User’s Manual

V850ES/PM1

32-Bit Single-Chip Microcontroller

Hardware

μPD703228

Document No. U16237EJ2V0UD00 (2nd edition)
Date Published September 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.

These commodities, technology or software, must be exported in accordance
with the export administration regulations of the exporting country.
Diversion contrary to the law of that country is prohibited.

- **The information in this document is current as of July, 2003. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. 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 product 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).

M8E 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

PREFACE

Readers

This manual is intended for users who wish to understand the functions of the V850ES/PM1 (μPD703228) and design application systems using this product.

Purpose

This manual is intended to give users an understanding of the hardware functions of the V850ES/PM1 shown in the Organization below.

Organization

This manual is divided into two parts: Hardware (this manual) and Architecture (**V850ES Architecture User's Manual**).

Hardware

- Pin functions
- CPU function
- On-chip peripheral functions
- Electrical specifications

Architecture

- Data types
- Register set
- Instruction format and instruction set
- Interrupts and exceptions
- Pipeline operation

How to Read This Manual

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

To find the details of a register where the name is known
→ Refer to **APPENDIX A REGISTER INDEX**.

To understand the details of an instruction function
→ Refer to the **V850ES Architecture User's Manual** available separately.

Register format
→ The name of the bit whose number is in angle brackets (<>) in the figure of the register format of each register is defined as a reserved word in the device file.

To understand the overall functions of the V850ES/PM1
→ Read this manual according to the **CONTENTS**.

To know the electrical specifications of the V850ES/PM1
→ Refer to **CHAPTER 20 ELECTRICAL SPECIFICATIONS**.

The mark ★ shows major revised points.

Conventions

Data significance: Higher digits on the left and lower digits on the right

Active low representation: $\overline{\text{xxx}}$ (overscore over pin or signal name)

Memory map address: Higher addresses on the top and lower addresses on the bottom

Note: Footnote for item marked with **Note** in the text

Caution: Information requiring particular attention

Remark: Supplementary information

Numeric representation: Binary ... xxxx or xxxxB
Decimal ... xxxx
Hexadecimal ... xxxxH

Prefix indicating power of 2 (address space, memory capacity): K (kilo): 2^{10} = 1,024
M (mega): 2^{20} = $1,024^{2}$
G (giga): 2^{30} = $1,024^{3}$

Related Documents

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

Documents related to V850ES/PM1

Document Name	Document No.
V850ES Architecture User’s Manual	U15943E
V850ES/PM1 Hardware User’s Manual	This manual

Documents related to development tools

Document Name		Document No.
IE-V850ES-G1 (In-Circuit Emulator)		U16313E
IE-703228-G1-EM1 (In-Circuit Emulator Option Board)		To be prepared
CA850 C Compiler Package Ver. 2.50 or Later ID850	Operation	U16053E
	C Language	U16054E
	Assembly Language	U16042E
PM plus Ver. 5.10		U16569E
ID850 Ver. 2.50 Integrated Debugger	Operation	U16217E
SM850 Ver. 2.50 System Simulator	Operation	U15182E
SM850 Ver. 2.00 System Simulator	External Part User Open Interface Specifications	U14873E
RX850 Real-Time OS Ver. 3.13 or Later	Basics	U13430E
	Installation	U13410E
	Technical	U13431E
RX850 Pro Real-Time OS Ver. 3.13	Basics	U13773E
	Installation	U13774E
	Technical	U13772E
RD850 Task Debugger Ver. 3.01		U13737E
RD850 Pro Task Debugger Ver. 3.01		U13916E
AZ850 System Performance Analyzer Ver. 3.0		U14410E

CONTENTS

CHAPTER 1 INTRODUCTION

The V850ES/PM1 is an NEC Electronics V850 Series single-chip microcontroller for real-time control.

1.1 Overview

The V850ES/PM1 is a 32-bit single-chip microcontroller that employs the V850ES CPU core and integrates peripheral functions such as ROM/RAM, timers/counters, serial interfaces, an A/D converter, and PWM.

The V850ES core is used as the CPU with peripheral functions, such as a ΔΣA/D converter, added.

