Generation of serial clock

A serial clock can be generated by using clock selection register 6 (CKSR6) and baud rate generator control register 6 (BRGC6).

Select the clock to be input to the 8-bit counter by using bits 3 to 0 (TPS63 to TPS60) of CKSR6.

Bits 7 to 0 (MDL67 to MDL60) of BRGC6 can be used to select the division value of the 8-bit counter.

(a) Baud rate

The baud rate can be calculated by the following expression.

- Baud rate = $\frac{f_{XCLK6}}{2 \times k}$ [bps]

f_{XCLK6}: Frequency of base clock selected by TPS63 to TPS60 bits of CKSR6 register

k: Value set by MDL67 to MDL60 bits of BRGC6 register (k = 8, 9, 10, ..., 255)

(b) Error of baud rate

The baud rate error can be calculated by the following expression.

- Error (%) = $\left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1\right) \times 100$ [%]

Cautions 1. Keep the baud rate error during transmission to within the permissible error range at the reception destination.

2. Make sure that the baud rate error during reception satisfies the range shown in (4) Permissible baud rate range during reception.

Example: Frequency of base clock = 10 MHz = 10,000,000 Hz
Set value of MDL67 to MDL60 bits of BRGC6 register = 00100001B (k = 33)
Target baud rate = 153600 bps

Baud rate = 10 M/(2 × 33)
= 10000000/(2 × 33) = 151,515 [bps]

Error = (151515/153600 – 1) × 100
= –1.357 [%]

(3) Example of setting baud rate

Table 11-4. Set Data of Baud Rate Generator

Baud Rate [bps]	f_{XP} = 10.0 MHz				f_{XP} = 8.0 MHz				f_{XP} = 4.19 MHz			
	TPS63 to TPS60	k	Calculated Value	ERR[%]	TPS63 to TPS60	k	Calculated Value	ERR[%]	TPS63 to TPS60	k	Calculated Value	ERR[%]
600	6H	130	601	0.16	6H	104	601	0.16	5H	109	601	0.11
1200	5H	130	1202	0.16	5H	104	1202	0.16	4H	109	1201	0.11
2400	4H	130	2404	0.16	4H	104	2404	0.16	3H	109	2403	0.11
4800	3H	130	4808	0.16	3H	104	4808	0.16	2H	109	4805	0.11
9600	2H	130	9615	0.16	2H	104	9615	0.16	1H	109	9610	0.11
10400	2H	120	10417	0.16	2H	96	10417	0.16	1H	101	10475	–0.28
19200	1H	130	19231	0.16	1H	104	19231	0.16	0H	109	19220	0.11
31250	1H	80	31250	0.00	0H	128	31250	0.00	0H	67	31268	0.06
38400	0H	130	38462	0.16	0H	104	38462	0.16	0H	55	38090	–0.80
76800	0H	65	76923	0.16	0H	52	76923	0.16	0H	27	77593	1.03
115200	0H	43	116279	0.94	0H	35	114286	–0.79	0H	18	116389	1.03
153600	0H	33	151515	–1.36	0H	26	153846	0.16	0H	14	149643	–2.58
230400	0H	22	227272	–1.36	0H	17	235294	2.12	0H	9	232778	1.03

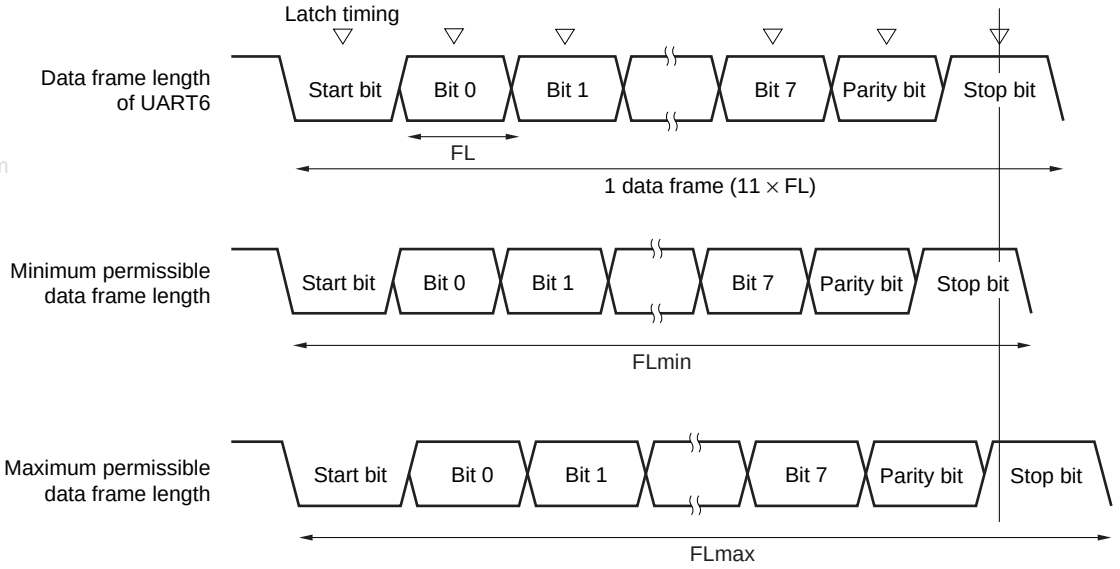
Remark TPS63 to TPS60: Bits 3 to 0 of clock selection register 6 (CKSR6) (setting of base clock (f_{XCLK6}))
k: Value set by MDL67 to MDL60 bits of baud rate generator control register 6 (BRGC6) (k = 8, 9, 10, ..., 255)
f_{XP}: Oscillation frequency of clock to peripheral hardware
ERR: Baud rate error

(4) Permissible baud rate range during reception

The permissible error from the baud rate at the transmission destination during reception is shown below.

Caution Make sure that the baud rate error during reception is within the permissible error range, by using the calculation expression shown below.

Figure 11-26. Permissible Baud Rate Range During Reception



As shown in Figure 11-26, the latch timing of the receive data is determined by the counter set by baud rate generator control register 6 (BRGC6) after the start bit has been detected. If the last data (stop bit) meets this latch timing, the data can be correctly received.

Assuming that 11-bit data is received, the theoretical values can be calculated as follows.

$FL = (Brate)^{-1}$

Brate: Baud rate of UART6
k: Set value of BRGC6
FL: 1-bit data length
Margin of latch timing: 2 clocks

Minimum permissible data frame length: $FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$

Therefore, the maximum receivable baud rate at the transmission destination is as follows.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k+2} \text{ Brate}$$

Similarly, the maximum permissible data frame length can be calculated as follows.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL$$

$$FLmax = \frac{21k-2}{20k} FL \times 11$$

Therefore, the minimum receivable baud rate at the transmission destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k-2} \text{ Brate}$$

The permissible baud rate error between UART6 and the transmission destination can be calculated from the above minimum and maximum baud rate expressions, as follows.

Table 11-5. Maximum/Minimum Permissible Baud Rate Error

Division Ratio (k)	Maximum Permissible Baud Rate Error	Minimum Permissible Baud Rate Error
8	+3.53%	–3.61%
20	+4.26%	–4.31%
50	+4.56%	–4.58%
100	+4.66%	–4.67%
255	+4.72%	–4.73%

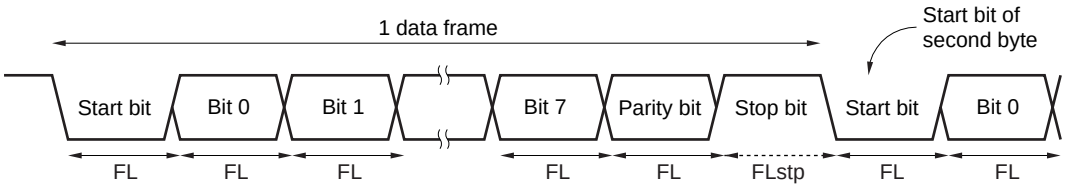
Remarks 1. The permissible error of reception depends on the number of bits in one frame, input clock frequency, and division ratio (k). The higher the input clock frequency and the higher the division ratio (k), the higher the permissible error.

2. k: Set value of BRGC6

(5) Data frame length during continuous transmission

When data is continuously transmitted, the data frame length from a stop bit to the next start bit is extended by two clocks of base clock from the normal value. However, the result of communication is not affected because the timing is initialized on the reception side when the start bit is detected.

Figure 11-27. Data Frame Length During Continuous Transmission



Where the 1-bit data length is FL, the stop bit length is FLstp, and base clock frequency is f_{XCLK6}, the following expression is satisfied.

$$FLstp = FL + 2/f_{XCLK6}$$

Therefore, the data frame length during continuous transmission is:

$$\text{Data frame length} = 11 \times FL + 2/f_{XCLK6}$$

CHAPTER 12 INTERRUPT FUNCTIONS

12.1 Interrupt Function Types

All interrupts are controlled as maskable interrupts.

- **Maskable interrupts**
  These interrupts undergo mask control. If two or more interrupt requests are simultaneously generated, each interrupt has a predetermined priority as shown in Table 12-1.
  A standby release signal is generated.
  There are ten internal sources and four external sources of maskable interrupts.

12.2 Interrupt Sources and Configuration

There are a total of 14 interrupt sources, and up to four reset sources (see **Table 12-1**).

Table 12-1. Interrupt Sources

Interrupt Type	Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	1	INTLVI	Low-voltage detection[Note 3]	Internal	0006H	(A)
	2	INTP0	Pin input edge detection	External	0008H	(B)
	3	INTP1			000AH	
	4	INTTMH1	Match between TMH1 and CMP01 (when compare register is specified)	Internal	000CH	(A)
	5	INTTM000	Match between TM00 and CR000 (when compare register is specified)		000EH	
	6	INTTM010	Match between TM00 and CR010 (when compare register is specified)		0010H	
	7	INTAD	End of A/D conversion		0012H	
	8	INTFLC	End of flash memory programming		0014H	
	9	INTP2	Pin input edge detection	External	0016H	(B)
	10	INTP3			0018H	
	11	INTTM80	Match between TM80 and CR80	Internal	001AH	(A)
	12	INTSRE6	UART6 reception error occurrence		001CH	
	13	INTSR6	End of UART6 reception		001EH	
	14	INTST6	End of UART6 transmission		0020H	
Reset	–	RESET	Reset input	–	0000H	–
		POC	Power-on-clear			
		LVI	Low-voltage detection[Note 4]			
		WDT	WDT overflow			

Notes 1. Priority is the priority order when several maskable interrupt requests are generated at the same time. 1 is the highest and 14 is the lowest.

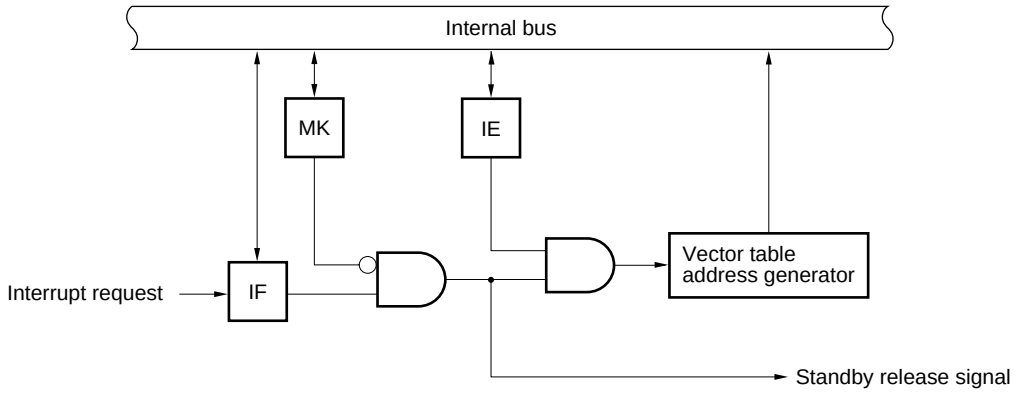
2. Basic configuration types (A) and (B) correspond to (A) and (B) in Figure 12-1.

3. When bit 1 (LVIMD) of low-voltage detection register (LVIM) = 0 is selected.

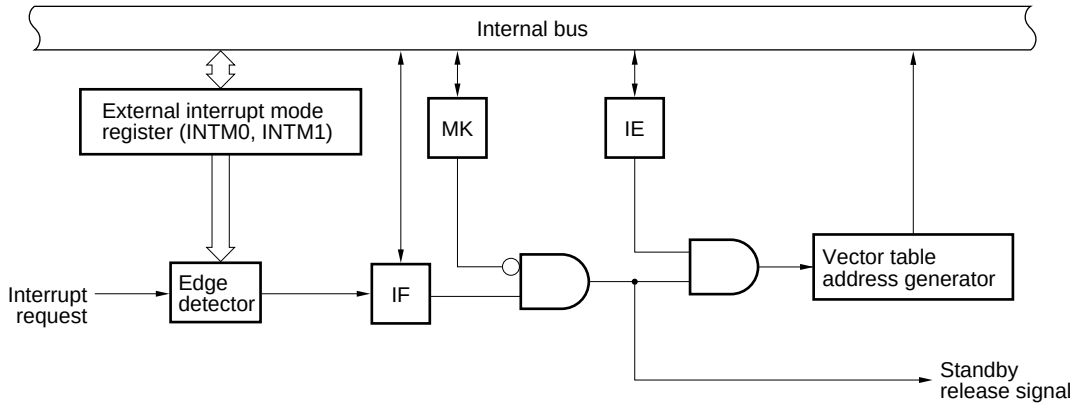
4. When bit 1 (LVIMD) of low-voltage detection register (LVIM) = 1 is selected.

Figure 12-1. Basic Configuration of Interrupt Function

(A) Internal maskable interrupt



(B) External maskable interrupt



IF: Interrupt request flag

IE: Interrupt enable flag

MK: Interrupt mask flag

12.3 Interrupt Function Control Registers

The interrupt functions are controlled by the following four types of registers.

- Interrupt request flag registers (IF0, IF1)
- Interrupt mask flag registers (MK0, MK1)
- External interrupt mode registers (INTM0, INTM1)
- Program status word (PSW)

Table 12-2 lists interrupt requests, the corresponding interrupt request flags, and interrupt mask flags.

Table 12-2. Interrupt Request Signals and Corresponding Flags

Interrupt Request Signal	Interrupt Request Flag	Interrupt Mask Flag
INTLVI	LVIIF	LVIMK
INTP0	PIF0	PMK0
INTP1	PIF1	PMK1
INTTMH1	TMIFH1	TMMKH1
INTTM000	TMIF000	TMMK000
INTTM010	TMIF010	TMMK010
INTAD	ADIF	ADMK
INTFLC	FLIF	FLMK
INTP2	PIF2	PMK2
INTP3	PIF3	PMK3
INTTM80	TMIF80	TMMK80
INTSRE6	SREIF6	SREMK6
INTSR6	SRIF6	SRMK6
INTST6	STIF6	STMK6

(1) Interrupt request flag registers (IF0, IF1)

An interrupt request flag is set to 1 when the corresponding interrupt request is issued, or when the instruction is executed. It is cleared to 0 by executing an instruction when the interrupt request is acknowledged or when a reset signal is input.

IF0 and IF1 are set with a 1-bit or 8-bit memory manipulation instruction.

Reset input clears IF0 and IF1 to 00H.

Figure 12-2. Format of Interrupt Request Flag Registers (IF0, IF1)

Address: FFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0	ADIF	TMIF010	TMIF000	TMIFH1	PIF1	PIF0	LVIIF	0

Address: FFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1	0	STIF6	SRIF6	SREIF6	TMIF80	PIF3	PIF2	FLIF

××IF×	Interrupt request flag
0	No interrupt request signal has been issued.
1	An interrupt request signal has been issued; an interrupt request status.

Caution Because P30, P31, P41, and P43 have an alternate function as external interrupt inputs, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, the interrupt mask flag should be set to 1 before using the output mode.

(2) Interrupt mask flag registers (MK0, MK1)

The interrupt mask flag is used to enable and disable the corresponding maskable interrupts.

MK0 and MK1 are set with a 1-bit or 8-bit memory manipulation instruction.

Reset input sets MK0 and MK1 to FFH.

Figure 12-3. Format of Interrupt Mask Flag Registers (MK0, MK1)

Address: FFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0	ADMK	TMMK010	TMMK000	TMMKH1	PMK1	PMK0	LVIMK	1

Address: FFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1	1	STMK6	SRMK6	SREMK6	TMMK80	PMK3	PMK2	FLMK

××MK×	Interrupt servicing control
0	Enables interrupt servicing.
1	Disables interrupt servicing.

Caution Because P30, P31, P41, and P43 have an alternate function as external interrupt inputs, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, the interrupt mask flag should be set to 1 before using the output mode.

(3) External interrupt mode register 0 (INTM0)

This register is used to set the valid edge of INTP0 to INTP2.

INTM0 is set with an 8-bit memory manipulation instruction.

Reset input clears INTM0 to 00H.

Figure 12-4. Format of External Interrupt Mode Register 0 (INTM0)

Address: FFECH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
INTM0	ES21	ES20	ES11	ES10	ES01	ES00	0	0

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. Be sure to clear bits 0 and 1 to 0.

2. Before setting the INTM0 register, be sure to set the corresponding interrupt mask flag (××MK× = 1) to disable interrupts. After setting the INTM0 register, clear the interrupt request flag (××IF× = 0), then clear the interrupt mask flag (××MK× = 0), which will enable interrupts.

(4) External interrupt mode register 1 (INTM1)

INTM1 is used to specify the valid edge for INTP3.

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

Reset input clears INTM1 to 00H.

Figure 12-5. Format of External Interrupt Mode Register 1 (INTM1)

Address: FFEDH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
INTM1	0	0	0	0	0	0	ES31	ES30

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. Be sure to clear bits 2 to 7 to 0.

2. Before setting INTM1, set PMK3 to 1 to disable interrupts.

To enable interrupts, clear PIF3 to 0, then clear PMK3 to 0.

(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 vectored interrupt is acknowledged, the PSW is automatically saved to a stack, and the IE flag is reset to 0.

Reset input sets PSW to 02H.

Figure 12-6. Program Status Word Configuration

Symbol	7	6	5	4	3	2	1	0	After reset
PSW	IE	Z	0	AC	0	0	1	CY	02H

Used in the execution of ordinary instructions

IE	Whether to enable/disable interrupt acknowledgment
0	Disabled
1	Enabled

12.4 Interrupt Servicing Operation

12.4.1 Maskable interrupt request acknowledgment operation

A maskable interrupt request can be acknowledged when the interrupt request flag is set to 1 and the corresponding interrupt mask flag is cleared to 0. A vectored interrupt request is acknowledged in the interrupt enabled status (when the IE flag is set to 1).

The time required to start the interrupt servicing after a maskable interrupt request has been generated is shown in Table 12-3.

See Figures 12-8 and 12-9 for the interrupt request acknowledgment timing.

Table 12-3. Time from Generation of Maskable Interrupt Request to Servicing

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 instructions.

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 acknowledged starting from the interrupt request assigned the highest priority.

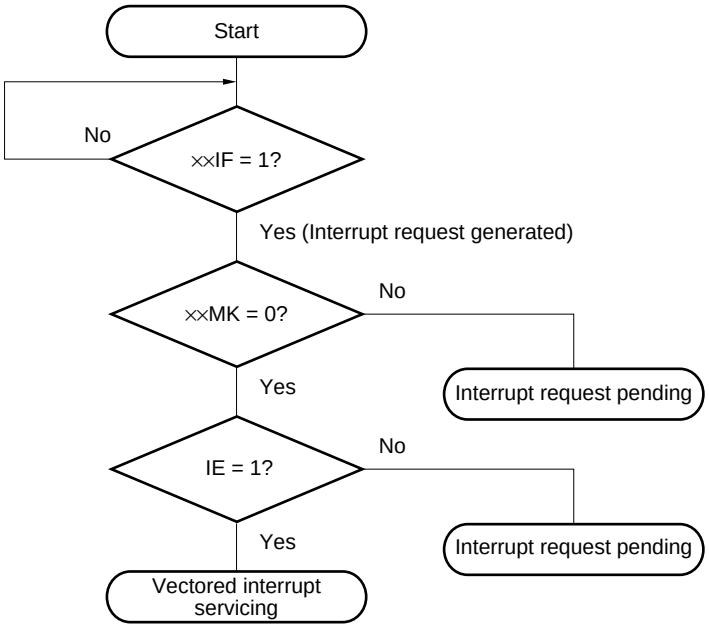
A pending interrupt is acknowledged when a status in which it can be acknowledged is set.

Figure 12-7 shows the algorithm of interrupt request acknowledgment.

When a maskable interrupt request is acknowledged, the contents of the 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.

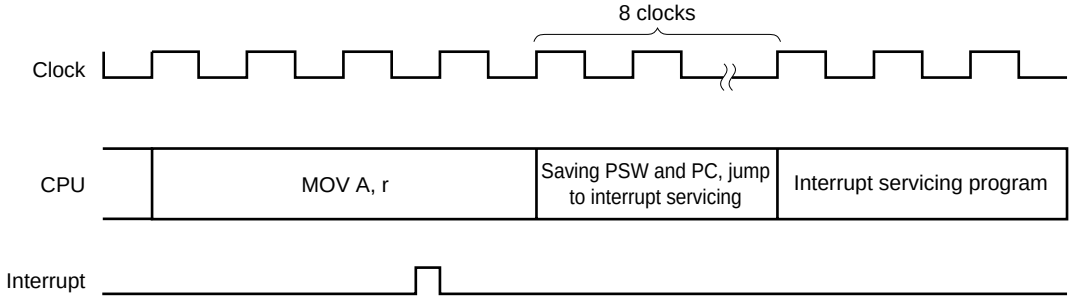
To return from interrupt servicing, use the RETI instruction.