The V850ES/PM1 features instructions ideal for digital servo control applications, such as multiplication instructions using a hardware multiplier, saturated operation instructions, and bit manipulation instructions, as well as basic instructions with a short real-time response speed and a 1-clock pitch. High-accuracy power measurement can be realized at a low cost by using the high-accuracy, 6-channel ΔΣA/D converter, making the V850ES/PM1 suitable for applications such as power meters and other measuring instruments.

1.2 Features

○ Minimum instruction execution time:	50 ns (main clock (f_X) = 20 MHz)
	100 ns (main clock (f_X) = 10 MHz)
	30.5 μs (subclock (f_{XT}) = 32.768 kHz)
○ General-purpose registers:	32 bits × 32 registers
○ CPU features:	Signed multiplication (16 × 16 → 32): 1 to 2 clocks
	Signed multiplication (32 × 32 → 64): 1 to 5 clocks
	Saturated operation (with overflow/underflow detection function)
	32-bit shift instruction: 1 clock
	Bit manipulation instruction
	Load/store instruction with long/short format
○ Memory space:	64 MB linear address space (for program/data)
	External expansion: Up to 8 MB
	(of which 1 MB is used as internal ROM space)
	Memory block division function: 2, 2, 4 MB (total: 3 blocks)
	Programmable wait function
	Idle state insertion function
• Internal memory:	RAM: 10 KB
	Mask ROM: 128 KB
• External bus interface:	Separate bus output
	8/16-bit data bus sizing function
	3-space chip select function
	Wait functions
	• Programmable wait function
	• External wait function
	Idle state function
○ Interrupts/exceptions:	Non-maskable interrupt: 1 source
	Maskable interrupts: 31 sources
	Software exception: 32 sources
	Exception trap: 1 source
○ I/O lines:	Total: 68 (I/O ports)
○ Timer function:	16-bit timer/event counter: 6 ch (PWM output)
	8-bit timer/event counter: 2 ch (connectable in cascade)
	Real-time counter (for watch): Subclock/main clock operation: 1 ch Counter for weeks, days, hours, minutes, and seconds, up to 4,095 weeks.
	Watchdog timer: 1 ch
○ PWM (Pulse Width Modulation):	4 ch
○ Serial interface:	Asynchronous serial interface (UART)
	Clocked serial interface (CSI)
	UART: 2 ch
	CSI: 2 ch
○ A/D converter:	16-bit resolution: 6 ch (12 inputs)
○ ROM correction:	4 places specifiable
○ Clock generator:	Main clock/subclock operation
	CPU clock: 5 steps (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, f_{XT})
○ Power save function:	HALT/IDLE/STOP/sub-IDLE/sub-STOP mode
○ Package:	100-pin plastic LQFP (fine pitch) (14 × 14)

1.3 Application Fields

Power meters, measuring instruments

1.4 Ordering Information

Part Number	Package	Internal ROM
μPD703228GC-××× -8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (128 KB)

Remarks 1. ××× indicates ROM code suffix.