Figure 12-7. Interrupt Request Acknowledgment Processing Algorithm



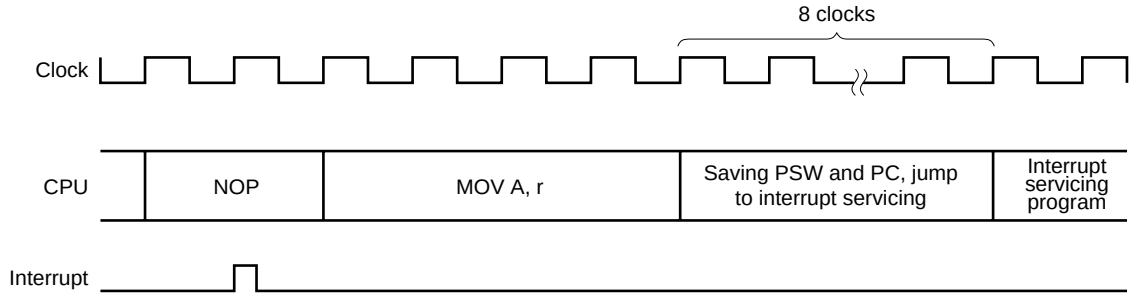
××IF: Interrupt request flag
××MK: Interrupt mask flag
IE: Flag to control maskable interrupt request acknowledgment (1 = enable, 0 = disable)

Figure 12-8. Interrupt Request Acknowledgment 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 acknowledged after the instruction under execution is complete. Figure 12-8 shows an example of the interrupt request acknowledgment 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 acknowledgment processing is performed after the MOV A, r instruction is executed.

Figure 12-9. Interrupt Request Acknowledgment Timing (When Interrupt Request Flag Is Set at Last Clock During Instruction Execution)



If an interrupt request flag (××IF) is set at the last clock of the instruction, the interrupt acknowledgment processing starts after the next instruction is executed.

Figure 12-9 shows an example of the interrupt request acknowledgment 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 acknowledgment processing is performed.

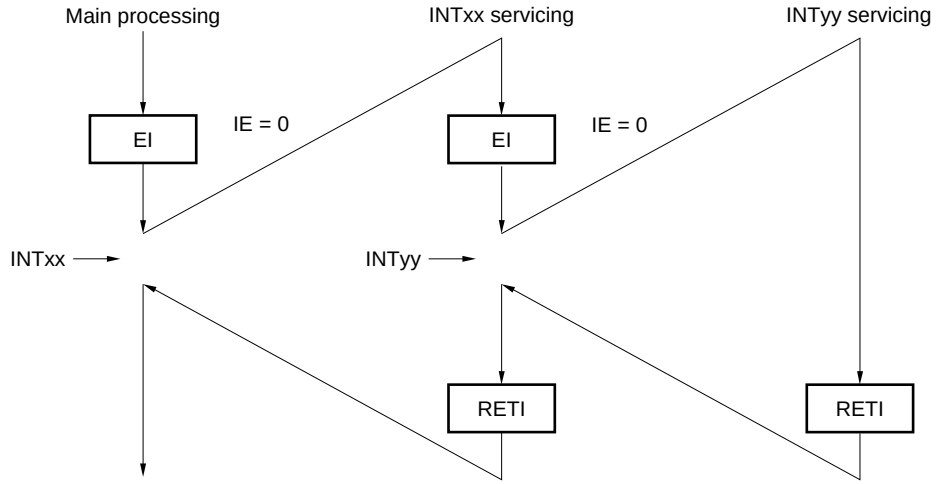
Caution Interrupt requests will be held pending while the interrupt request flag registers (IF0, IF1) or interrupt mask flag registers (MK0, MK1) are being accessed.

12.4.2 Multiple interrupt servicing

Multiple interrupt servicing in which another interrupt is acknowledged while an interrupt is being serviced can be performed using a priority order system. When two or more interrupts are generated at once, interrupt servicing is performed according to the priority assigned to each interrupt request in advance (see **Table 12-1**).

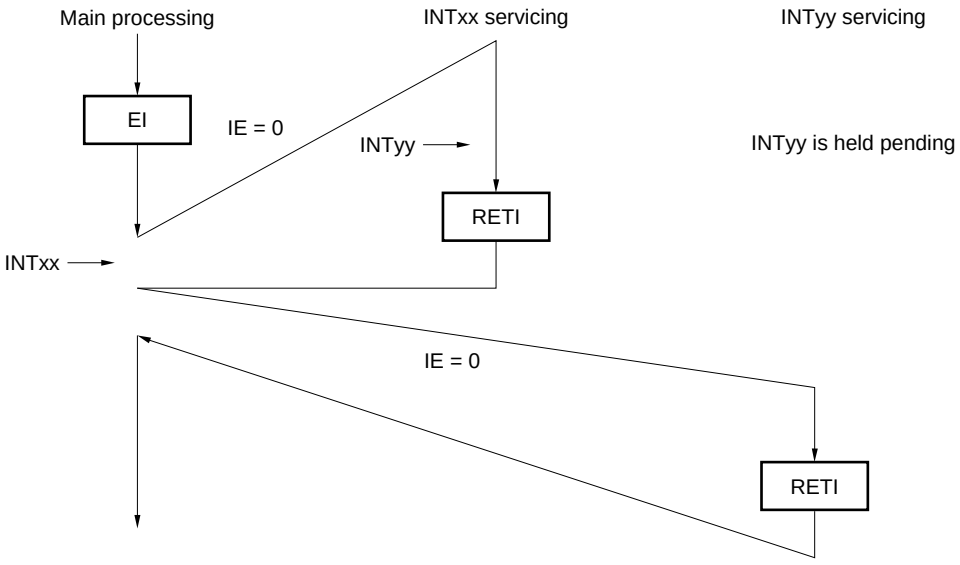
Figure 12-10. Example of Multiple Interrupts

Example 1. Multiple interrupts are acknowledged



During interrupt INTxx servicing, interrupt request INTyy is acknowledged, and multiple interrupts are generated. The EI instruction is issued before each interrupt request acknowledgment, and the interrupt request acknowledgment enable state is set.

Example 2. Multiple interrupts are not generated because interrupts are not enabled



Because interrupts are not enabled in interrupt INTxx servicing (the EI instruction is not issued), interrupt request INTyy is not acknowledged, and multiple interrupts are not generated. The INTyy request is held pending and acknowledged after the INTxx servicing is performed.

IE = 0: Interrupt request acknowledgment disabled

12.4.3 Interrupt request pending

Some instructions may keep pending the acknowledgment of an instruction request until the completion of the execution of the next instruction even if the interrupt request (maskable interrupt and external interrupt) is generated during the execution. The following shows such instructions (interrupt request pending instruction).

- Manipulation instruction for interrupt request flag registers (IF0, IF1)
- Manipulation instruction for interrupt mask flag registers (MK0, MK1)

CHAPTER 13 STANDBY FUNCTION

13.1 Standby Function and Configuration

13.1.1 Standby function

Table 13-1. Relationship Between Operation Clocks in Each Operation Status

Status / Operation Mode	Low-Speed Ring-OSC Oscillator			System Clock	Clock Supplied to Peripheral Hardware
	Note 1	**Note 2**			
		LSRSTOP = 0	LSRSTOP = 1		
Reset	Stopped			Stopped	Stopped
STOP	Oscillating	Oscillating[Note 3]	Stopped		
HALT				Oscillating	Oscillating

Notes 1. When “Cannot be stopped” is selected for low-speed Ring-OSC by the option byte.

2. When it is selected that the low-speed Ring-OSC oscillator “can be stopped by software”, oscillation of the low-speed Ring-OSC oscillator can be stopped by LSRSTOP.

3. If the operating clock of the watchdog timer is the low-speed Ring-OSC clock, the watchdog timer is stopped.

Caution The LSRSTOP setting is valid only when “Can be stopped by software” is set for the low-speed Ring-OSC oscillator by the option byte.

Remark LSRSTOP: Bit 0 of the low-speed Ring-OSC mode register (LSRCM)

The standby function is designed to reduce the operating current of the system. The following two modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. Oscillation of the system clock oscillator continues. If the low-speed Ring-OSC oscillator is operating before the HALT mode is set, oscillation of the clock of the low-speed Ring-OSC oscillator continues (refer to **Table 13-1**. Oscillation of the low-speed Ring-OSC clock (whether it cannot be stopped or can be stopped by software) is set by the option byte). In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the system clock oscillator stops, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, select the HALT mode if processing must be immediately started by an interrupt request when the STOP mode is released because the operation stops for the duration of eight clocks of the low-speed Ring-OSC clock (because an additional wait time for stabilizing oscillation elapses when crystal/ceramic oscillation is used).

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Cautions 1. When shifting to the STOP mode, be sure to stop the peripheral hardware operation before executing STOP instruction (except the peripheral hardware that operates on the low-speed Ring-OSC clock).

2. The following sequence is recommended for operating current reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) and bit 0 (ADCE) of the A/D converter mode register (ADM) to 0 to stop the A/D conversion operation, and then execute the HALT or STOP instruction.

3. If the low-speed Ring-OSC oscillator is operating before the STOP mode is set, oscillation of the low-speed Ring-OSC clock cannot be stopped in the STOP mode (refer to Table 13-1).

13.1.2 Registers used during standby

The oscillation stabilization time after the standby mode is released is controlled by the oscillation stabilization time select register (OSTS).

Remark For the registers that start, stop, or select the clock, see **CHAPTER 5 CLOCK GENERATORS**.

(1) Oscillation stabilization time select register (OSTS)

This register is used to select oscillation stabilization time of the clock supplied from the oscillator when the STOP mode is released. The wait time set by OSTS is valid only when the crystal/ceramic oscillation clock is selected as the system clock and after the STOP mode is released. If the high-speed Ring-OSC oscillator or external clock input is selected as the system clock source, no wait time elapses.

The system clock oscillator and the oscillation stabilization time that elapses after power application or release of reset are selected by the option byte. For details, refer to **CHAPTER 17 OPTION BYTE**.

OSTS is set by using the 8-bit memory manipulation instruction.

Figure 13-1. Format of Oscillation Stabilization Time Select Register (OSTS)

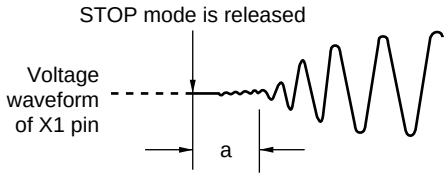
Address: FFF4H, After reset: Undefined, R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	0	OSTS1	OSTS0

OSTS1	OSTS0	Selection of oscillation stabilization time
0	0	$2^{10}/f_X$ (102.4 μs)
0	1	$2^{12}/f_X$ (409.6 μs)
1	0	$2^{15}/f_X$ (3.27 ms)
1	1	$2^{17}/f_X$ (13.1 ms)

Cautions 1. To set and then release the STOP mode, set the oscillation stabilization time as follows. Expected oscillation stabilization time of resonator ≤ Oscillation stabilization time set by OSTS

2. The wait time after the STOP mode is released does not include the time from the release of the STOP mode to the start of clock oscillation (“a” in the figure below), regardless of whether STOP mode was released by reset input or interrupt generation.



3. The oscillation stabilization time that elapses on power application or after release of reset is selected by the option byte. For details, refer to CHAPTER 17 OPTION BYTE.

Remarks 1. ( ): f_X = 10 MHz

2. Determine the oscillation stabilization time of the resonator by checking the characteristics of the resonator to be used.

13.2 Standby Function Operation

13.2.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction.

The operating statuses in the HALT mode are shown below.

Caution Because an interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set.

Table 13-2. Operating Statuses in HALT Mode

Item \ Setting of HALT Mode			When Low-speed Ring-OSC Oscillation Continues	When Low-speed Ring-OSC Oscillation Stops [Note 1]
System clock			Clock supply to CPU is stopped.	
CPU			Operation stops.	
Port (latch)			Holds status before HALT mode was set.	
16-bit timer/event counter 00			Operable	
8-bit timer 80			Operable	
8-bit timer H1	Sets count clock to f_{XP} to $f_{XP}/2^{12}$		Operable	
	Sets count clock to $f_{RL}/2^{7}$	Low-speed Ring-OSC cannot be stopped[Note 2].	Operable	–
		Low-speed Ring-OSC can be stopped[Note 2].		Operation stops.
Watchdog timer	"Clock to peripheral hardware" selected as operating clock		Operation stops.	
	"Low-speed Ring-OSC clock" selected as operating clock	Low-speed Ring-OSC cannot be stopped[Note 2].	Operable	–
		Low-speed Ring-OSC can be stopped[Note 2].	Operation stops.	
A/D converter			Operable	
Serial interface UART6			Operable	
Low-voltage detector			Operable	
External interrupt			Operable	

Notes 1. When "Stopped by software" is selected for low-speed Ring-OSC by the option byte and low-speed Ring-OSC is stopped by software (for the option byte, see **CHAPTER 17 OPTION BYTE**).

2. "Low-speed Ring-OSC cannot be stopped" or "low-speed Ring-OSC can be stopped by software" can be selected by the option byte.

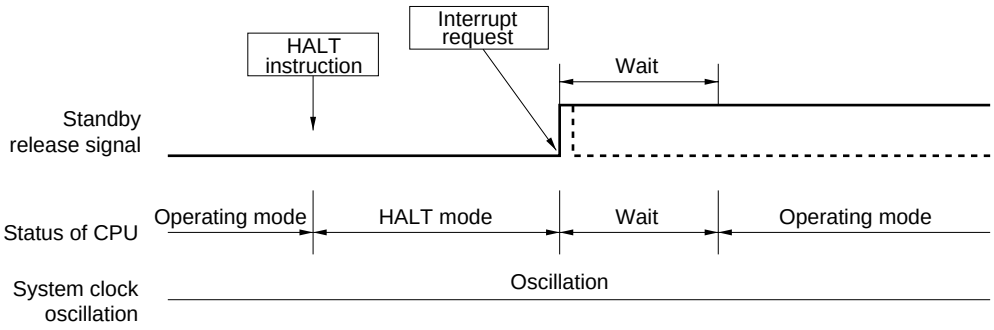
(2) HALT mode release

The HALT mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledgement is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgement is disabled, the next address instruction is executed.

Figure 13-2. HALT Mode Release by Interrupt Request Generation



Remarks 1. The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

2. The wait time is as follows:

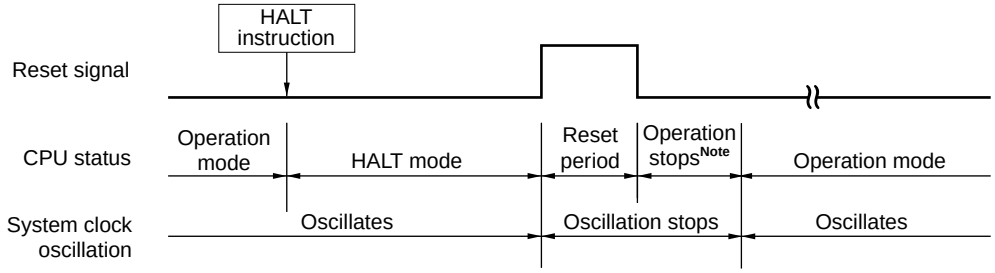
- When vectored interrupt servicing is carried out: 9 or 10 clocks
- When vectored interrupt servicing is not carried out: 1 or 2 clocks

(b) Release by reset input

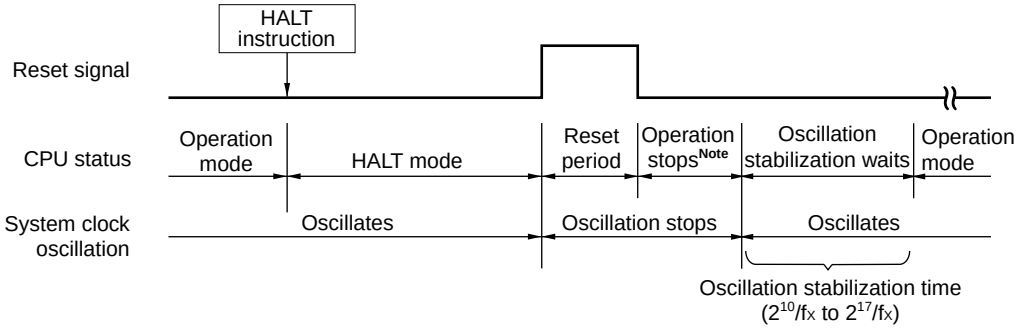
When the reset signal is input, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 13-3. HALT Mode Release by Reset Input

(1) When CPU clock is high-speed Ring-OSC clock or external input clock



(2) When CPU clock is crystal/ceramic oscillation clock



Note The operation is stopped ($8/f_{RL} + 96/f_{RH}$) because the option byte is referenced.

Remark f_X: System clock oscillation frequency
f_{RL}: Low-speed Ring-OSC clock oscillation frequency
f_{RH}: High-speed Ring-OSC clock oscillation frequency

Table 13-3. Operation in Response to Interrupt Request in HALT Mode

Release Source	MK××	IE	Operation
Maskable interrupt request	0	0	Next address instruction execution
	0	1	Interrupt servicing execution
	1	×	HALT mode held
Reset input	–	×	Reset processing

×: don't care

13.2.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction.

Caution Because an interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, in the STOP mode, the normal operation mode is restored after the STOP instruction is executed and then the operation is stopped for the duration of eight low-speed Ring-OSC clocks (after an additional wait time for stabilizing oscillation set by the oscillation stabilization time select register (OSTS) has elapsed when crystal/ceramic oscillation is used).

The operating statuses in the STOP mode are shown below.

Table 13-4. Operating Statuses in STOP Mode

Item \ Setting of STOP Mode			When Low-Speed Ring-OSC Oscillation Continues	When Low-Speed Ring-OSC Oscillation Stops[Note 1]
System clock			Oscillation stops.	
CPU			Operation stops.	
Port (latch)			Holds status before STOP mode is set.	
16-bit timer/event counter 00			Operation stops.	
8-bit timer 80			Operation stops.	
8-bit timer H1	Sets count clock to f_{XP} to $f_{XP}/2^{12}$		Operation stops.	
	Sets count clock to $f_{RL}/2^{7}$	Low-speed Ring-OSC cannot be stopped[Note 2].	Operable	–
		Low-speed Ring-OSC can be stopped[Note 2].		Operation stops.
Watchdog timer	"Clock to peripheral hardware" selected as operating clock		Operation stops.	
	"Low-speed Ring-OSC clock" selected as operating clock	Low-speed Ring-OSC cannot be stopped[Note 2].	Operable	–
		Low-speed Ring-OSC can be stopped[Note 2].	Operation stops.	
A/D converter			Operation stops.	
Serial interface UART6			Operation stops.	
Low-voltage detector			Operable	
External interrupt			Operable	

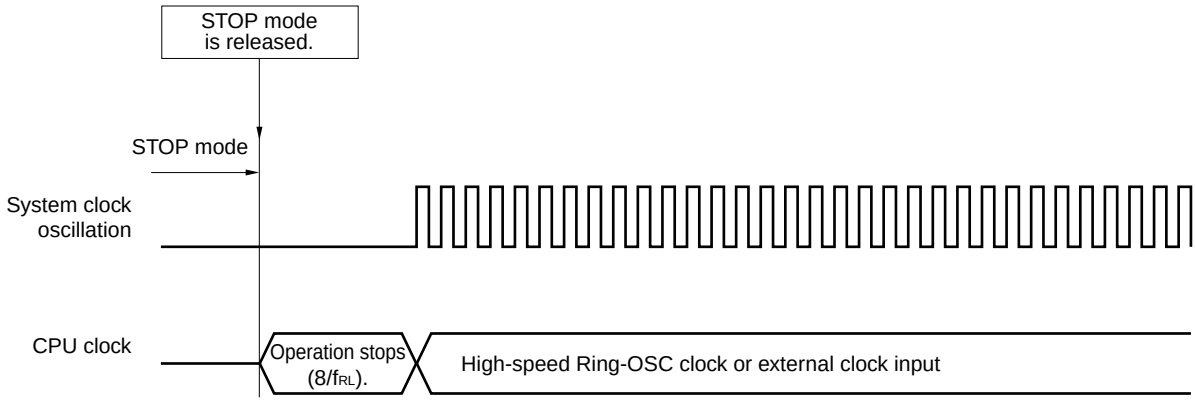
Notes 1. When "Stopped by software" is selected for low-speed Ring-OSC by the option byte the low-speed Ring-OSC is stopped by software (for the option byte, see **CHAPTER 17 OPTION BYTE**).

2. "Low-speed Ring-OSC cannot be stopped" or "low-speed Ring-OSC can be stopped by software" can be selected by the option byte.

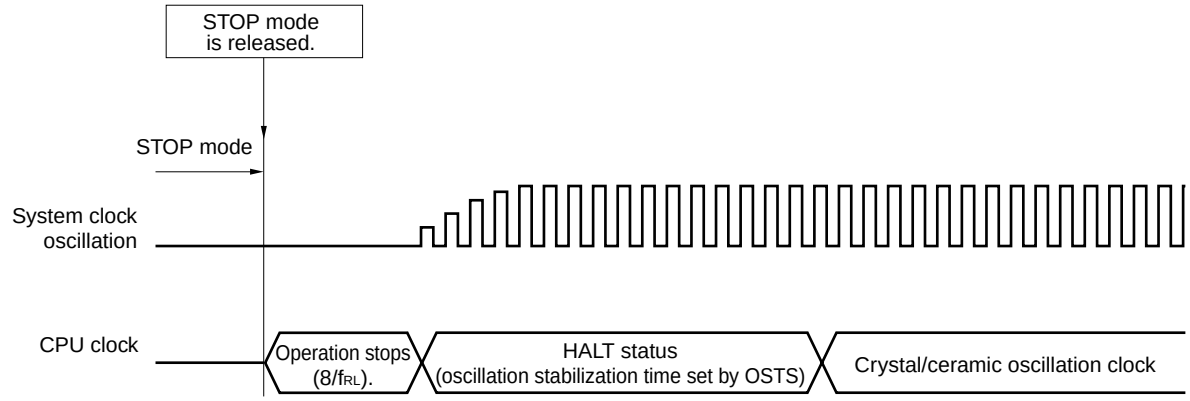
(2) STOP mode release

Figure 13-4. Operation Timing When STOP Mode Is Released

<1> If high-speed Ring-OSC clock or external input clock is selected as system clock to be supplied



<2> If crystal/ceramic oscillation clock is selected as system clock to be supplied



Remark f_{RL}: Low-speed Ring-OSC clock oscillation frequency

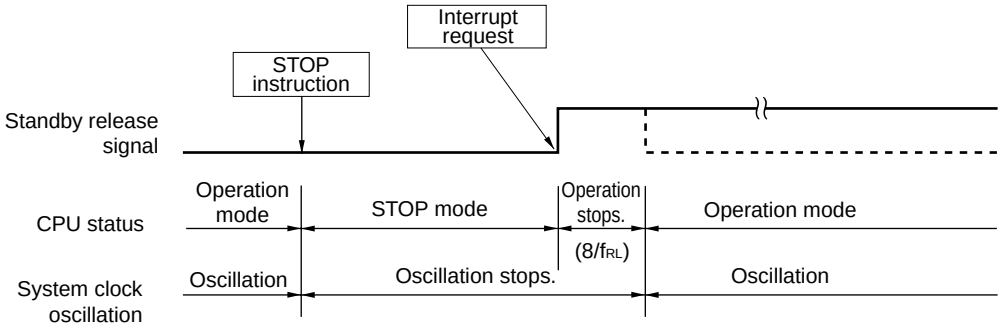
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

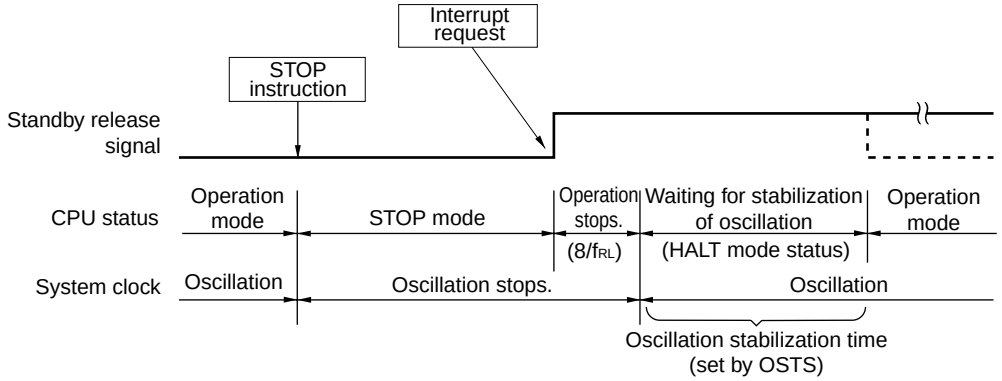
When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 13-5. STOP Mode Release by Interrupt Request Generation

(1) If CPU clock is high-speed Ring-OSC clock or external input clock



(2) If CPU clock is crystal/ceramic oscillation clock



Remarks 1. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

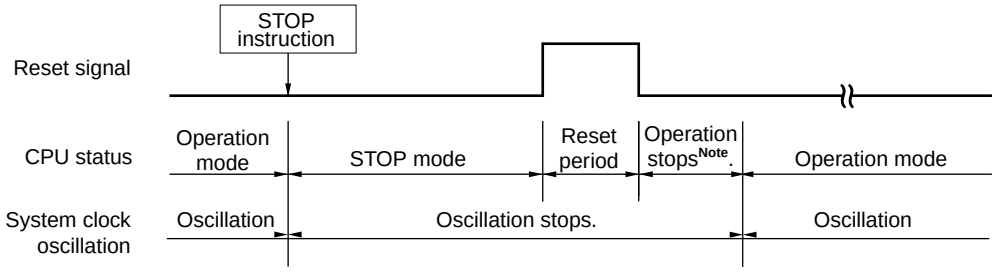
2. f_{RL}: Low-speed Ring-OSC clock oscillation frequency

(b) Release by reset input

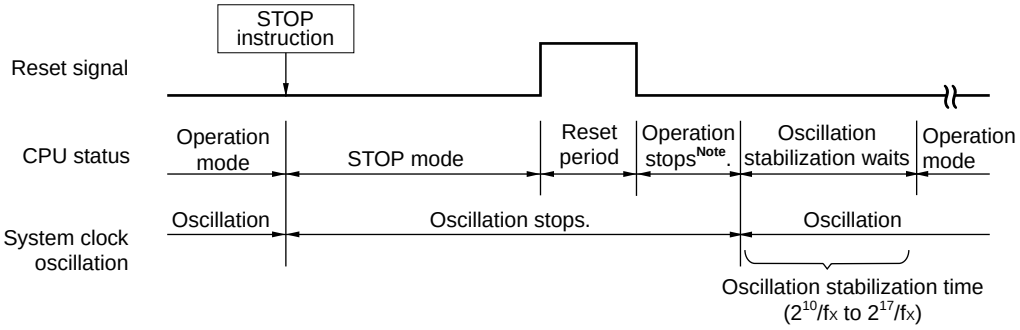
When the reset signal is input, STOP mode is released and a reset operation is performed after the oscillation stabilization time has elapsed.

Figure 13-6. STOP Mode Release by Reset Input

(1) If CPU clock is high-speed Ring-OSC clock or external input clock



(2) If CPU clock is crystal/ceramic oscillation clock



Note The operation is stopped ($8/f_{RL} + 96/f_{RH}$) because the option byte is referenced.

Remark f_X: System clock oscillation frequency
f_{RL}: Low-speed Ring-OSC clock oscillation frequency
f_{RH}: High-speed Ring-OSC clock oscillation frequency

Table 13-5. Operation in Response to Interrupt Request in STOP Mode

Release Source	MK××	IE	Operation
Maskable interrupt request	0	0	Next address instruction execution
	0	1	Interrupt servicing execution
	1	×	STOP mode held
Reset input	–	×	Reset processing

×: don't care

CHAPTER 14 RESET FUNCTION

The following four operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop detection
(3) Internal reset by comparison of supply voltage and detection voltage of power-on-clear (POC) circuit
(4) Internal reset by comparison of supply voltage and detection voltage of low-power-supply detector (LVI)

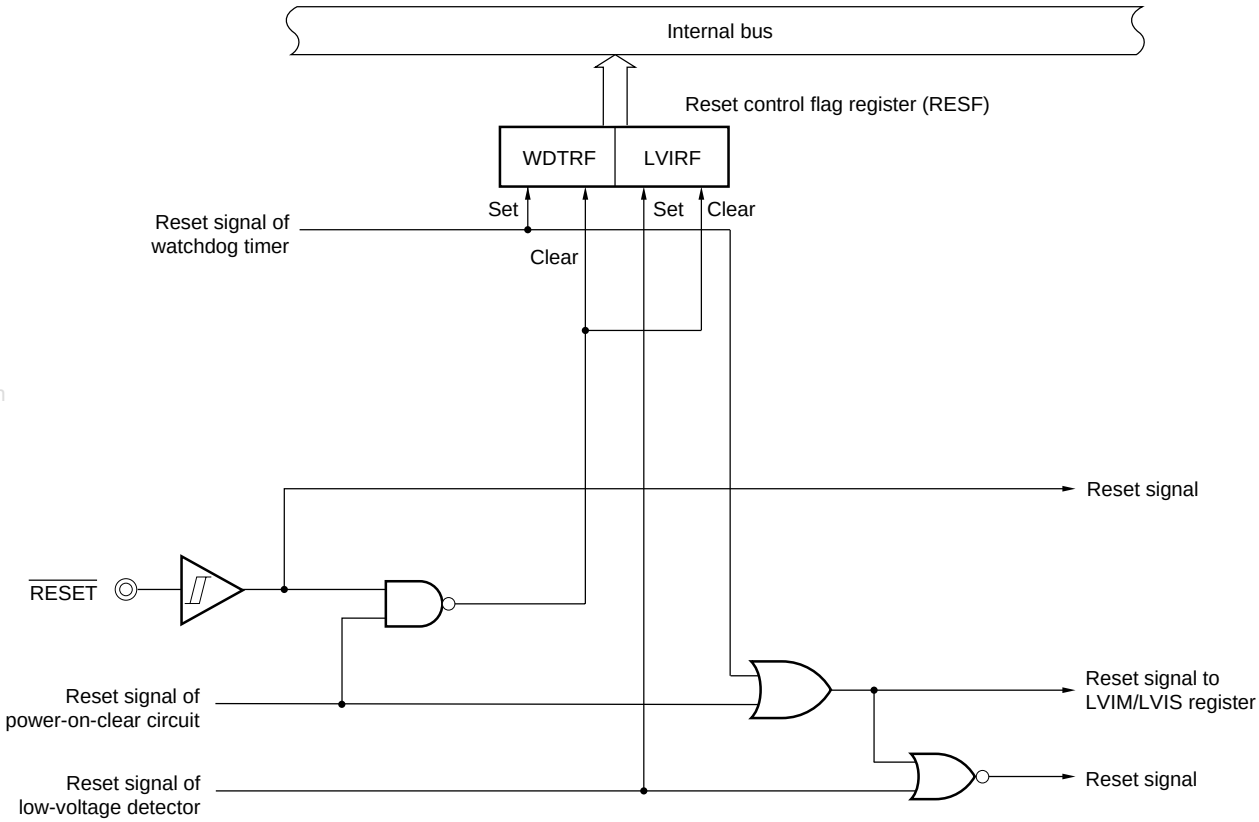
External and internal resets have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H when the reset signal is input.

A reset is applied when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, or by POC and LVI circuit voltage detection, and each item of hardware is set to the status shown in Table 14-1. Each pin is high impedance during reset input or during the oscillation stabilization time just after reset release, except for P130, which is low-level output.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is released and program execution starts using the CPU clock after referencing the option byte (after the option byte is referenced and the clock oscillation stabilization time elapses if crystal/ceramic oscillation is selected). A reset generated by the watchdog timer source is automatically released after the reset, and program execution starts using the CPU clock after referencing the option byte (after the option byte is referenced and the clock oscillation stabilization time elapses if crystal/ceramic oscillation is selected). (see **Figures 14-2** to **14-4**). Reset by POC and LVI circuit power supply detection is automatically released when $V_{DD} > V_{POC}$ or $V_{DD} > V_{LVI}$ after the reset, and program execution starts using the CPU clock after referencing the option byte (after the option byte is referenced and the clock oscillation stabilization time elapses if crystal/ceramic oscillation is selected) (see **CHAPTER 15 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 16 LOW-VOLTAGE DETECTOR**).

Cautions 1. For an external reset, input a low level for 1 μs or more to the $\overline{\text{RESET}}$ pin.
2. During reset input, the system clock and low-speed Ring-OSC clock stop oscillating.
3. When the $\overline{\text{RESET}}$ pin is used as an input-only port pin (P34), the 78K0S/KA1+ is reset if a low level is input to the $\overline{\text{RESET}}$ pin after reset is released by the POC circuit and before the option byte is referenced again. The reset status is retained until a high level is input to the $\overline{\text{RESET}}$ pin.

Figure 14-1. Block Diagram of Reset Function



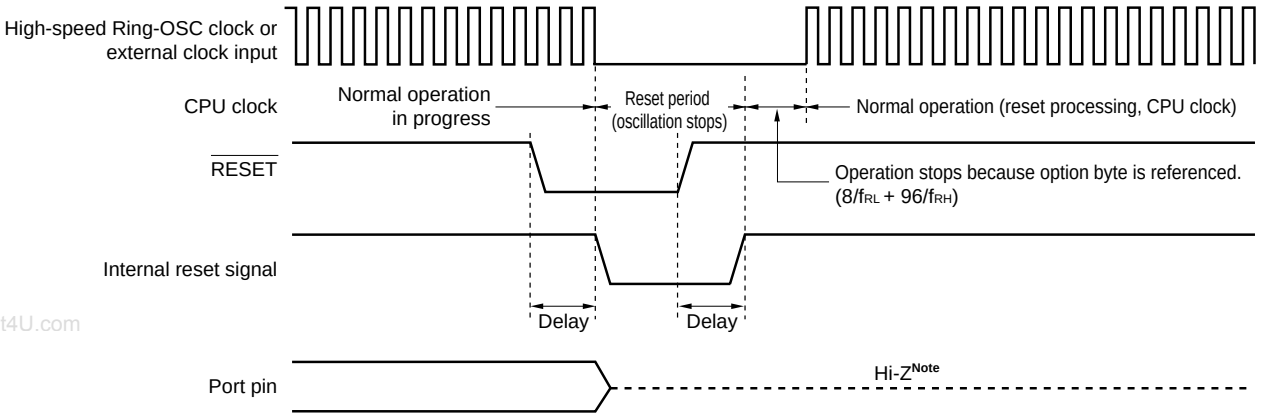
Caution The LVI circuit is not reset by the internal reset signal of the LVI circuit.

Remarks 1. LVIM: Low-voltage detect register

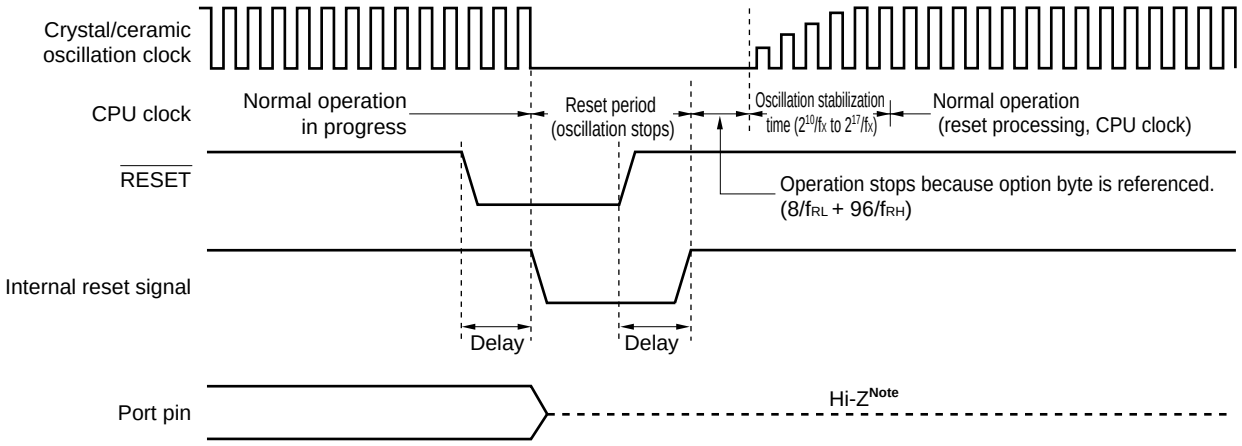
2. LVIS: Low-voltage detection level select register

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

<1> With high-speed Ring-OSC clock or external clock input



<2> With crystal/ceramic oscillation clock

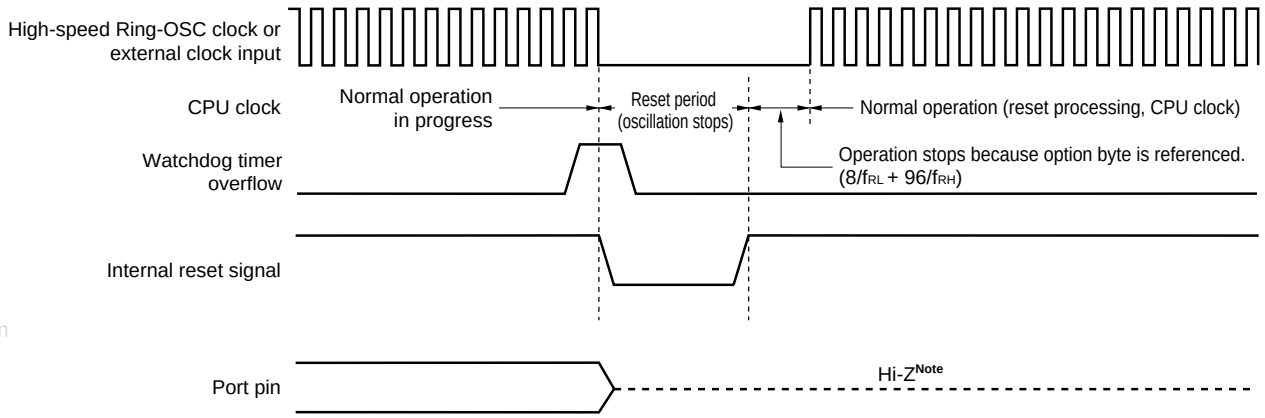


Note P130 outputs a low level, and the other port pins go into a high-impedance state.

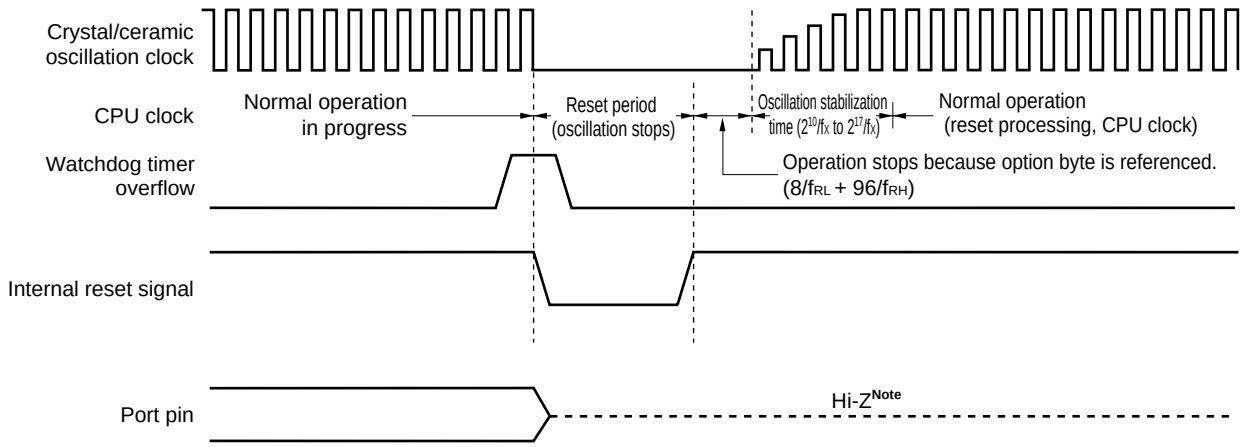
Remark f_X: System clock oscillation frequency
f_{RL}: Low-speed Ring-OSC clock oscillation frequency
f_{RH}: High-speed Ring-OSC clock oscillation frequency

Figure 14-3. Timing of Reset by Overflow of Watchdog Timer

<1> With high-speed Ring-OSC clock or external clock input



<2> With crystal/ceramic oscillation clock



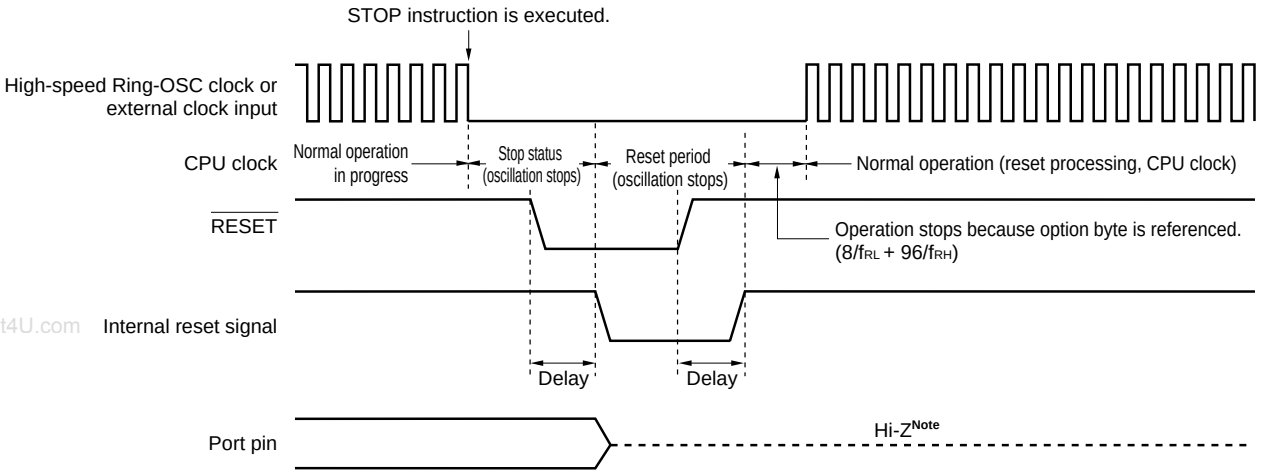
Note P130 outputs a low level, and the other port pins go into a high-impedance state.

Caution The watchdog timer is also reset in the case of an internal reset of the watchdog timer.