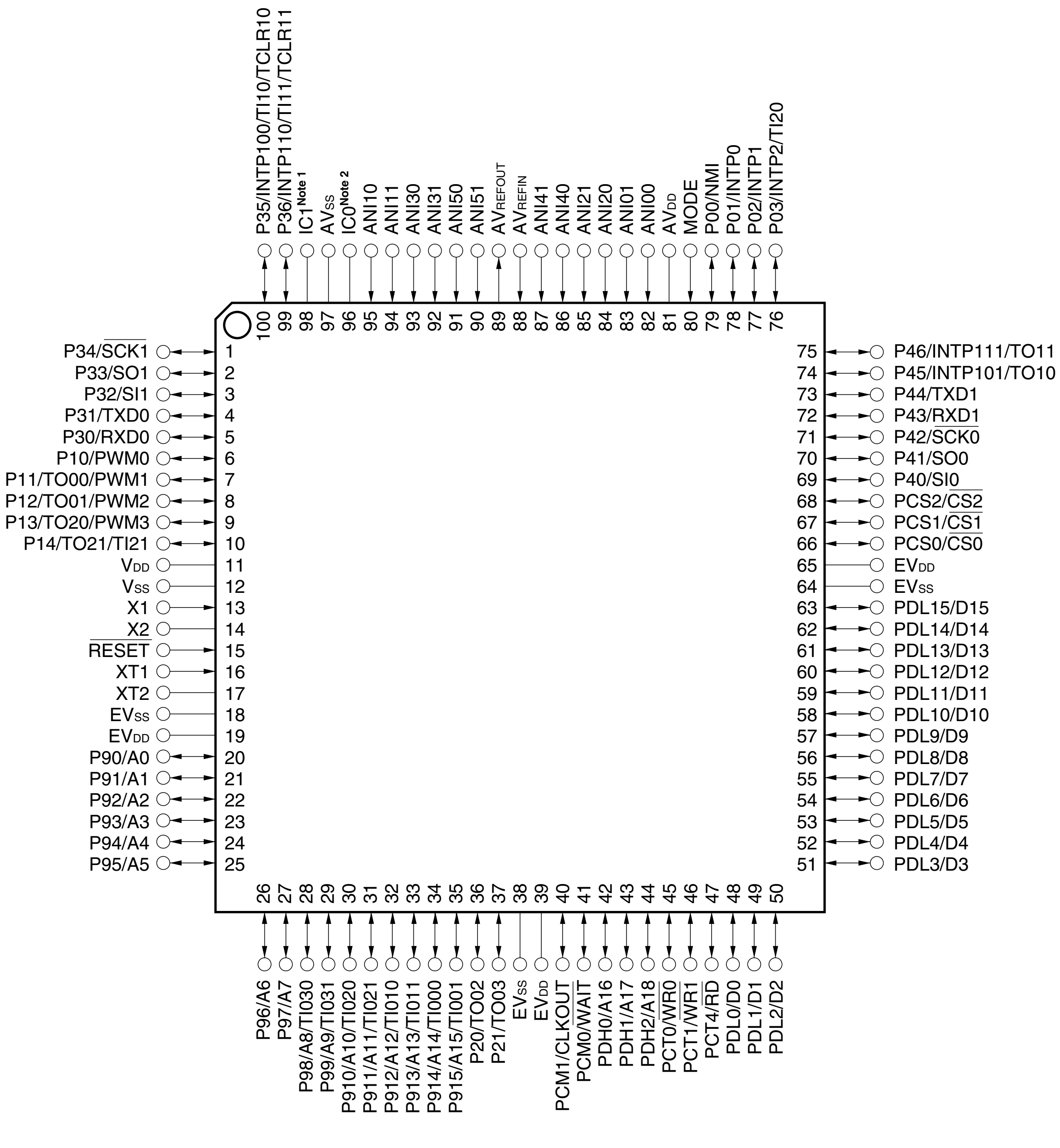
2. No ROMless model is available.

1.5 Pin Configuration (Top View)

○ V850ES/PM1

100-pin plastic LQFP (fine pitch) (14 × 14)

- μPD703228GC-×××-8EU



Notes 1. Connect this pin directly to V_{SS}.