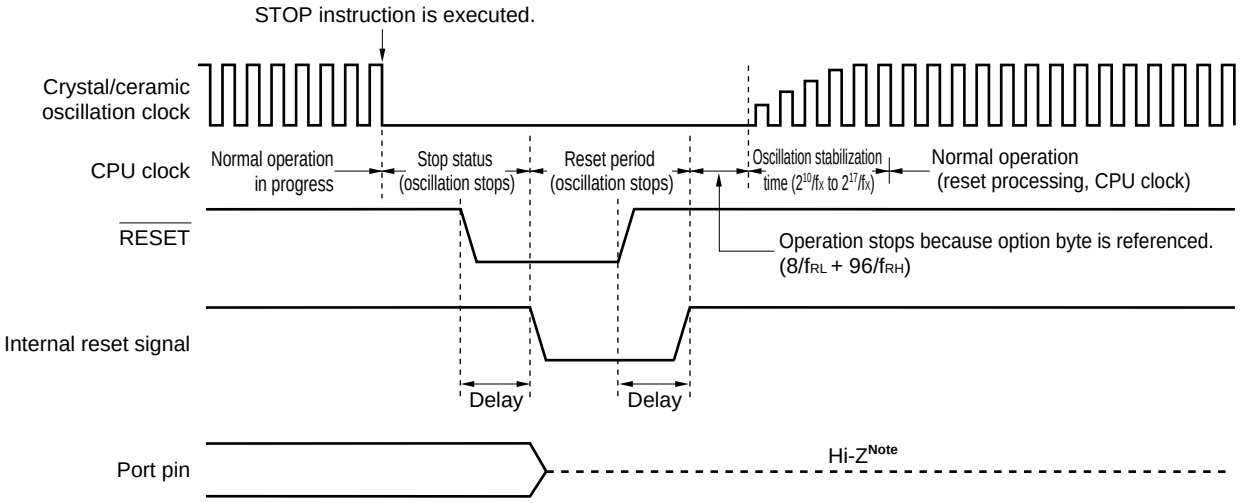
Remark f_X: System clock oscillation frequency
f_{RL}: Low-speed Ring-OSC clock oscillation frequency
f_{RH}: High-speed Ring-OSC clock oscillation frequency

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

<1> With high-speed Ring-OSC clock or external clock input



<2> With crystal/ceramic oscillation clock



Note P130 outputs a low level, and the other port pins go into a high-impedance state.

Remarks 1. For the reset timing of the power-on-clear circuit and low-voltage detector, refer to **CHAPTER 15 POWER-ON-CLEAR CIRCUIT and CHAPTER 16 LOW-VOLTAGE DETECTOR**.

2. f_X: System clock oscillation frequency

f_{RL}: Low-speed Ring-OSC clock oscillation frequency

f_{RH}: High-speed Ring-OSC clock oscillation frequency

Table 14-1. Hardware Statuses After Reset Acknowledgment (1/2)

Hardware		Status After Reset
Program counter (PC) Note 1		Contents of reset vector table (0000H and 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined Note 2
	General-purpose registers	Undefined Note 2
Ports (P2 to P4, P12, P13) (output latches)		00H
Port mode registers (PM2 to PM4, PM12)		FFH
Port mode control register (PMC2)		00H
Pull-up resistor option registers (PU2, PU3, PU4, PU12)		00H
Processor clock control register (PCC)		02H
Preprocessor clock control register (PPCC)		02H
Low-speed Ring-OSC mode register (LSRCM)		00H
High-speed Ring-OSC mode register (HSRCM)		00H
Oscillation stabilization time select register (OSTS)		Undefined
16-bit timer 00	Timer counter 00 (TM00)	0000H
	Capture/compare registers 000, 010 (CR000, CR010)	0000H
	Mode control register 00 (TMC00)	00H
	Prescaler mode register 00 (PRM00)	00H
	Capture/compare control register 00 (CRC00)	00H
	Timer output control register 00 (TOC00)	00H
8-bit timer 80	Timer counter 80 (TM80)	00H
	Compare register (CR80)	Undefined
	Mode control register 80 (TMC80)	00H
8-bit timer H1	Compare registers (CMP01, CMP11)	00H
	Mode register 1 (TMHMD1)	00H
Watchdog timer	Mode register (WDTM)	67H
	Enable register (WDTE)	9AH
A/D converter	Conversion result registers (ADCR, ADCRH)	Undefined
	Mode register (ADM)	00H
	Analog input channel specification register (ADS)	00H

Notes 1. Only the contents of PC are undefined while reset is being input and while the oscillation stabilization time elapses. The statuses of the other hardware units remain unchanged.

2. The status after reset is held in the standby mode.

Table 14-1. Hardware Statuses After Reset Acknowledgment (2/2)

Hardware		Status After Reset
Serial interface UART6	Receive buffer register 6 (RXB6)	FFH
	Transmit buffer register 6 (TXB6)	FFH
	Asynchronous serial interface operation mode register 6 (ASIM6)	01H
	Asynchronous serial interface reception error status register 6 (ASIS6)	00H
	Asynchronous serial interface transmission error status register 6 (ASIF6)	00H
	Clock select register 6 (CKSR6)	00H
	Baud rate generator control register 6 (BRGC6)	FFH
	Asynchronous serial interface control register 6 (ASICL6)	16H
	Input select control register (ISC)	00H
Reset function	Reset control flag register (RESF)	00H[Note]
Low-voltage detector	Low-voltage detection register (LVIM)	00H[Note]
	Low-voltage detection level select register (LVIS)	00H[Note]
Interrupt	Request flag registers (IF0, IF1)	00H
	Mask flag registers (MK0, MK1)	FFH
	External interrupt mode registers (INTM0, INTM1)	00H

Note These values change as follows depending on the reset source.

Register \ Reset Source	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by LVI
RESF	See **Table 14-2**.			
LVIM	Cleared (00H)	Cleared (00H)	Cleared (00H)	Held
LVIS				

14.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the 78K0S/KA1+. The reset control flag register (RESF) is used to store which source has generated the reset request.

RESF can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input, reset input by power-on-clear (POC) circuit, and reading RESF clear RESF to 00H.

Figure 14-5. Format of Reset Control Flag Register (RESF)

Address: FF54H After reset: 00H[Note] R

Symbol	7	6	5	4	3	2	1	0
RESF	0	0	0	WDTRF	0	0	0	LVIRF

WDTRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by low-voltage detector (LVI)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

Note The value after reset varies depending on the reset source.

Caution Do not read data by a 1-bit memory manipulation instruction.

The status of RESF when a reset request is generated is shown in Table 14-2.

Table 14-2. RESF Status When Reset Request Is Generated

Flag \ Reset Source	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by LVI
WDTRF	Cleared (0)	Cleared (0)	Set (1)	Held
LVIRF			Held	Set (1)

CHAPTER 15 POWER-ON-CLEAR CIRCUIT

15.1 Functions of Power-on-Clear Circuit

The power-on-clear circuit (POC) has the following functions.

- Generates internal reset signal at power on.
- Compares supply voltage (V_{DD}) and detection voltage (V_{POC} = 2.1 V ±0.1 V), and generates internal reset signal when $V_{DD} < V_{POC}$.

Cautions 1. If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

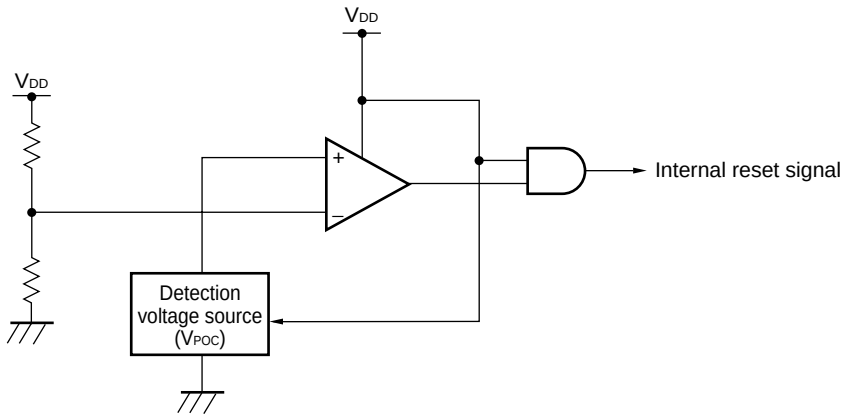
2. Because the detection voltage (V_{POC}) of the POC circuit is in a range of 2.1 V ±0.1 V, use a voltage in the range of 2.2 to 5.5 V.

Remark This product incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset cause is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT) or low-voltage-detection (LVI) circuit. RESF is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by WDT or LVI.
For details of RESF, see **CHAPTER 14 RESET FUNCTION**.

15.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 15-1.

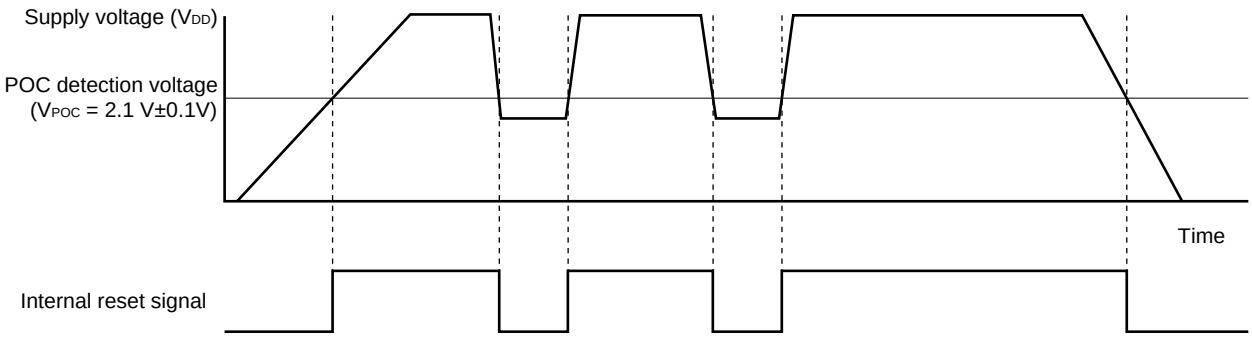
Figure 15-1. Block Diagram of Power-on-Clear Circuit



15.3 Operation of Power-on-Clear Circuit

In the power-on-clear circuit, the supply voltage (V_{DD}) and detection voltage (V_{POC} = 2.1 V ±0.1 V) are compared, and when $V_{DD} < V_{POC}$, an internal reset signal is generated.

Figure 15-2. Timing of Internal Reset Signal Generation in Power-on-Clear Circuit



15.4 Cautions for Power-on-Clear Circuit

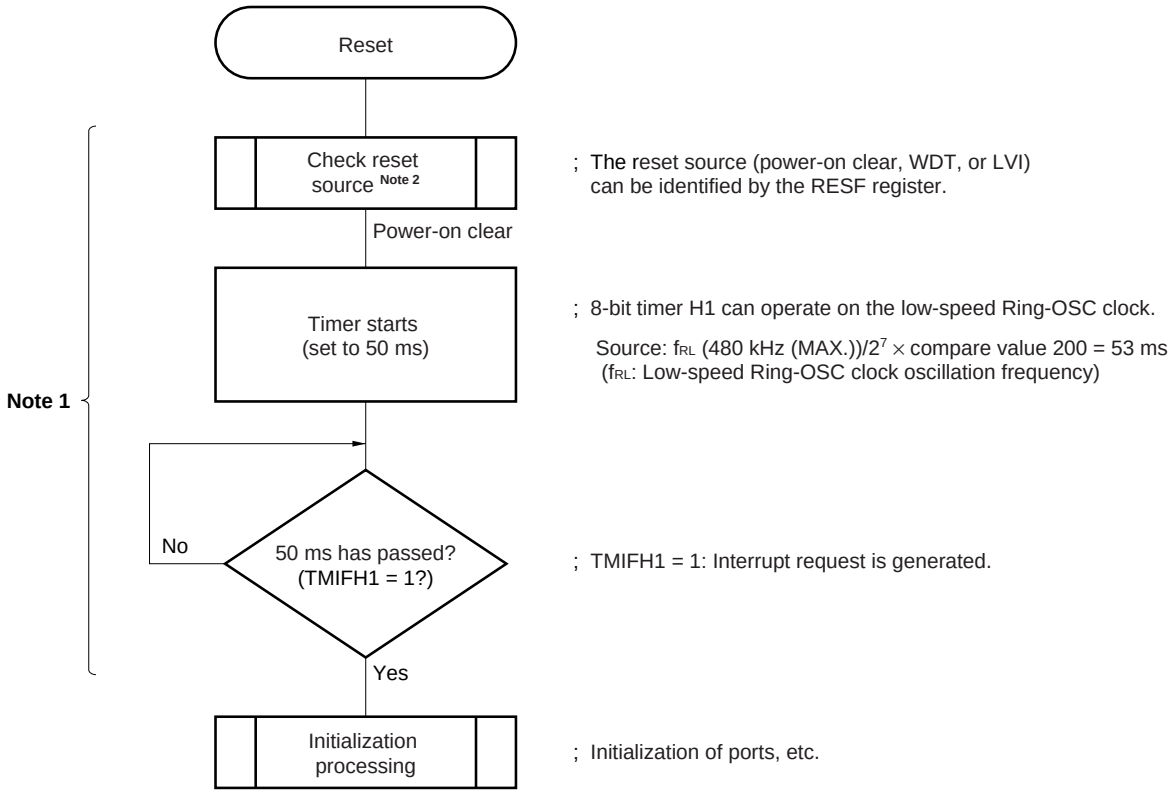
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POC}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 15-3. Example of Software Processing After Release of Reset (1/2)

• If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage

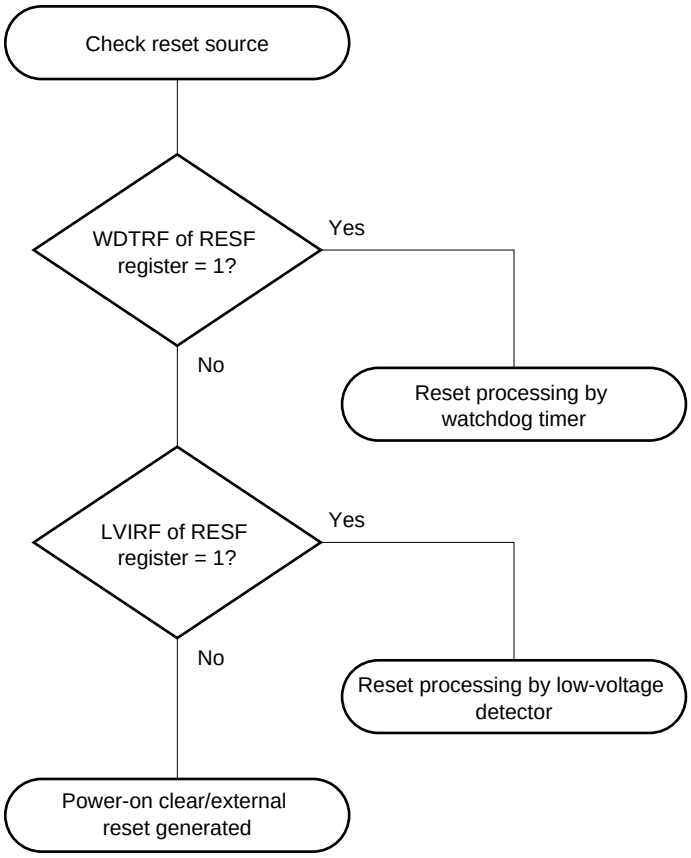


Notes 1. If reset is generated again during this period, initialization processing is not started.

2. A flowchart is shown on the next page.

Figure 15-3. Example of Software Processing After Release of Reset (2/2)

- Checking reset cause



CHAPTER 16 LOW-VOLTAGE DETECTOR

16.1 Functions of Low-Voltage Detector

The low-voltage detector (LVI) has following functions.

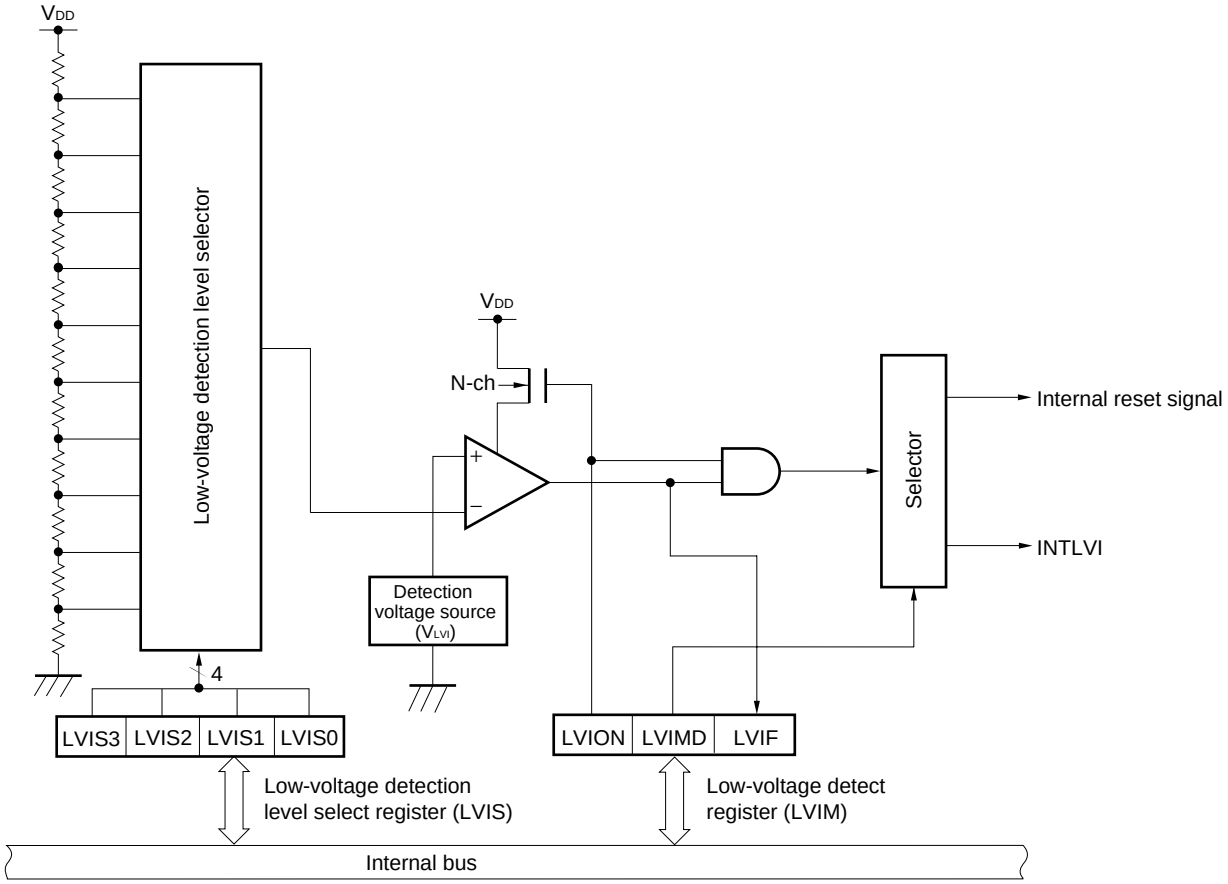
- Compares supply voltage (V_{DD}) and detection voltage (V_{LVI}), and generates an internal interrupt signal or internal reset signal when $V_{DD} < V_{LVI}$.
- Detection levels (ten levels) of supply voltage can be changed by software.
- Interrupt or reset function can be selected by software.
- Operable in STOP mode.

When the low-voltage detector is used to reset, bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of RESF, refer to **CHAPTER 14 RESET FUNCTION**.

16.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown in Figure 16-1.

Figure 16-1. Block Diagram of Low-Voltage Detector



16.3 Registers Controlling Low-Voltage Detector

The low-voltage detector is controlled by the following registers.

- Low-voltage detect register (LVIM)
- Low-voltage detection level select register (LVIS)

(1) Low-voltage detect register (LVIM)

This register sets low-voltage detection and the operation mode.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset input clears this register to 00H.

Figure 16-2. Format of Low-Voltage Detect Register (LVIM)

Address: FF50H After reset: 00H R/W[Note 1]

Symbol	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVION	0	0	0	0	0	LVIMD	LVIF

LVION[Notes 2, 3]	Enabling low-voltage detection operation
0	Disable operation
1	Enable operation

LVIMD[Note 2]	Low-voltage detection operation mode selection
0	Generate interrupt signal when supply voltage (V_{DD}) < detection voltage (V_{LVI})
1	Generate internal reset signal when supply voltage (V_{DD}) < detection voltage (V_{LVI})

LVIF[Note 4]	Low-voltage detection flag
0	Supply voltage (V_{DD}) > detection voltage (V_{LVI}), or when operation is disabled
1	Supply voltage (V_{DD}) < detection voltage (V_{LVI})

Notes 1. Bit 0 is a read-only bit.

2. LVION, and LVIMD are cleared to 0 at a reset other than an LVI reset. These are not cleared to 0 at an LVI reset.

3. When LVION is set to 1, operation of the comparator in the LVI circuit is started. Use software to instigate a wait of at least 0.2 ms from when LVION is set to 1 until the voltage is confirmed at LVIF.

4. The value of LVIF is output as the interrupt request signal INTLVI when LVION = 1 and LVIMD = 0.

Cautions 1. To stop LVI, follow either of the procedures below.

- **When using 8-bit manipulation instruction: Write 00H to LVIM.**
- **When using 1-bit memory manipulation instruction: Clear LVION to 0.**

2. Be sure to set bits 2 to 6 to 0.

(2) Low-voltage detection level select register (LVIS)

This register selects the low-voltage detection level.

This register can be set by an 8-bit memory manipulation instruction.

Reset input clears this register to 00H.

Figure 16-3. Format of Low-Voltage Detection Level Select Register (LVIS)

Address: FF51H, After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0

LVIS3	LVIS2	LVIS1	LVIS0	Detection level
0	0	0	0	V_{LVI0} (4.3 V ±0.2 V)
0	0	0	1	V_{LVI1} (4.1 V ±0.2 V)
0	0	1	0	V_{LVI2} (3.9 V ±0.2 V)
0	0	1	1	V_{LVI3} (3.7 V ±0.2 V)
0	1	0	0	V_{LVI4} (3.5 V ±0.2 V)
0	1	0	1	V_{LVI5} (3.3 V ±0.15 V)
0	1	1	0	V_{LVI6} (3.1 V ±0.15 V)
0	1	1	1	V_{LVI7} (2.85 V ±0.15 V)
1	0	0	0	V_{LVI8} (2.6 V ±0.15 V)
1	0	0	1	V_{LVI9} (2.35 V ±0.15 V)
Other than above				Setting prohibited

Caution Bits 4 to 7 must be set to 0.

16.4 Operation of Low-Voltage Detector

The low-voltage detector can be used in the following two modes.

- Used as reset
  Compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}), and generates an internal reset signal when $V_{DD} < V_{LVI}$.
- Used as interrupt
  Compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}), and generates an interrupt signal (INTLVI) when $V_{DD} < V_{LVI}$.

The operation is set as follows.

(1) When used as reset

- When starting operation

<1> Mask the LVI interrupt (LVIMK = 1).

<2> Set the detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level select register (LVIS).

<3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<4> Use software to instigate a wait of at least 0.2 ms.

<5> Wait until "supply voltage (V_{DD}) > detection voltage (V_{LVI})" at bit 0 (LVIF) of LVIM is confirmed.

<6> Set bit 1 (LVIMD) of LVIM to 1 (generates internal reset signal when supply voltage (V_{DD}) < detection voltage (V_{LVI})).

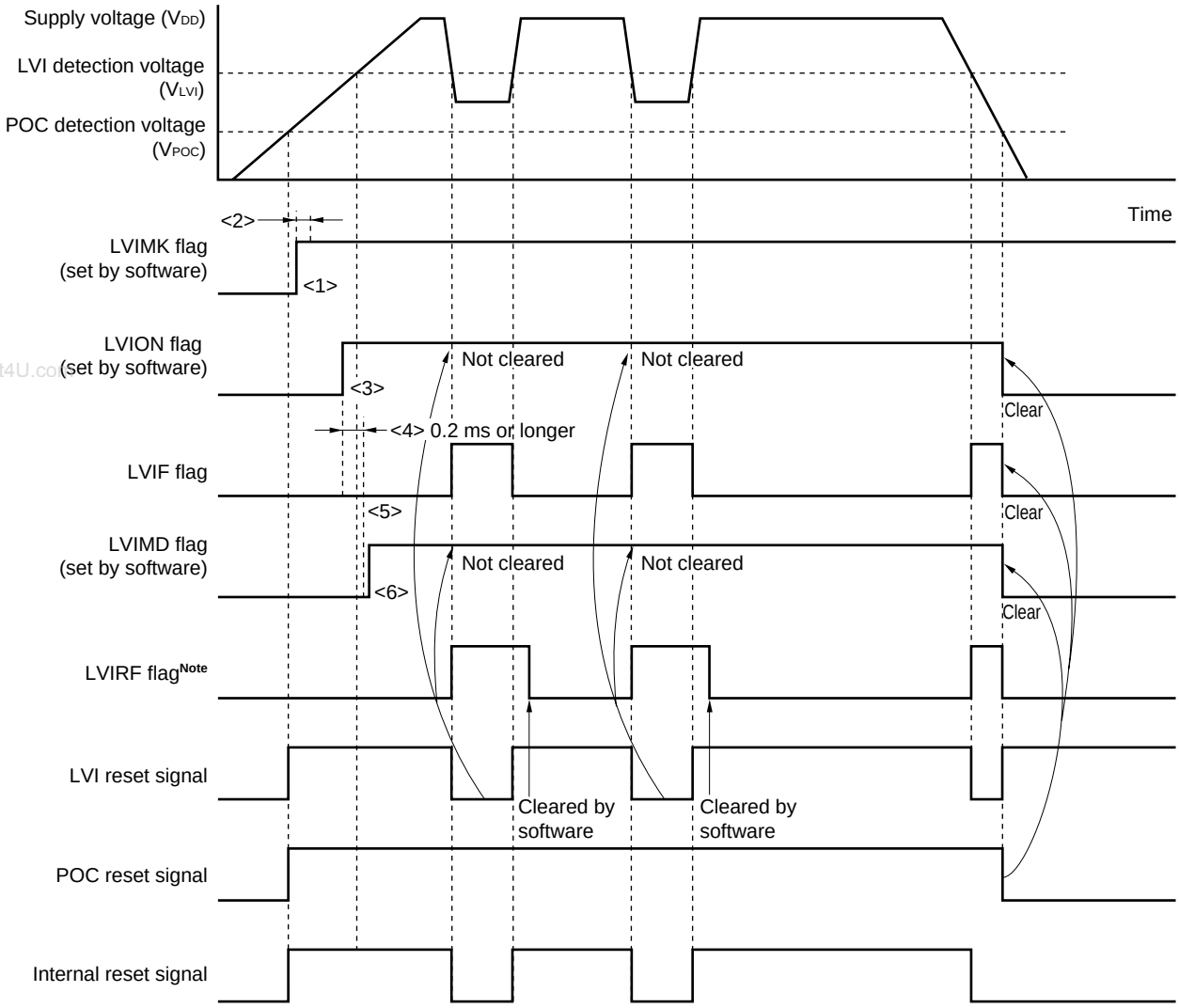
Figure 16-4 shows the timing of generating the internal reset signal of the low-voltage detector. Numbers <1> to <6> in this figure correspond to <1> to <6> above.

Cautions 1. <1> must always be executed. When LVIMK = 0, an interrupt may occur immediately after the processing in <3>.

2. If supply voltage (V_{DD}) > detection voltage (V_{LVI}) when LVIM is set to 1, an internal reset signal is not generated.

- When stopping operation
  Either of the following procedures must be executed.
  - When using 8-bit memory manipulation instruction: Write 00H to LVIM.
  - When using 1-bit memory manipulation instruction: Clear LVIMD to 0 and LVION to 0 in that order.

Figure 16-4. Timing of Low-Voltage Detector Internal Reset Signal Generation



Note LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, refer to **CHAPTER 14 RESET FUNCTION**.

Remark <1> to <6> in Figure 16-4 above correspond to <1> to <6> in the description of “when starting operation” in **16.4 (1) When used as reset**.

(2) When used as interrupt

- When starting operation

<1> Mask the LVI interrupt (LVIMK = 1).

<2> Set the detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level select register (LVIS).

<3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<4> Use software to instigate a wait of at least 0.2 ms.

<5> Wait until "supply voltage (V_{DD}) > detection voltage (V_{LVI})" at bit 0 (LVIF) of LVIM is confirmed.

<6> Clear the interrupt request flag of LVI (LVIIF) to 0.

<7> Release the interrupt mask flag of LVI (LVIMK).

<8> Execute the EI instruction (when vector interrupts are used).

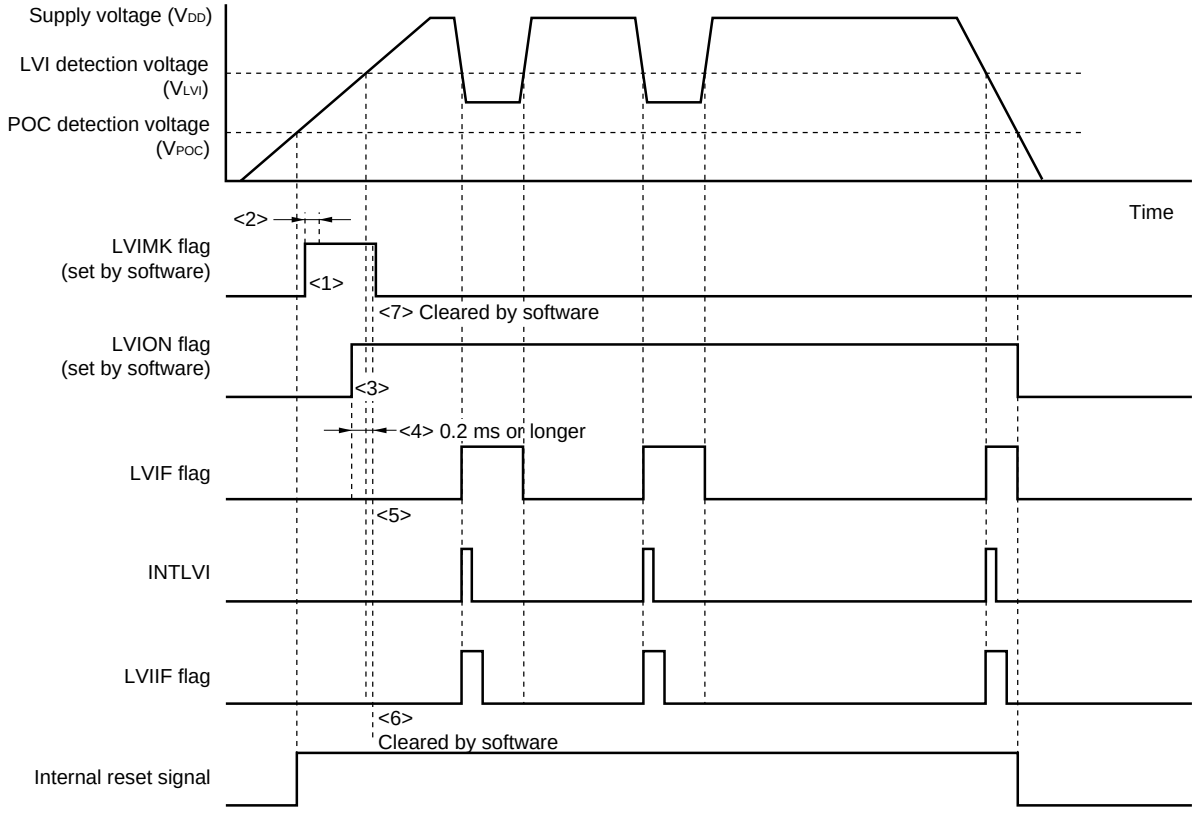
Figure 16-5 shows the timing of generating the interrupt signal of the low-voltage detector. Numbers <1> to <7> in this figure correspond to <1> to <7> above.

- When stopping operation

  Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction: Write 00H to LVIM.
  - When using 1-bit memory manipulation instruction: Clear LVION to 0.

Figure 16-5. Timing of Low-Voltage Detector Interrupt Signal Generation



Remark <1> to <7> in Figure 16-5 above correspond to <1> to <7> in the description of "when starting operation" in **16.4 (2) When used as interrupt**.

16.5 Cautions for Low-Voltage Detector

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVI detection voltage (V_{LVI}), the operation is as follows depending on how the low-voltage detector is used.

<1> When used as reset

The system may be repeatedly reset and released from the reset status.

In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking action (1) below.

<2> When used as interrupt

Interrupt requests may be frequently generated. Take action (2) below.

In this system, take the following actions.

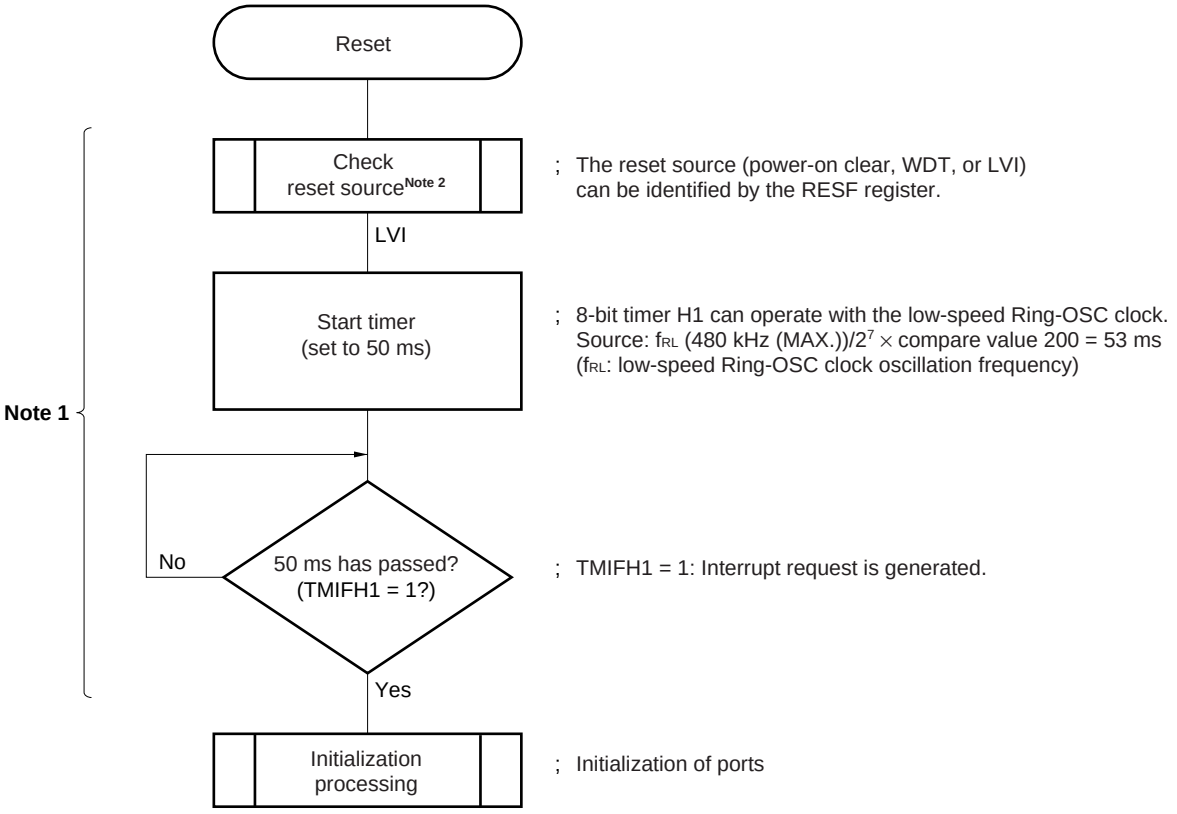
<Action>

(1) When used as reset

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 16-6. Example of Software Processing After Release of Reset (1/2)

• If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage

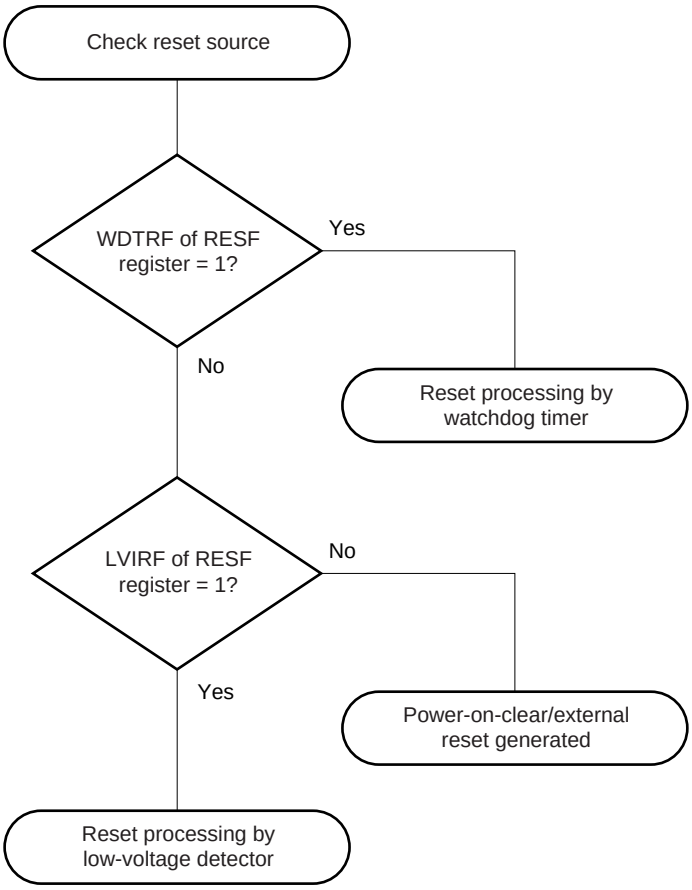


Notes 1. If reset is generated again during this period, initialization processing is not started.

2. A flowchart is shown on the next page.

Figure 16-6. Example of Software Processing After Release of Reset (2/2)

• Checking reset source



(2) When used as interrupt

Check that “supply voltage (V_{DD}) > detection voltage (V_{LVI})” in the servicing routine of the LVI interrupt by using bit 0 (LVIF) of the low-voltage detection register (LVIM). Clear bit 1 (LVIIF) of interrupt request flag register 0 (IF0) to 0 and enable interrupts (EI).

In a system where the supply voltage fluctuation period is long in the vicinity of the LVI detection voltage, wait for the supply voltage fluctuation period, check that “supply voltage (V_{DD}) > detection voltage (V_{LVI})” using the LVIF flag, and then enable interrupts (EI).

CHAPTER 17 OPTION BYTE

The 78K0S/KA1+ has an area called an option byte at address 0080H of the flash memory. When using the product, be sure to set the following functions by using the option byte.

1. Selection of system clock source

- High-speed Ring-OSC clock
- Crystal/ceramic oscillation clock
- External clock input

2. Low-speed Ring-OSC clock oscillation

- Cannot be stopped.
- Can be stopped by software.

3. Control of $\overline{\text{RESET}}$ pin

- Used as $\overline{\text{RESET}}$ pin
- $\overline{\text{RESET}}$ pin is used as an input port pin (P34).

4. Oscillation stabilization time on power application or reset input

- $2^{10}/f_X$
- $2^{12}/f_X$
- $2^{15}/f_X$
- $2^{17}/f_X$

Figure 17-1. Positioning of Option Byte

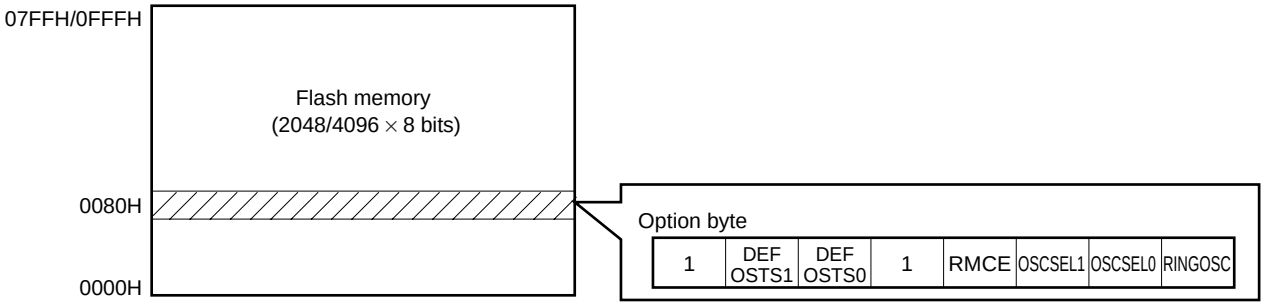


Figure 17-2. Format of Option Byte (1/2)

Address: FF80H

7	6	5	4	3	2	1	0
1	DEFOSTS1	DEFOSTS0	1	RMCE	OSCSEL1	OSCSEL0	RINGOSC

RINGOSC	Low-speed Ring-OSC clock oscillation
1	Cannot be stopped
0	Can be stopped by software

Cautions 1. If it is selected that low-speed Ring-OSC clock oscillation cannot be stopped, the count clock to the watchdog timer (WDT) is fixed to low-speed Ring-OSC.

2. If it is selected that low-speed Ring-OSC can be stopped by software, supply of the count clock to WDT is stopped in the HALT/STOP mode, regardless of the setting of bit 0 (LSRSTOP) of the low-speed Ring-OSC mode register (LSRCM). If low-speed Ring-OSC is selected as the count clock to 8-bit timer H1, however, the count clock is supplied in the HALT/STOP mode while low-speed Ring-OSC operates (LSRSTOP = 0).

OSCSEL1	OSCSEL0	Selection of system clock source
0	0	Crystal/ceramic oscillation clock
0	1	External clock input
1	×	High-speed Ring-OSC clock

Caution Because the X1 and X2 pins are also used as the P121 and P122 pins, the conditions under which the X1 and X2 pins can be used as port pins differ depending on the selected system clock source.

(1) High-speed Ring-OSC clock

P121 and P122 can be used as I/O port pins.

(2) Crystal/ceramic oscillation clock

The X1 and X2 pins cannot be used as I/O port pins because they are used as clock input pins.

(3) External clock input

Because the X1 pin is used as an external clock input pin, P121 cannot be used as an I/O port pin.

Remark × : don't care

RMCE	Control of $\overline{\text{RESET}}$ pin
1	$\overline{\text{RESET}}$ pin is used as is.
0	$\overline{\text{RESET}}$ pin is used as input port pin (P34).

Caution If a low level is input to the $\overline{\text{RESET}}$ pin after reset is released by the power-on clear function and before the option byte is referenced again, the 78K0S/KA1+ is reset, and the status is held until a high level is input to the $\overline{\text{RESET}}$ pin.

Figure 17-2. Format of Option Byte (2/2)

DEFOSTS1	DEFOSTS0	Oscillation stabilization time on power application or reset input
0	0	2^{10}/fx (102.4 μs)
0	1	2^{12}/fx (409.6 μs)
1	0	2^{15}/fx (3.27 ms)
1	1	2^{17}/fx (13.1 ms)

Caution The setting of this option is valid only when the crystal/ceramic oscillation clock is selected as the system clock source. No wait time elapses if the high-speed Ring-OSC or external clock input is selected as the system clock source.

Remarks 1. ( ): fx = 10 MHz

2. For the oscillation stabilization time of the resonator, refer to the characteristics of the resonator to be used.

CHAPTER 18 FLASH MEMORY

Flash memory versions are commonly used in the following development environments and mass production applications.