2. Connect this pin directly to AV_{SS}.

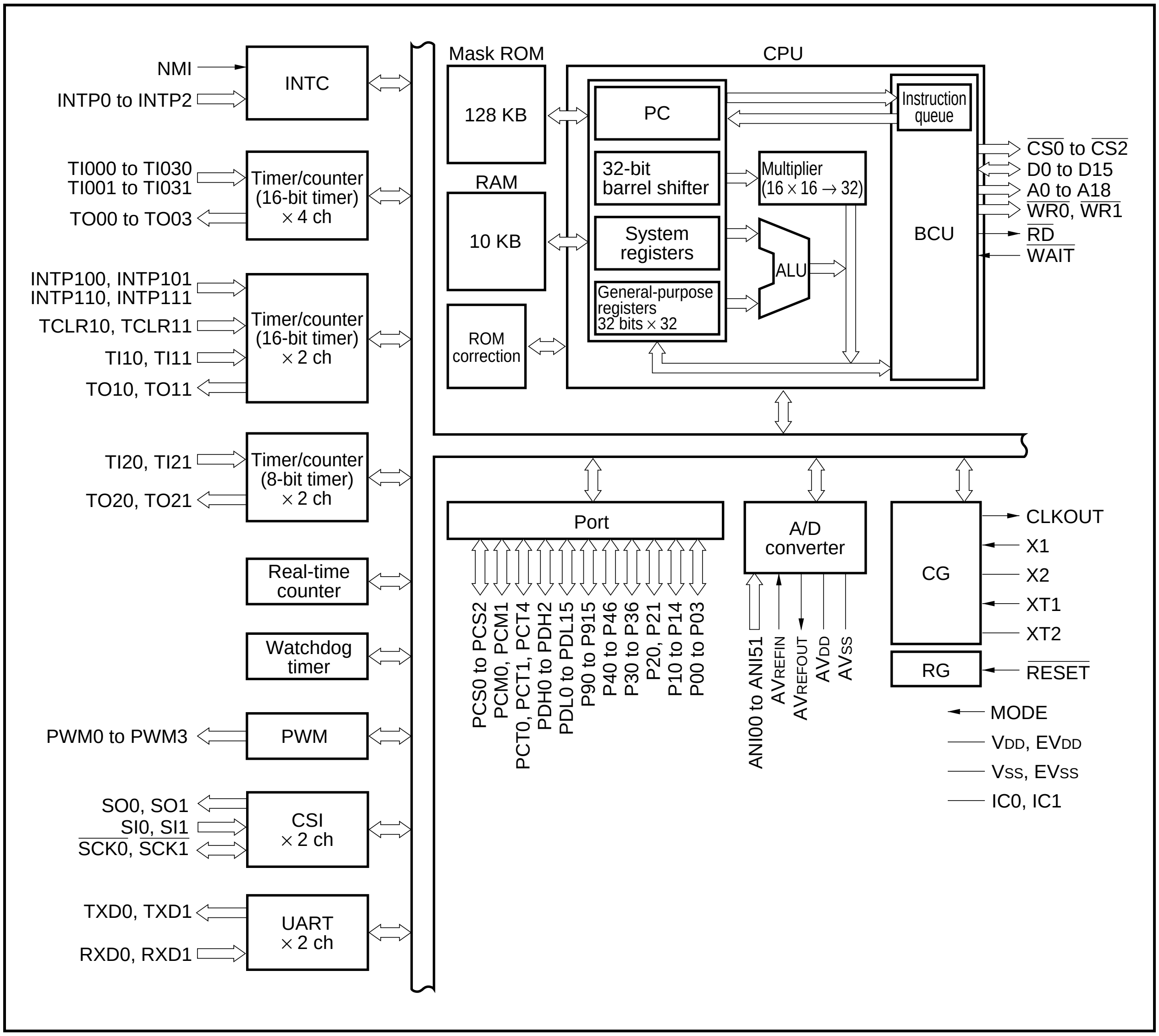
Pin Identification

A0 to A18:	Address bus
ANI00, ANI01, ANI10, ANI11, ANI20, ANI21, ANI30, ANI31, ANI40, ANI41, ANI50, ANI51:	Analog input
AV_{DD}:	Analog V_{DD}
AV_{REFIN}:	Analog reference voltage input
AV_{REFOUT}:	Analog reference voltage output
AV_{SS}:	Analog V_{SS}
CLKOUT:	Clock output
$\overline{CS0}$ to $\overline{CS2}$:	Chip select
D0 to D15:	Data bus
EV_{DD}:	Power supply for port
EV_{SS}:	Ground for port
IC0, IC1:	Internally connected
INTP0 to INTP2:	External interrupt input
INTP100, INTP101, INTP110, INTP111:	Timer input
MODE:	Operation mode select
NMI:	Non-maskable interrupt request
P00 to P03:	Port 0
P10 to P14:	Port 1
P20, P21:	Port 2
P30 to P36:	Port 3
P40 to P46:	Port 4
P90 to P915:	Port 9
PCM0, PCM1:	Port CM
PCS0 to PCS2:	Port CS
PCT0, PCT1, PCT4:	Port CT
PDH0 to PDH2:	Port DH
PDL0 to PDL15:	Port DL
PWM0 to PWM3:	Pulse width modulation
$\overline{RD}$:	Read strobe
$\overline{RESET}$:	Reset
RXD0, RXD1:	Receive data
$\overline{SCK0}$, $\overline{SCK1}$:	Serial clock
SI0, SI1:	Serial input
SO0, SO1:	Serial output
TCLR10, TCLR11:	Timer clear input
TI000, TI001, TI010, TI011, TI020, TI021, TI030, TI031, TI10, TI11, TI20, TI21:	Timer input
TO00 to TO03, TO10, TO11, TO20, TO21:	Timer output
TXD0, TXD1:	Transmit data
V_{DD}:	Power supply
V_{SS}:	Ground
$\overline{WAIT}$:	Wait
$\overline{WR0}$:	Lower byte write strobe
$\overline{WR1}$:	Upper byte write strobe
X1, X2:	Crystal for main clock
XT1, XT2:	Crystal for subclock

1.6 Function Block Configuration

1.6.1 Internal block diagram

- V850ES/PM1



1.6.2 Internal units

(1) CPU

The CPU can execute almost all instruction processing, such as address calculation, arithmetic logic operations, and data transfer, with 1 clock, using a 5-stage pipeline.

The CPU has dedicated hardware units such as a multiplier (16 bits × 16 bits → 32 bits) and a barrel shifter (32 bits) to speed up complicated processing.

(2) Bus control unit (BCU)

The BCU starts the required external bus cycles in accordance with the physical address obtained by the CPU. If the CPU does not request the start of a bus cycle when an instruction is fetched from the external memory area, the BCU generates a prefetch address and prefetches an instruction code. The prefetched instruction code is loaded to the internal instruction queue.

(3) ROM

This is 128 KB mask ROM mapped to addresses 0000000H to 001FFFFH. The CPU can access the ROM with 1 clock when an instruction is fetched.

(4) RAM

This is 10 KB RAM mapped to addresses 3FFC800H to 3FFEFFFH. It can be accessed by the CPU with 1 clock when data is accessed.

(5) Interrupt controller (INTC)

The INTC processes hardware interrupt requests (NMI, INTP0 to INTP2) from the internal peripheral hardware and external sources. Eight levels of priority can be specified for these interrupt requests. The INTC can also control multiple interrupt servicing.

(6) Clock generator (CG)

Two oscillators, the main clock oscillator and subclock oscillator, are provided and generate the main clock oscillation frequency (f_X) and subclock frequency (f_{XT}).

The CPU clock frequency (f_{CPU}) can be selected from five types of clocks, f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, and f_{XT}.

(7) Timer/counter

A six-channel 16-bit timer/event counter is available, enabling pulse interval and frequency measurement and programmable pulse output.

A two-channel 8-bit timer/event counter is also available and can be cascaded as a 16-bit timer.

(8) Real-time counter (for watch)

This real-time counter counts the reference time from the subclock and can also be used as an interval timer. Week, day, hour, minute, and second counters are provided, and up to 4,095 weeks can be counted.

(9) Watchdog timer

A watchdog timer that detects program hang-up and system errors is provided.

This watchdog timer can also be used as an interval timer.

When used as a watchdog timer, an internal reset request signal (WDTRES) is generated if the watchdog timer overflows. When used as an interval timer, a maskable interrupt request signal (INTWDTM) is generated when the timer overflows.

(10) PWM (Pulse Width Modulation)

Four PWM signal output channels are available. The resolution is selectable from 8 to 10, or 12 bits.

(11) Serial interface

The V850ES/PM1 has asynchronous serial interfaces (UART0 and UART1) and clocked serial interfaces (CSI0 and CSI1) as the serial interfaces. The V850ES/PM1 can use up to four channels at the same time.

UART0 and UART1 transfer data using the TXD0, TXD1, RXD0, and RXD1 pins.

CSI0 and CSI1 transfer data using the SO0, SO1, SI0, SI1, $\overline{\text{SCK0}}$, and $\overline{\text{SCK1}}$ pins.

(12) A/D converter

The V850ES/PM1 has 12 analog input pins and a six-channel 16-bit A/D converter that uses a $\Delta\Sigma$ conversion method. It also has a function to input/output a reference voltage, and includes six A/D conversion result registers.

(13) ROM correction

This is a function to replace part of the program in the mask ROM with program in the internal RAM for execution. The program can be corrected at up to four places.