- For altering software after the 78K0S/KA1+ is soldered onto the target system.
- For data adjustment when starting mass production.
- For differentiating software according to the specification in small scale production of various models.
- For facilitating inventory management.
- For updating software after shipment.

Caution For the electrical specifications related to the flash memory rewriting, refer to CHAPTER 20 ELECTRICAL SPECIFICATIONS (TARGET VALUES).

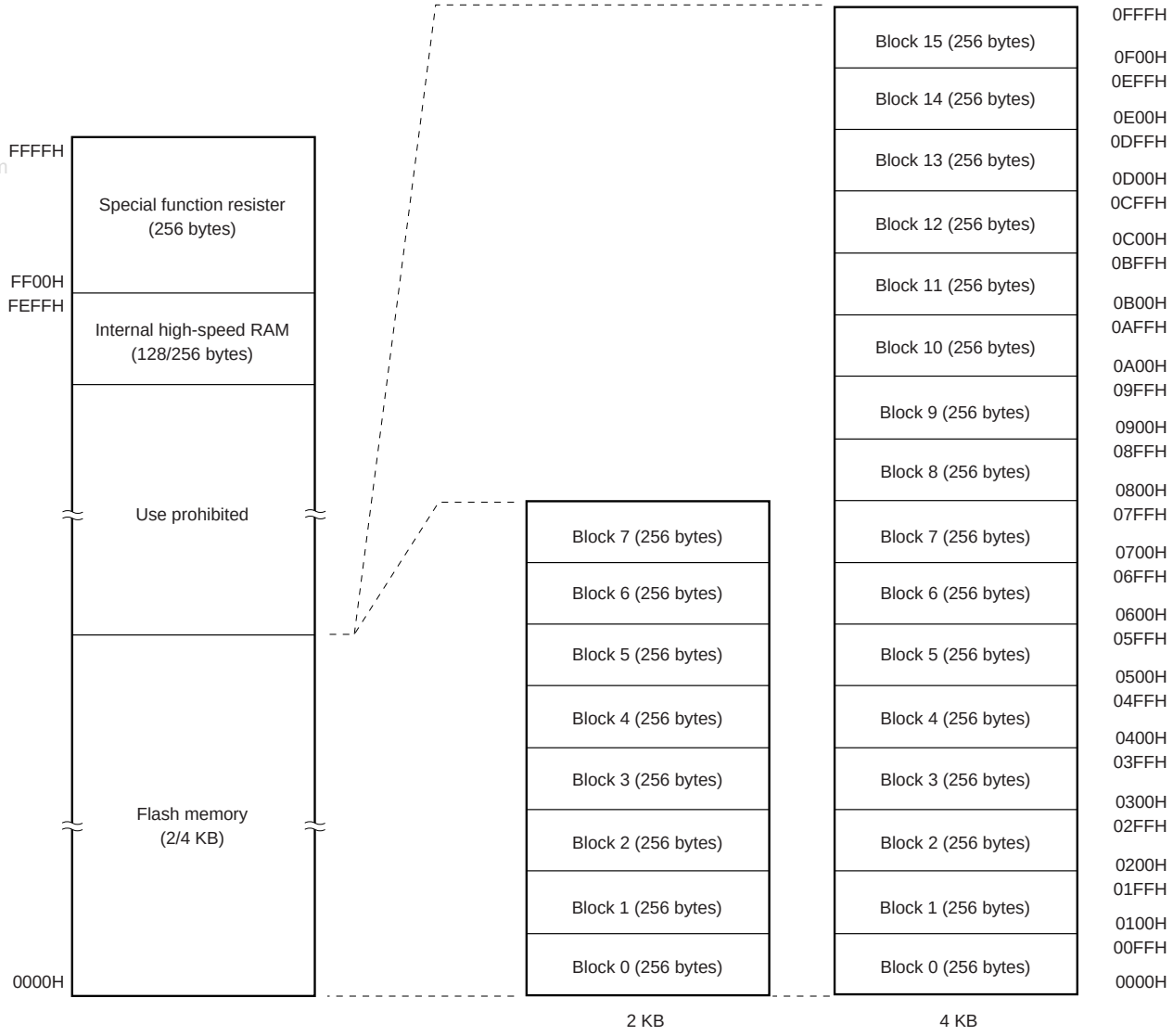
18.1 Features

- Capacity: 4 KB/2 KB
- Write voltage: Erase/write with a single power supply
- Rewriting method
  - Rewriting by communication with dedicated flash programmer via serial interface (on-board/off-board programming)
  - Rewriting flash memory by user program (self programming)
- Flash memory write prohibit function supported (security function)

18.2 Memory Configuration

The 4/2 KB internal flash memory area is divided into 16/8 blocks and can be programmed/erased in block units. All the blocks can also be erased at once.

Figure 18-1. Flash Memory Mapping



18.3 Functional Outline

The internal flash memory of the 78K0S/KA1+ can be rewritten by using the rewrite function of the dedicated flash programmer, regardless of whether the 78K0S/KA1+ has already been mounted on the target system or not (on-board/off-board programming).

In addition, a security function that prohibits rewriting the user program written to the internal flash memory is also supported, so that the program cannot be changed by an unauthorized person.

The rewrite function using the user program (self programming) is ideal for an application where it is assumed that the program is changed after production/shipment of the target system.

Table 18-1. Rewrite Method

Rewrite Method	Functional Outline	Operation Mode
On-board programming	Flash memory can be rewritten after the device is mounted on the target system, by using a dedicated flash programmer.	Flash memory programming mode
Off-board programming	Flash memory can be rewritten before the device is mounted on the target system, by using a dedicated flash programmer and a dedicated program adapter board (FA series).	
Self programming	Flash memory can be rewritten by executing a user program that has been written to the flash memory in advance by means of on-board/off-board programming.	Normal operation mode

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

Table 18-2. Basic Functions

Function	Functional Outline	Support (○: Supported, ×: Not supported)	
		On-Board/Off-Board Programming	Self Programming
Block erasure	The contents of specified memory blocks are erased.	○	○
Chip erasure	The contents of the entire memory area are erased all at once.	○	×
Write	Writing to specified addresses, and a verify check to see if write level is secured are performed.	○	○
Checksum	Data read from the flash memory is compared with data transferred from the flash programmer.	○	×
Security setting	Use of the block erase command, chip erase command, and program command can be prohibited.	○	× (Only values set by on-board/off-board programming can be retained)

The following table lists the security functions. The block erase command prohibit, chip erase command prohibit, and program command prohibit functions are enabled by default after shipment, and security can be set by rewriting via on-board/off-board programming. Each security function can be used in combination with the others at the same time.

Table 18-3. Security Functions

Function	Function Outline	Rewriting Operation When Prohibited (○: Executable, ×: Not Executable)	
		On-Board/Off-Board Programming	Self Programming
Block erase command prohibit	Execution of a block erase command on all blocks is prohibited. Setting of prohibition can be initialized by execution of a chip erase command.	Block erase command: × Chip erase command: ○ Program command: ○	Can always be rewritten regardless of setting of prohibition
Chip erase command prohibit	Execution of block erase and chip erase commands on all the blocks is prohibited. Once prohibition is set, setting of prohibition cannot be initialized because the chip erase command cannot be executed.	Block erase command: × Chip erase command: × Program command: ○	
Program command prohibit	Write and block erase commands on all the blocks are prohibited. Setting of prohibition can be initialized by execution of the chip erase command.	Block erase command: × Chip erase command: ○ Program command: ×	

18.4 Writing with Flash Programmer

Data can be written to the flash memory on-board or off-board, by using a dedicated flash programmer.

(1) On-board programming

The contents of the flash memory can be rewritten after the 78K0S/KA1+ has been mounted on the target system. The connectors that connect the dedicated flash programmer must be mounted on the target system.

(2) Off-board programming

Data can be written to the flash memory with a dedicated program adapter (FA series) before the 78K0S/KA1+ is mounted on the target system.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

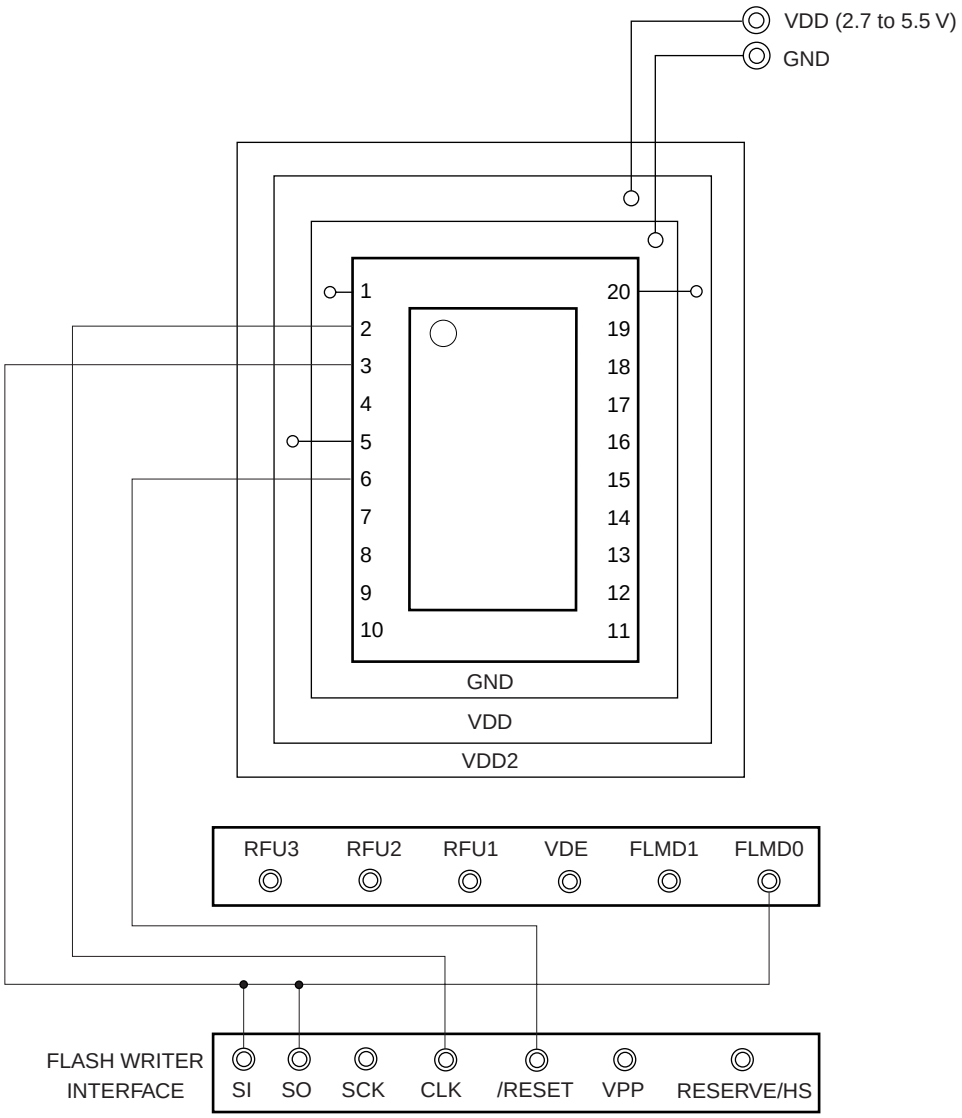
Table 18-4. Wiring Between 78K0S/KA1+ and Dedicated Flash Programmer

Pin Configuration of Dedicated Flash Programmer			Pin Configuration of 78K0S/KA1+	
Pin Name	I/O	Pin Function	Pin Name	Pin No.
CLK[Note]	Output	Clock to 78K0S/KA1+	X1	2
FLMD0[Note]	Output	On-board mode signal		
RxD[Note]	Input	Receive signal	X2	3
TxD[Note]	Output	Receive signal/on-board mode signal		
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	6
V_{DD}	I/O	V_{DD} voltage generation	V_{DD}	5
GND	–	Ground	V_{SS}	1

Note In the 78K0S/KA1+, the CLK and FLMD0 signals are connected to the X1 pin and the RxD and TxD signals to the X2 signal; therefore, these signals need to be directly connected. However, the dedicated flash programmer provides a dedicated adapter for the 78K0S/KA1+, so signals do not need to be internally connected. Therefore, connect the X1 or X2 pin on-board to either one of the corresponding signal lines.

Examples of the recommended connection when using the adapter for flash memory writing are shown below.

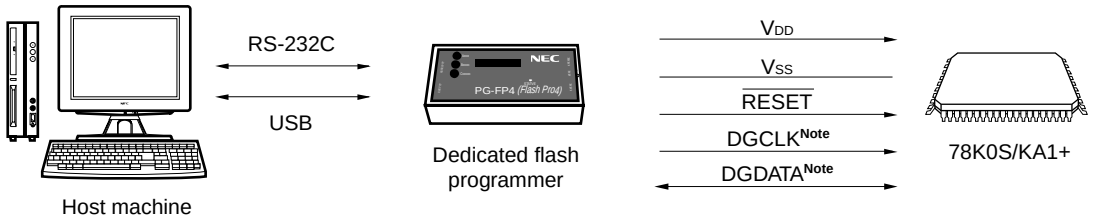
Figure 18-2. Example of Wiring Adapter for Flash Memory Writing



18.5 Programming Environment

The environment required for writing a program to the flash memory is illustrated below.

Figure 18-3. Environment for Writing Program to Flash Memory



Note DGCLK and DGDATA are single-wire bidirectional communication interfaces. They use UART as the communication mode.

A host machine that controls the dedicated flash programmer is necessary.

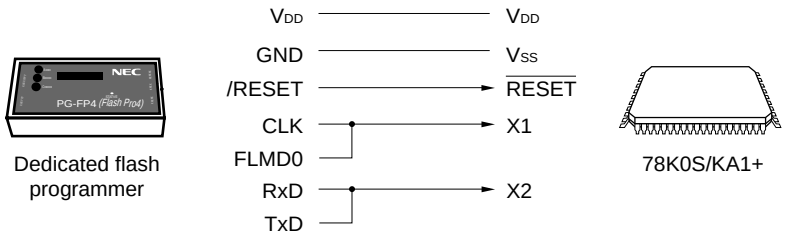
UART is used for manipulation such as writing and erasing when interfacing between the dedicated flash programmer and the 78K0S/KA1+. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

18.6 Communication Mode

Communication between the dedicated flash programmer and the 78K0S/KA1+ is established by serial communication via UART using the X1 or X2 pin of the 78K0S/KA1+.

- Transfer rate: 115200 bps

Figure 18-4. Communication with Dedicated Flash Programmer



18.7 Processing of Pins on Board

To write the flash memory on-board, connectors that connect the dedicated flash programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be processed as described below.

18.7.1 X1 and X2 pins

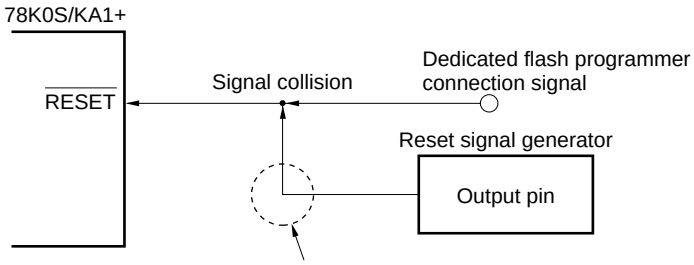
The X1 and X2 pins are used as the serial interface of flash memory programming. Therefore, if the X1 and X2 pins are connected to an external device, a signal conflict occurs. To prevent the conflict of signals, isolate the connection with the external device.

18.7.2 $\overline{\text{RESET}}$ pin

If the reset signal of the dedicated flash programmer is connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on the board, signal collision takes place. To prevent this collision, isolate the connection with the reset signal generator.

If the reset signal is input from the user system while the flash memory programming mode is set, the flash memory will not be correctly programmed. Do not input any signal other than the reset signal of the dedicated flash programmer.

Figure 18-5. Signal Collision ($\overline{\text{RESET}}$ Pin)



In the flash memory programming mode, the signal output by the reset signal generator collides with the signal output by the dedicated flash programmer. Therefore, isolate the signal of the reset signal generator.

18.7.3 Port pins

When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to V_{DD} or V_{SS} via a resistor.

18.7.4 Power supply

Connect the V_{DD} pin to V_{DD} of the flash programmer, and the V_{SS} pin to V_{SS} of the flash programmer.

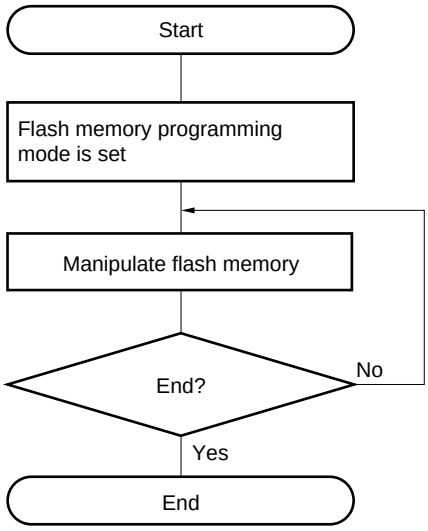
Supply AV_{REF} with the same power supply as that in the normal operation mode.

18.8 Programming Method

18.8.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 18-6. Flash Memory Manipulation Procedure



18.8.2 Flash memory programming mode

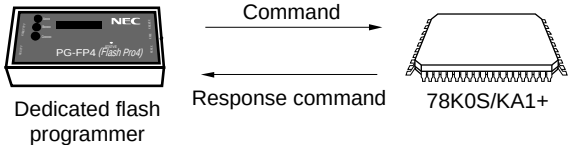
To rewrite the contents of the flash memory by using the dedicated flash programmer, set the 78K0S/KA1+ in the flash memory programming mode. When the 78K0S/KA1+ is connected to the flash programmer and a communication command is transmitted to the microcontroller, the microcontroller is set in the flash memory programming mode.

Change the mode by using a jumper when writing the flash memory on-board.

18.8.3 Communication commands

The 78K0S/KA1+ communicates with the dedicated flash programmer by using commands. The signals sent from the flash programmer to the 78K0S/KA1+ are called commands, and the commands sent from the 78K0S/KA1+ to the dedicated flash programmer are called response commands.

Figure 18-7. Communication Commands



The flash memory control commands of the 78K0S/KA1+ are listed in the table below. All these commands are issued from the programmer and the 78K0S/KA1+ perform processing corresponding to the respective commands.

Table 18-5. Flash Memory Control Commands

Classification	Command Name	Function
Erase	Batch erase command	Erases the contents of the entire memory
	Block erase command	Erases the contents of the memory of the specified block
Write	Write command	Writes to the specified address range and executes a verify check of the contents.
Checksum	Checksum command	Reads the checksum of the specified address range and compares with the written data.
Security	Security set command	Prohibits chip erase command, block erase command, and write command to prevent operation by third parties.

The 78K0S/KA1+ returns a response command for the command issued by the dedicated flash programmer. The response commands sent from the 78K0S/KA1+ are listed below.

Table 18-6. Response Commands

Command Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

18.9 Flash Memory Programming by Self Writing

The 78K0S/KA1+ supports a self programming function that can be used to rewrite the flash memory via a user program, making it possible to upgrade programs in the field.

Cautions 1. Self programming processing must have been implemented before performing self writing.
2. Temporarily store the data to be rewritten in the internal high-speed RAM.
3. Switch the CPU clock to high-speed Ring-OSC when executing self programming.
4. Interrupt processing cannot be used while self programming is in progress.

CHAPTER 19 INSTRUCTION SET OVERVIEW

This chapter lists the instruction set of the 78K0S/KA1+. For details of the operation and machine language (instruction code) of each instruction, refer to **78K/0S Series Instructions User's Manual (U11047E)**.

19.1 Operation

19.1.1 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 details). When there are two or more description methods, select one of them. Uppercase letters and the 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 parentheses in the table below, R0, R1, R2, etc.) can be used for description.

Table 19-1. 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 For symbols of special function registers, see **Table 4-3 Special Function Registers**.

19.1.2 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 parentheses
×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)

19.1.3 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

19.2 Operation List

Mnemonic	Operand	Bytes	Clocks	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 the processor clock control register (PCC).

Mnemonic	Operand	Bytes	Clocks	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 the processor clock control register (PCC).

Mnemonic	Operand	Bytes	Clocks	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 the processor clock control register (PCC).

Mnemonic	Operand	Bytes	Clocks	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 the processor clock control register (PCC).

Mnemonic	Operand	Bytes	Clocks	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 the processor clock control register (PCC).

19.3 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]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV[Note] XCH[Note] ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											
[HL + byte]		MOV											

Note Except r = A.

(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 20 ELECTRICAL SPECIFICATIONS (TARGET VALUES)

These specifications are only target values, and may not be satisfied by mass-produced products.

Absolute Maximum Ratings (T_A = 25°C)

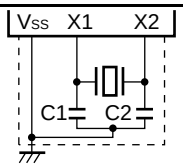
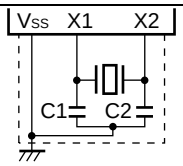
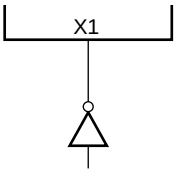
Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.3 to +6.5	V
	V_{SS}		–0.3 to +0.3	V
	AV_{REF}		–0.3 to V_{DD} + 0.3[Note]	V
Input voltage	V_{I1}	P30, P31, P34, P40 to P45, P121 to P123	–0.3 to V_{DD} + 0.3[Note]	V
	V_{I2}	P20 to 23	– 0.3 to AV_{REF} + 0.3[Note] and –0.3 to V_{DD} + 0.3[Note]	V
Output voltage	V_O		–0.3 to V_{DD} + 0.3[Note]	V
Analog input voltage	V_{AN}		– 0.3 to AV_{REF} + 0.3[Note] and –0.3 to V_{DD} + 0.3[Note]	V
Output current, high	I_{OH}	Per pin	–10	mA
		Total of pins other than P20 to P23	–30	mA
Output current, low	I_{OL}	Per pin	20	mA
		Total of all pins	35	mA
Operating ambient temperature	T_A	In normal operation mode	–40 to +85	°C
		During flash memory programming	–10 to +70	°C
Storage temperature	T_{stg}		–40 to +125	°C

Note Must be 6.5 V or lower

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

X1 Oscillator Characteristics (T_A = –40 to +85°C, V_{DD} = 2.0 to 5.5 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		Oscillation frequency (f_X)[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V	0.5		10.0	MHz
			3.0 V ≤ V_{DD} < 4.0 V	0.5		6.0	
			2.7 V ≤ V_{DD} < 3.0 V	0.5		5.0	
			2.0 V ≤ V_{DD} < 2.7 V	0.5		0.5	
Crystal resonator		Oscillation frequency (f_X)[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V	0.5		10.0	MHz
			3.0 V ≤ V_{DD} < 4.0 V	0.5		6.0	
			2.7 V ≤ V_{DD} < 3.0 V	0.5		5.0	
			2.0 V ≤ V_{DD} < 2.7 V	0.5		0.5	
External clock		X1 input frequency (f_X)[Note]	4.5 V ≤ V_{DD} ≤ 5.5 V	0.5		10.0	MHz
			4.0 V ≤ V_{DD} < 4.5 V	0.5		6.0	
			2.7 V ≤ V_{DD} < 4.0 V	0.5		5.0	
			2.0 V ≤ V_{DD} < 2.7 V	0.5		0.5	
		X1 input high-/low-level width (t_{XH}, t_{XL})	4.5 V ≤ V_{DD} ≤ 5.5 V	0.045		1	μs
			4.0 V ≤ V_{DD} < 4.5 V	0.075		1	
			2.7 V ≤ V_{DD} < 4.0 V	0.085		1	
			2.0 V ≤ V_{DD} < 2.7 V	1		1	

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Caution When using the X1 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_{SS}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

Remark For the resonator selection and oscillator constant, users are required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

High-Speed Ring-OSC Oscillator Characteristics (T_A = –40 to +85°C, V_{DD} = 2.0 to 5.5 V)

Resonator	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
On-chip high-speed Ring-OSC	Oscillation frequency (f_X)	2.7 V ≤ V_{DD} ≤ 5.5 V	7.60	8.00	8.40	MHz
		2.0 V ≤ V_{DD} < 2.7 V		T.B.D		MHz

Low-Speed Ring-OSC Oscillator Characteristics (T_A = –40 to +85°C, V_{DD} = 2.0 to 5.5 V)

Resonator	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
On-chip low-speed Ring-OSC	Oscillation frequency (f_{RL})	2.7 V ≤ V_{DD} ≤ 5.5 V	120	240	480	kHz
		2.0 V ≤ V_{DD} < 2.7 V		T.B.D		kHz

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 2.0 to 5.5 V) (1/2)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Output current, high	I_{OH1}	Pins other than P20 to P23	Per pin	2.0 V ≤ V_{DD} ≤ 5.5 V			–5	mA
			Total	4.0 V ≤ V_{DD} ≤ 5.5 V			–25	mA
				2.0 V ≤ V_{DD} < 4.0 V			–15	mA
	I_{OH2}	P20 to P23	Per pin	2.0 V ≤ V_{DD} ≤ 5.5 V			–15	mA
			Total	2.0 V ≤ V_{DD} ≤ 5.5 V			–5	mA
Output current, low	I_{OL}	Per pin		2.0 V ≤ V_{DD} ≤ 5.5 V			10	mA
		Total of all pins		4.0 V ≤ V_{DD} ≤ 5.5 V			30	mA
				2.0 V ≤ V_{DD} < 4.0 V			15	mA
Input voltage, high	V_{IH1}	P30, P31, P34, P40 to P45, P123			0.8V_{DD}		V_{DD}	V
	V_{IH2}	P20 to P23			0.7AV_{REF}		AV_{REF}	V
	V_{IH3}	P121, P122			0.7V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P30, P31, P34, P40 to P45, P123			0		0.2V_{DD}	V
	V_{IL2}	P20 to P23			0		0.3AV_{REF}	V
	V_{IL3}	P121, P122			0		0.3V_{DD}	V
Output voltage, high	V_{OH1}	Total of pins other than P20 to P23 I_{OH} = –15 mA		4.0 V ≤ V_{DD} ≤ 5.5 V I_{OH} = –5 mA	T.B.D			V
		I_{OH} = –100 μA		2.0 V ≤ V_{DD} < 4.0 V	V_{DD} – 0.5			V
	V_{OH2}	Total of pins P20 to P23 I_{OH} = –10 mA		4.5 V ≤ AV_{REF} ≤ 5.5 V I_{OH} = –5 mA	T.B.D			V
				4.0 V ≤ AV_{REF} < 4.5 V I_{OH} = –5 mA	T.B.D			V
				2.85 V ≤ AV_{REF} < 4.0 V I_{OH} = –5 mA	T.B.D			V
				2.7 V ≤ AV_{REF} < 2.85 V I_{OH} = –5 mA	T.B.D			V
Output voltage, low	V_{OL}	Total of pins I_{OL} = 30 mA		I_{OL} = 10 mA			T.B.D	V
Input leakage current, high	I_{LIH1}	V_I = V_{DD}	Pins other than X1				3	μA
	I_{LIH2}		X1				T.B.D	μA
Input leakage current, low	I_{LIL1}	V_I = 0 V	Pins other than X1				–3	μA
	I_{LIL2}		X1				T.B.D	μA
Output leakage current, high	I_{LOH}	V_O = V_{DD}	Pins other than X2				3	μA
			X2				T.B.D	μA
Output leakage current, low	I_{LOL}	V_O = 0 V	Pins other than X2				–3	μA
			X2				T.B.D	μA

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 2.0 to 5.5 V) (2/2)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Pull-up resistance value	R	V_I = 0 V			10	30	100	kΩ
Supply current[Note 1]	I_{DD1}[Note 2]	Crystal/ceramic oscillation, external clock input oscillation operating mode[Note 5]	f_X = 10 MHz V_{DD} = 5.0 V ±10%[Note 3]	When A/D converter is stopped		6.1	12.2	mA
				When A/D converter is operating[Note 7]		7.6	15.2	
			f_X = 6 MHz V_{DD} = 3.0 V ±10%[Note 3]	When A/D converter is stopped		T.B.D	T.B.D	mA
				When A/D converter is operating[Note 7]		T.B.D	T.B.D	
			f_X = 5 MHz V_{DD} = 2.7 V ±10%[Note 4]	When A/D converter is stopped		T.B.D	T.B.D	mA
				When A/D converter is operating[Note 7]		T.B.D	T.B.D	
	I_{DD2}	Crystal/ceramic oscillation, external clock input HALT mode[Note 5]	f_X = 10 MHz V_{DD} = 5.0 V ±10%[Note 3]	When peripheral functions are stopped		T.B.D	T.B.D	mA
				When peripheral functions are operating		T.B.D	T.B.D	
			f_X = 6 MHz V_{DD} = 3.0 V ±10%[Note 3]	When peripheral functions are stopped		T.B.D	T.B.D	mA
				When peripheral functions are operating		T.B.D	T.B.D	
			f_X = 5 MHz V_{DD} = 2.7 V ±10%[Note 4]	When peripheral functions are stopped		T.B.D	T.B.D	mA
				When peripheral functions are operating		T.B.D	T.B.D	
	I_{DD3}	High-speed Ring-OSC operation mode[Note 6]	f_X = 8 MHz V_{DD} = 5.0 V ±10%[Note 3]	When A/D converter is stopped		5.5	11.0	mA
				When A/D converter is operating[Note 7]		7	14.0	
			f_X = 4 MHz V_{DD} = 2.7 V ±10%[Note 3]	When A/D converter is stopped		T.B.D	T.B.D	mA
				When A/D converter is operating[Note 7]		T.B.D	T.B.D	
	I_{DD4}	High-speed Ring-OSC HALT mode[Note 6]	f_X = 8 MHz V_{DD} = 5.0 V ±10%[Note 3]	When peripheral functions are stopped		T.B.D	T.B.D	mA
				When peripheral functions are operating		T.B.D	T.B.D	
			f_X = 4 MHz V_{DD} = 2.7 V ±10%[Note 3]	When peripheral functions are stopped		T.B.D	T.B.D	mA
				When peripheral functions are operating		T.B.D	T.B.D	
	I_{DD5}	STOP mode	V_{DD} = 5.0 V ±10%	When low-speed Ring-OSC is stopped		3.5	35.5	μA
				When low-speed Ring-OSC is operating		17.5	63.5	
			V_{DD} = 3.0 V ±10%	When low-speed Ring-OSC is stopped		3.5	15.5	μA
				When low-speed Ring-OSC is operating		11	30.5	
			V_{DD} = 2.7 V ±10%	When low-speed Ring-OSC is stopped		T.B.D	T.B.D	μA
				When low-speed Ring-OSC is operating		T.B.D	T.B.D	

Notes 1. Total current flowing through the internal power supply (V_{DD}). Peripheral operation current is included (however, the current that flows through the pull-up resistors of ports is not included).

2. I_{DD1} includes peripheral operation current.

3. When the processor clock control register (PCC) is set to 00H.

4. When the processor clock control register (PCC) is set to 02H.

5. When crystal/ceramic oscillation clock, external clock input is selected as the system clock source using the option byte.

6. When the high-speed Ring-OSC is selected as the system clock source using the option byte.

7. The current that flows through the AV_{REF} pin is included.

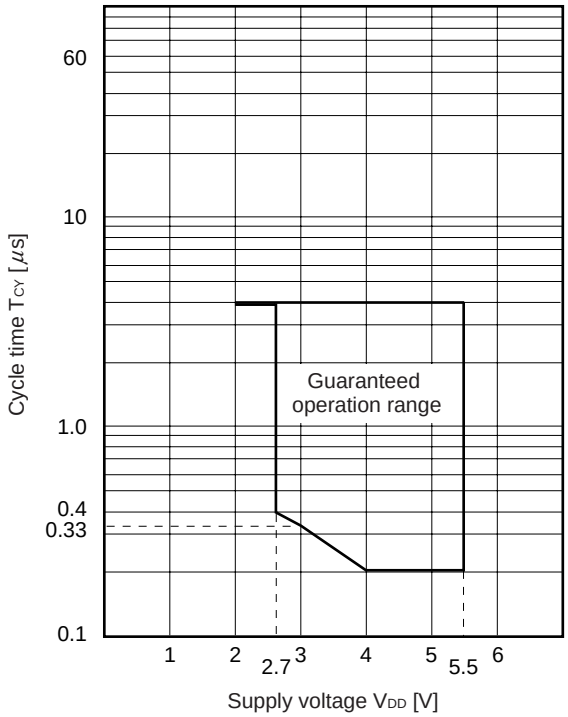
AC Characteristics

(1) Basic operation (T_A = –40 to +85°C, V_{DD} = 2.0 to 5.5 V)

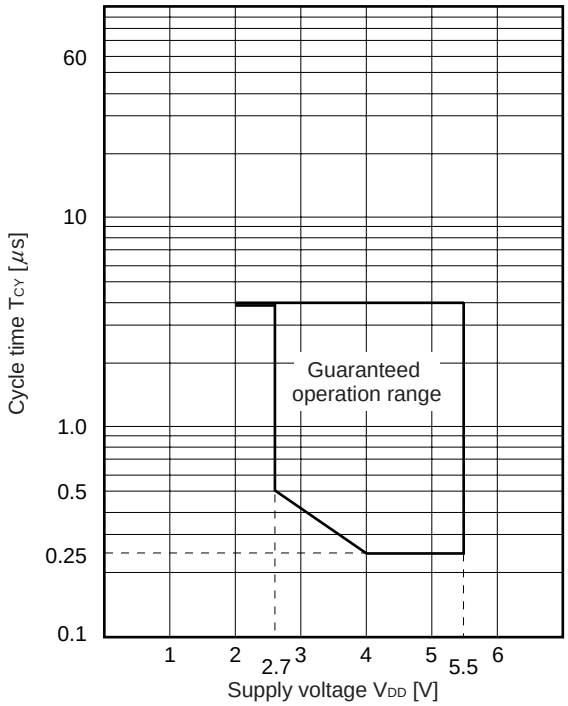
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Cycle time (minimum instruction execution time)	T_{CY}	Crystal/ceramic oscillation clock, external clock input	4.0 V ≤ V_{DD} ≤ 5.5 V	0.2		4	μs
			3.0 V ≤ V_{DD} < 4.0 V	0.33		4	μs
			2.7 V ≤ V_{DD} < 3.0 V	0.4		4	μs
			2.0 V ≤ V_{DD} < 2.7 V	4		4	μs
		High-speed Ring-OSC clock	4.0 V ≤ V_{DD} ≤ 5.5 V	0.25		4	μs
			2.7 V ≤ V_{DD} < 4.0 V	0.5		4	μs
			2.0 V ≤ V_{DD} < 2.7 V	4		4	μs
TI000 input high-level width, low-level width	t_{TIH}, t_{TIL}	4.0 V ≤ V_{DD} ≤ 5.5 V		2/fsam+ 0.1[Note]			μs
		2.0 V ≤ V_{DD} < 4.0 V		2/fsam+ 0.2[Note]			μs
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}			1			μs
$\overline{\text{RESET}}$ input low-level width	t_{RSL}			1			μs

Note Selection of fsam = f_{XP}, f_{XP}/4, or f_{XP}/256 is possible using bits 0 and 1 (PRM000, PRM001) of prescaler mode register 00 (PRM00). Note that when selecting the TI000 valid edge as the count clock, fsam = f_{XP}.

T_{CY} vs. V_{DD} (Crystal/Ceramic Oscillation Clock, External Clock Input)



T_{CY} vs. V_{DD} (High-speed Ring-OSC Clock)

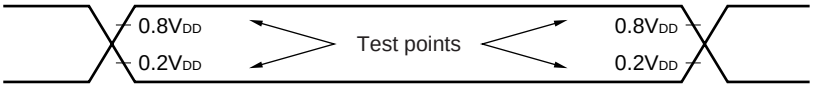


(2) Serial interface (T_A = –40 to +85°C, V_{DD} = 2.0 to 5.5 V)

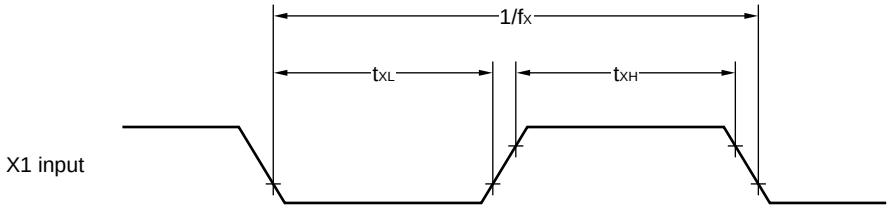
UART mode (UART6, dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					312.5	kbps

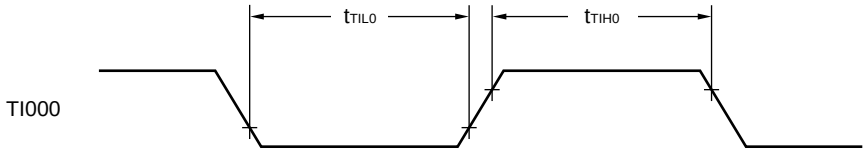
AC Timing Test Points (Excluding X1 Input)



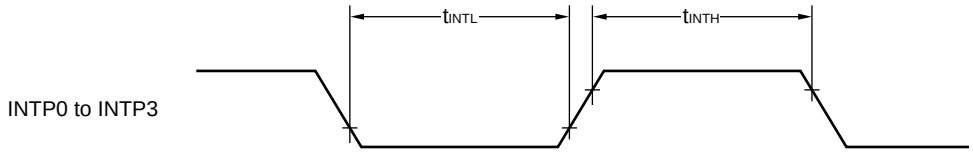
Clock Timing



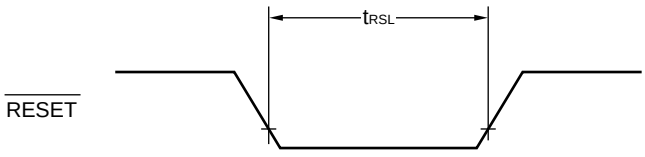
TI000 Timing



Interrupt Input Timing



$\overline{\text{RESET}}$ Input Timing



A/D Converter Characteristics (T_A = –40 to +85°C, 2.7 V ≤ AV_{REF} ≤ V_{DD} ≤ 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Notes 1, 2]	AINL	4.0 V ≤ AV_{REF} ≤ 4.5 V		±0.2	±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V		±0.3	±0.6	%FSR
Conversion time	t_{CONV}	4.5 V ≤ AV_{REF} ≤ 5.5 V	3.0		100	μs
		4.0 V ≤ AV_{REF} < 4.5 V	4.8		100	μs
		2.85 V ≤ AV_{REF} < 4.0 V	6.0		100	μs
		2.7 V ≤ AV_{REF} < 2.85 V	14.0		100	μs
Zero-scale error[Notes 1, 2]	Ezs	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
Full-scale error[Notes 1, 2]	Efs	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
Integral non-linearity error[Note 1]	ILE	4.0 V ≤ AV_{REF} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±4.5	LSB
Differential non-linearity error[Note 1]	DLE	4.0 V ≤ AV_{REF} ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±2.0	LSB
Analog input voltage	V_{IAN}		V_{SS}[Note 3]		AV_{REF}	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

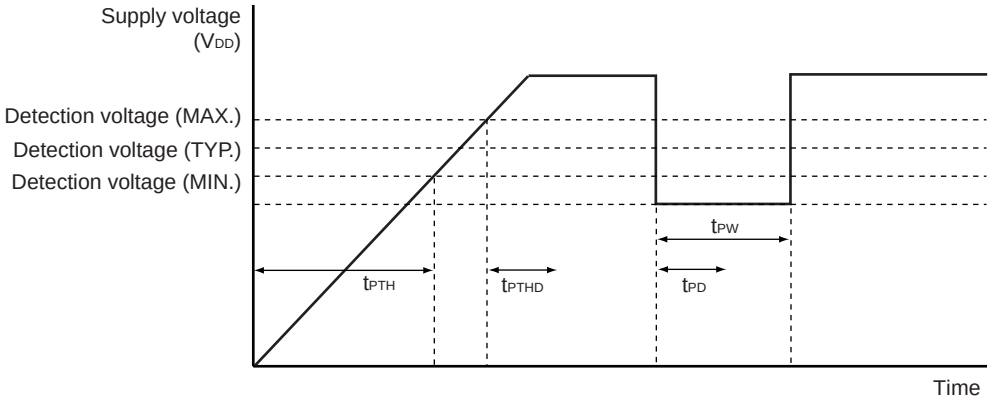
3. V_{SS} and AV_{SS} are internally connected in the 78K0S/KA1+. Be sure to stabilize V_{SS} by connecting it to a stable GND (= 0 V). If the status of the output port is varied during A/D conversion, the conversion characteristics may be degraded.

POC Circuit Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC}		2.0	2.1	2.2	V
Power supply rise time	t_{PTH}	V_{DD}: 0 V → 2.0 V	1.5			μs
Response delay time 1[Note]	t_{PTHD}	When power supply rises, after reaching detection voltage (MAX.)			3.0	ms
Response delay time 2[Note]	t_{PD}	When power supply falls			1.0	ms
Minimum pulse width	t_{PW}		0.2			ms

Note Time required from voltage detection to reset release.

POC Circuit Timing



LVI Circuit Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{LVI0}		4.1	4.3	4.5	V
	V_{LVI1}		3.9	4.1	4.3	V
	V_{LVI2}		3.7	3.9	4.1	V
	V_{LVI3}		3.5	3.7	3.9	V
	V_{LVI4}		3.3	3.5	3.7	V
	V_{LVI5}		3.15	3.3	3.45	V
	V_{LVI6}		2.95	3.1	3.25	V
	V_{LVI7}		2.7	2.85	3.0	V
	V_{LVI8}		2.5	2.6	2.7	V
	V_{LVI9}		2.25	2.35	2.45	V
Response time[Note 1]	t_{LD}			0.2	2.0	ms
Minimum pulse width	t_{LW}		0.2			ms
Operation stabilization wait time[Note 2]	t_{WAIT}			0.1	0.2	ms

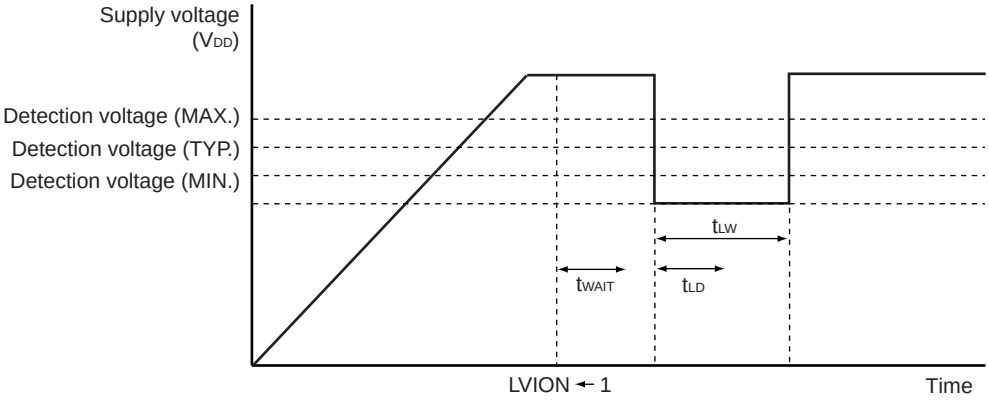
Notes 1. Time required from voltage detection to interrupt output or $\overline{\text{RESET}}$ output.