(14) Ports

Some port pins have a control function as well as a general-purpose port function, as shown below.

Port	I/O	Alternate Function
P0	4-bit I/O	NMI, external interrupt, timer input
P1	5-bit I/O	PWM output, timer I/O
P2	2-bit I/O	Timer output
P3	7-bit I/O	Serial interface, timer input, timer trigger
P4	7-bit I/O	Serial interface, timer output, timer trigger
P9	16-bit I/O	External address bus, timer input
PCM	2-bit I/O	External bus control signal
PCS	3-bit I/O	Chip select output
PCT	3-bit I/O	External bus control signal
PDH	3-bit I/O	External address bus
PDL	16-bit I/O	External data bus

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

This chapter explains the names and functions of the pins in the V850ES/PM1, classified into port pins and non-port pins.

Two power supplies are available for the pin I/O buffers: AV_{DD} and EV_{DD}. The relationship between the power supplies and pins is shown below.

Table 2-1. I/O Buffer Power Supply for Each Pin

Power Supply	Corresponding Pin
AV_{DD}	ANIn0, ANIn1 (n = 0 to 5)
EV_{DD}	Ports 0 to 4, 9, CM, CS, CT, DH, DL, $\overline{\text{RESET}}$

(1) Port pins

(1/2)

Pin Name	Pin No.	I/O	On-Chip Pull-up Resistor	Function	Alternate Function
P00	79	I/O	Provided	Port 0 4-bit I/O port Input/output can be specified in 1-bit units.	NMI
P01	78				INTP0
P02	77				INTP1
P03	76				INTP2/TI20
P10	6	I/O	Provided	Port 1 5-bit I/O port Input/output can be specified in 1-bit units.	PWM0
P11	7				TO00/PWM1
P12	8				TO01/PWM2
P13	9				TO20/PWM3
P14	10				TO21/TI21
P20	36	I/O	Provided	Port 2 2-bit I/O port Input/output can be specified in 1-bit units.	TO02
P21	37				TO03
P30	5	I/O	Provided	Port 3 7-bit I/O port Input/output can be specified in 1-bit units.	RXD0
P31	4				TXD0
P32	3				SI1
P33	2				SO1
P34	1				$\overline{\text{SCK1}}$
P35	100				INTP100/TI10/TCLR10
P36	99				INTP110/TI11/TCLR11
P40	69	I/O	Provided	Port 4 7-bit I/O port Input/output can be specified in 1-bit units.	SI0
P41	70				SO0
P42	71				$\overline{\text{SCK0}}$
P43	72				RXD1
P44	73				TXD1
P45	74				INTP101/TO10
P46	75				INTP111/TO11
P90	20	I/O	None	Port 9 16-bit I/O port Input/output can be specified in 1-bit units.	A0
P91	21				A1
P92	22				A2
P93	23				A3
P94	24				A4
P95	25				A5
P96	26				A6
P97	27				A7
P98	28				A8/TI030
P99	29				A9/TI031
P910	30				A10/TI020
P911	31				A11/TI021
P912	32				A12/TI010
P913	33				A13/TI011
P914	34				A14/TI000
PDL15	35				A15/TI001

(2/2)

Pin Name	Pin No.	I/O	On-Chip Pull-up Resistor	Function	Alternate Function
PCM0	41	I/O	None	Port CM 2-bit I/O port Input/output can be specified in 1-bit units.	$\overline{\text{WAIT}}$
PCM1	40				CLKOUT
PCS0	66	I/O	None	Port CS 3-bit I/O port Input/output can be specified in 1-bit units.	$\overline{\text{CS0}}$
PCS1	67				$\overline{\text{CS1}}$
PCS2	68				$\overline{\text{CS2}}$
PCT0	45	I/O	None	Port CT 3-bit I/O port Input/output can be specified in 1-bit units.	$\overline{\text{WR0}}$
PCT1	46				$\overline{\text{WR1}}$
PCT4	47				$\overline{\text{RD}}$
PDH0	42	I/O	None	Port DH 3-bit I/O port Input/output can be specified in 1-bit units.	A16
PDH1	43				A17
PDH2	44				A18
PDL0	48	I/O	None	Port DL 16-bit I/O port Input/output can be specified in 1-bit units.	D0
PDL1	49				D1
PDL2	50				D2
PDL3	51				D3
PDL4	52				D4
PDL5	53				D5
PDL6	54				D6
PDL7	55				D7
PDL8	56				D8
PDL9	57				D9
PDL10	58				D10
PDL11	59				D11
PDL12	60				D12
PDL13	61				D13
PDL14	62				D14
PDL15	63				D15