2. Time required from setting LVION to 1 to operation stabilization.

Remarks 1. $V_{LVI0} > V_{LVI1} > V_{LVI2} > V_{LVI3} > V_{LVI4} > V_{LVI5} > V_{LVI6} > V_{LVI7} > V_{LVI8} > V_{LVI9}$

2. $V_{POC} < V_{LVIm}$ (m = 0 to 9)

LVI Circuit Timing



Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		2.0		5.5	V
Release signal set time	t_{SREL}		0			μs

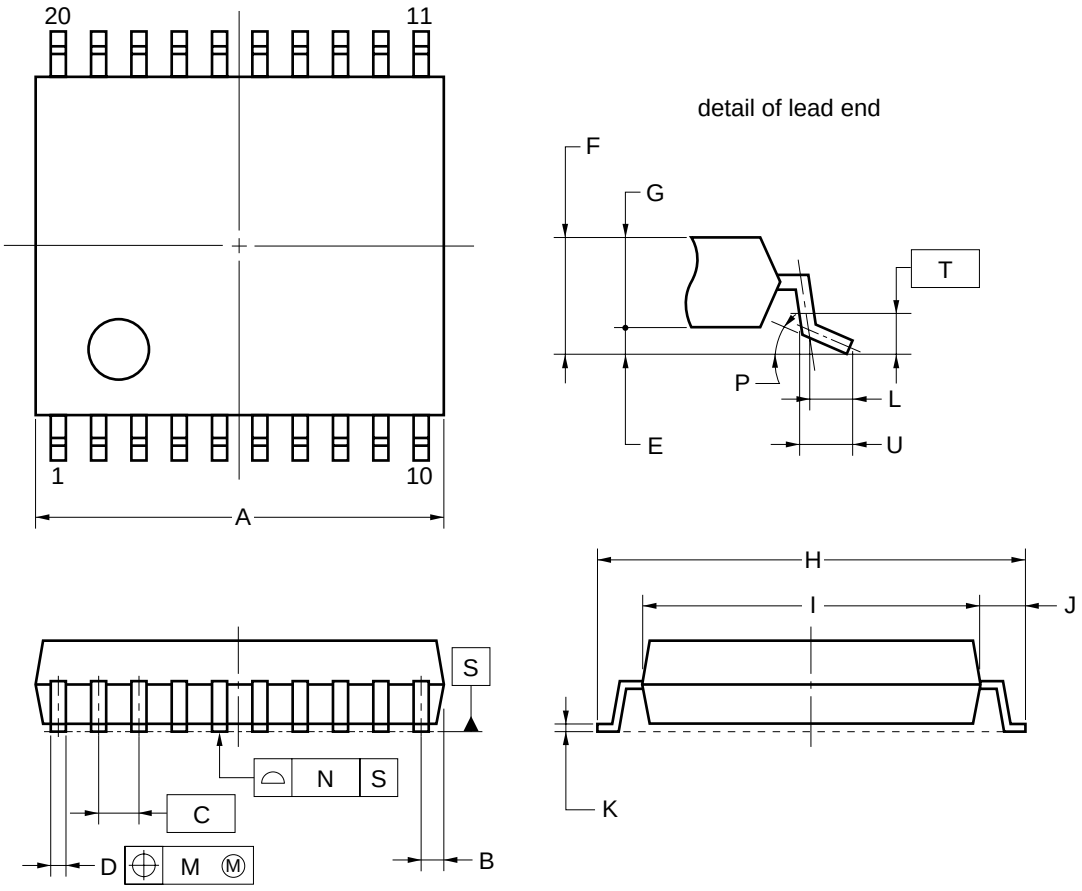
Flash Memory Programming Characteristics (T_A = –10°C to +70°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Erase time[Note 1]	Chip unit	t_{eraca}			T.B.D	T.B.D	ms
	Sector unit	t_{erasa}			T.B.D	T.B.D	ms
Write time		t_{wrwa}			T.B.D	T.B.D	μs
Number of rewrites per chip		C_{erwr}	1 erase + 1 write after erase = 1 rewrite[Note 2]	100			Times

Notes 1. The erase verify time is not included.

2. When a product is first written after shipment, “erase → write” and “write only” are both taken as one rewrite.

20-PIN PLASTIC SSOP (7.62 mm (300))



NOTE

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	6.65±0.15
B	0.475 MAX.
C	0.65 (T.P.)
D	$0.24^{+0.08}_{-0.07}$
E	0.1±0.05
F	1.3±0.1
G	1.2
H	8.1±0.2
I	6.1±0.2
J	1.0±0.2
K	0.17±0.03
L	0.5
M	0.13
N	0.10
P	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
T	0.25
U	0.6±0.15

S20MC-65-5A4-2

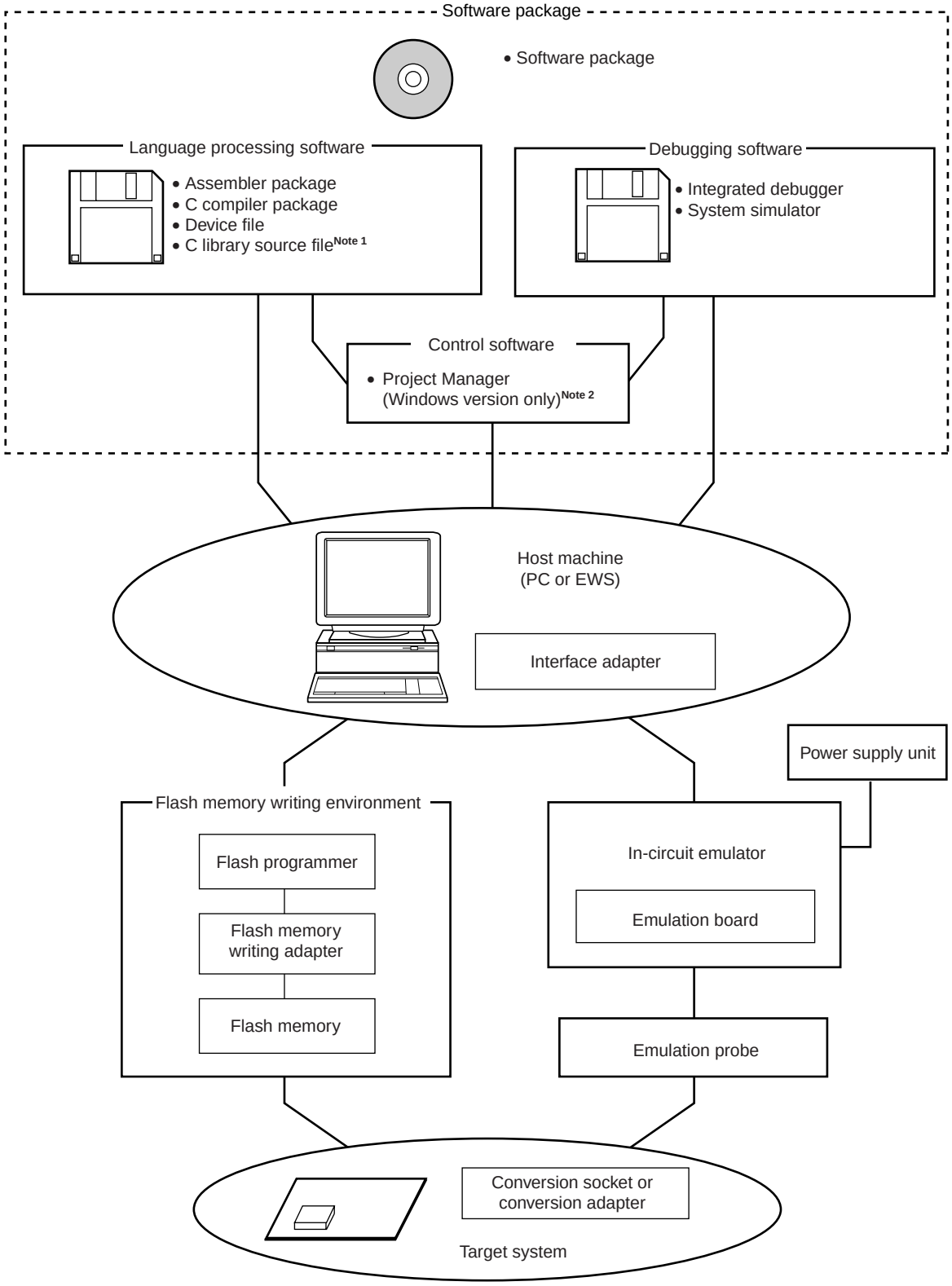
APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for development of systems using the 78K0S/KA1+. Figure A-1 shows development tools.

- Compatibility with PC98-NX series
  Unless stated otherwise, products which are supported by IBM PC/AT™ and compatibles can also be used with the PC98-NX series. When using the PC98-NX series, therefore, refer to the explanations for IBM PC/AT and compatibles.

- Windows
  Unless stated otherwise, “Windows” refers to the following operating systems.
  - Windows 3.1
  - Windows 95, 98, 2000, XP
  - Windows NT™ Ver. 4.0

Figure A-1. Development Tools



Notes 1. The C library source file is not included in the software package.

2. The Project Manager is included in the assembler package.
The Project Manager is used only in the Windows environment.

A.1 Software Package

SP78K0S Software package	This is a package that bundles the software tools required for development of the 78K/0S Series. The following tools are included. RA78K0S, CC78K0S, ID78K0S-NS, SM78K0S/Kx1 (provisional name), and device files
	Part number: μS××××SP78K0S

Remark ×××× in the part number differs depending on the operating system to be used.

μS××××SP78K0S

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT and compatibles	Japanese Windows	CD-ROM
BB17		English Windows	

A.2 Language Processing Software

RA78K0S Assembler package	Program that converts program written in mnemonic into object code that can be executed by microcontroller. In addition, automatic functions to generate symbol table and optimize branch instructions are also provided. Used in combination with device file (DF78K0S/Kx1 (provisional name)) (sold separately). **<Caution when used in PC environment>** The assembler package is a DOS-based application but may be used under the Windows environment by using Project Manager of Windows (included in the assembler package).
	Part number: μS××××RA78K0S
CC78K0S C library package	Program that converts program written in C language into object codes that can be executed by microcontroller. Used in combination with assembler package (RA78K0S) and device file (DF78K0S/Kx1) (both sold separately). **<Caution when used in PC environment>** The C compiler package is a DOS-based application but may be used under the Windows environment by using Project Manager of Windows (included in the assembler package).
	Part number: μS××××CC78K0S
DF78K0S/Kx1 (provisional name)[Notes 1, 2] Device file	File containing the information inherent to the device. Used in combination with other tools (RA78K0S, CC78K0S, ID78K0S-NS, or SM78K0S/Kx1 (provisional name)) (all sold separately).
	Part number: T.B.D.
CC78K0S-L[Note 3] C library source file	Source file of functions constituting object library included in C compiler package. Necessary for changing object library included in C compiler package according to customer's specifications. Since this is the source file, its working environment does not depend on any particular operating system.
	Part number: μS××××CC78K0S-L

Notes 1. DF78K0S/Kx1 (provisional name) is a common file that can be used with RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S/Kx1 (provisional name).

2. Under development

3. CC78K0S-L is not included in the software package (SP78K0S).

Remark ××××in the part number differs depending on the host machine and operating system to be used.

μS××××RA78K0S
μS××××CC78K0S

××××	Host Machine	OS	Supply Media
AB13	PC-9800 series, IBM PC/AT and compatibles	Japanese Windows	3.5” 2HD FD
BB13		English Windows	
AB17		Japanese Windows	CD-ROM
BB17		English Windows	
3P17	HP9000 series 700™	HP-UX™ (Rel.10.10)	
3K17	SPARCstation™	SunOS™ (Rel.4.1.4), Solaris™ (Rel.2.5.1)	

μS××××CC78K0S-L

××××	Host Machine	OS	Supply Media
AB13	PC-9800 series, IBM PC/AT and compatibles	Japanese Windows	3.5” 2HD FD
BB13		English Windows	
3P16	HP9000 series 700	HP-UX (Rel.10.10)	DAT
3K13	SPARCstation	SunOS (Rel.4.1.4), Solaris (Rel.2.5.1)	3.5” 2HD FD
3K15			1/4” CGMT

A.3 Control Software

Project Manager	This is control software designed so that the user program can be efficiently developed in the Windows environment. With this software, a series of user program development operations, including starting the editor, build, and starting the debugger, can be executed on the Project Manager. **<Caution>** The Project Manager is included in the assembler package (RA78K0S). It can be used only in the Windows environment.

A.4 Flash Memory Writing Tools

Flashpro IV (FL-PR4, PG-FP4) Flash programmer	Flash programmer dedicated to the microcontrollers incorporating a flash memory
FA-20MC Flash memory writing adapter	Flash memory writing adapter. Used in connection with Flashpro IV. Designed for use with a 20-pin plastic SSOP (MC-5A4 type).

Remark FL-PR4 and FA-20MC are products of Naito Densei Machida Mfg. Co., Ltd.
For further information, contact: Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191)

A.5 Debugging Tools (Hardware)

IE-78K0S-NS In-circuit emulator	In-circuit emulator for debugging hardware and software of application system using 78K/0S Series. Supports integrated debugger (ID78K0S-NS). Used in combination with AC adapter, emulation probe, and interface adapter for connecting the host machine.
IE-78K0S-NS-A In-circuit emulator	This in-circuit emulator has a coverage function in addition to the functions of the IE-78K0S-NS, and enhanced debugging functions such as an enhanced tracer function and timer function.
IE-70000-MC-PS-B AC adapter	Adapter for supplying power from 100 to 240 VAC outlet.
IE-70000-98-IF-C Interface adapter	Adapter required when using a PC-9800 series (except notebook type) as the host machine (C bus supported).
IE-70000-CD-IF-A PC card interface	PC card and interface cable required when using a notebook type PC as the host machine (PCMCIA socket supported).
IE-70000-PC-IF-C Interface adapter	Adapter required when using IBM PC/AT and compatibles as the host machine (ISA bus supported).
IE-70000-PCI-IF-A Interface adapter	Adapter required when using a personal computer incorporating the PCI bus is used as the host machine.
IE-789244-NS-EM1 (provisional name) [Note] Emulation board	Emulation board for emulating the peripheral hardware inherent to the device. Used in combination with in-circuit emulator.
NP-20GS Emulation probe	Probe for connecting in-circuit emulator and target system. For 20-pin plastic SSOP (MC-5A4 type)
EV-9500GS-20 Conversion adapter	Conversion adapter for connecting target system board for mounting 20-pin plastic SSOP (MC-5A4 type) and NP-20GS.
NP-30MC Emulation probe	This probe is used to connect the in-circuit emulator to the target system and is designed for use with a 30-pin plastic SSOP (MC-5A4 type).
NSPACK20BK YSPACK30BK Conversion socket	This conversion socket connects the NP-30MC to a target system board designed to mount a 20-pin plastic SSOP (MC-5A4 type). • NSPACK20BK: Socket for connecting target • YSPACK30BK: Socket for connecting emulator

Note Under development

Remarks 1. NP-20GS and NP-30MC are products of Naito Densei Machida Mfg. Co., Ltd.
For further information, contact: Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191)

2. NSPACK20BK and YSPACK30BK are products of TOKYO ELETECH CORPORATION.
For further information, contact Daimaru Kogyo Co., Ltd.
Tokyo Electronics Department (TEL: +81-3-3820-7112)
Osaka Electronics Department (TEL: +81-6-6244-6672)

A.6 Debugging Tools (Software)

ID78K0S-NS Integrated debugger	This debugger supports the in-circuit emulators for the 78K/0S Series, IE-78K0S-NS and IE-78K0S-NS-A. ID78K0S-NS is Windows-based software. This debugger has enhanced debugging functions supporting C language. By using its window integration function that associates the source program, disassemble display, and memory display with trace results, the trace results can be displayed corresponding to the source program. It is used with a device file (DF78K0S/Kx1 (provisional name)) (sold separately).
	Part number: μS××××ID78K0S-NS
SM78K0S/Kx1 (provisional name)[Notes 1, 2] System simulator	This is a system simulator for the 78K/0S series. SM78K0S/Kx1 (provisional name) is Windows-based software. This simulator can execute C-source-level or assembler-level debugging while simulating the operations of the target system on the host machine. By using SM78K0S/Kx1 (provisional name), the logic and performance of the application can be verified independently of hardware development. Therefore, the development efficiency can be enhanced and the software quality can be improved. This simulator is used with a device file (DF78K0S/Kx1 (provisional name)) (sold separately).
	Part number: T.B.D.
DF78K0S/Kx1 (provisional name)[Notes 1, 2] Device file	This is a file that has device-specific information. It is used with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S/Kx1 (provisional name) (all sold separately).
	Part number: T.B.D.

Notes 1. DF78K0S/Kx1 (provisional name) is a common file that can be used with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S/Kx1 (provisional name).

2. Under development

Remark ×××× in the part number differs depending on the operating system to be used and the supply medium.

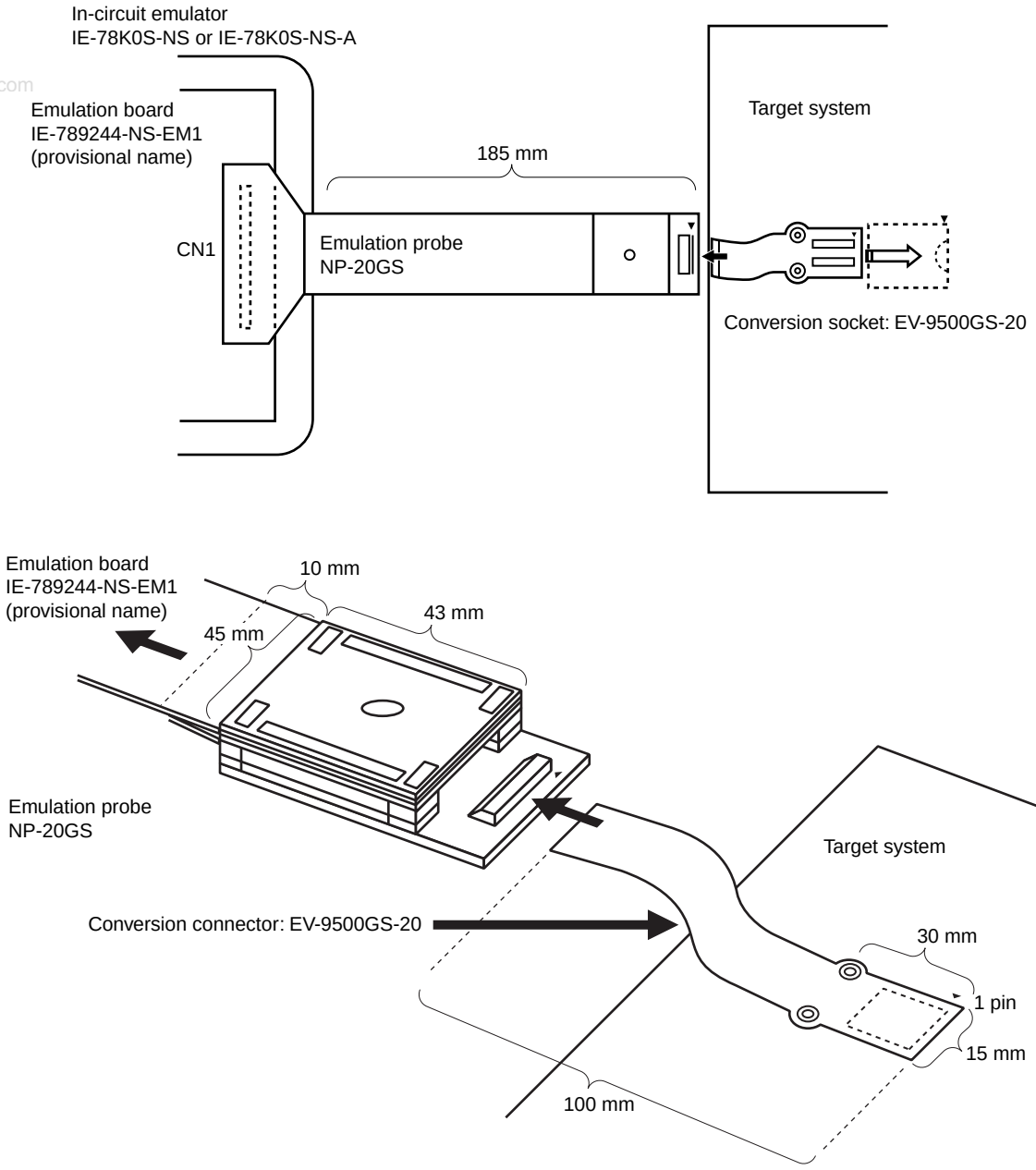
μS××××ID78K0S-NS

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT and compatibles	Japanese Windows	3.5" 2HD FD
BB13		English Windows	
AB17		Japanese Windows	CD-ROM
BB17		English Windows	

APPENDIX B NOTES ON TARGET SYSTEM DESIGN

The following show the conditions when connecting the emulation probe to the conversion connector and conversion socket. Follow the configuration below and consider the shape of parts to be mounted on the target system when designing a system.

Figure B-1. Connection Condition of Target



Remark The NP-20GS is a product made by Naito Densei Machida Mfg. Co., Ltd.

APPENDIX C REGISTER INDEX

C.1 Register Index (Register Name)

C.2 Register Index (Symbol)