(2) Non-port pins

(1/3)

Pin Name	Pin No.	I/O	On-Chip Pull-up Resistor	Function	Alternate Function
A0	20	Output	None	Address bus for external memory	P90
A1	21				P91
A2	22				P92
A3	23				P93
A4	24				P94
A5	25				P95
A6	26				P96
A7	27				P97
A8	28				P98/TI030
A9	29				P99/TI031
A10	30				P910/TI020
A11	31				P911/TI021
A12	32				P912/TI010
A13	33				P913/TI011
A14	34				P914/TI000
A15	35				P915/TI001
A16	42				PDH0
A17	43				PDH1
A18	44				PDH2
D0 to D15	48 to 63	I/O	None	Data bus for external memory	PDL0 to PDL15
ANI00	82	Input	None	Analog voltage input for A/D converter	–
ANI01	83				–
ANI10	95				–
ANI11	94				–
ANI20	84				–
ANI21	85				–
ANI30	93				–
ANI31	92				–
ANI40	86				–
ANI41	87				–
ANI50	91				–
ANI51	90				–
AV$_{DD}$	81	–	–	Positive power supply for A/D converter (same potential as V$_{DD}$)	–
AV$_{REFIN}$	88	Input	–	Reference voltage input for A/D converter	–
AV$_{REFOUT}$	89	Output		Reference voltage output for A/D converter	–
AV$_{SS}$	97	–	–	Ground potential for A/D converter (same potential as V$_{SS}$)	–
CLKOUT	40	Output	None	Internal system clock output	PCM1
$\overline{\text{CS0}}$ to $\overline{\text{CS2}}$	66 to 68	Output	None	Chip select output	PCS0 to PCS2
EV$_{DD}$	19, 39, 65	–	–	Positive power supply for external device (same potential as V$_{DD}$)	–
EV$_{SS}$	18, 38, 64	–	–	Ground potential for external device (same potential as V$_{SS}$)	–

(2/3)

Pin Name	Pin No.	I/O	On-Chip Pull-up Resistor	Function	Alternate Function
IC0	96	–	–	Internally connected (connect this pin directly to AV$_{\text{SS}}$)	–
IC1	98	–	–	Internally connected (connect this pin directly to V$_{\text{SS}}$)	–
INTP0	78	Input	Provided	External interrupt request input (maskable, analog noise elimination)	P01
INTP1	77				P02
INTP2	76				P03/TI20
INTP100	100	Input	Provided	Capture trigger input (TM10)	P35/TI10/TCLR10
INTP101	74				P45/TO10
INTP110	99			Capture trigger input (TM11)	P36/TI11/TCLR11
INTP111	75				P46/TO11
MODE	80	Input	None	Operation mode specification	–
NMI	79	Input	Provided	External interrupt request input (non-maskable, analog noise elimination)	P00
PWM0	6	Output	Provided	PWM output	P10
PWM1	7				P11/TO00
PWM2	8				P12/TO01
PWM3	9				P13/TO20
$\overline{\text{RD}}$	47	Output	None	Read strobe signal output for external memory	PCT4
$\overline{\text{RESET}}$	15	Input	–	System reset input	–
RXD0	5	Input	Provided	Serial receive data input (UART0)	P30
RXD1	72			Serial receive data input (UART1)	P43
$\overline{\text{SCK0}}$	71	I/O	Provided	Serial clock I/O (CSI0)	P42
$\overline{\text{SCK1}}$	1			Serial clock I/O (CSI1)	P34
SI0	69	Input	Provided	Serial receive data input (CSI0)	P40
SI1	3			Serial receive data input (CSI1)	P32
SO0	70	Output	Provided	Serial transmit data output (CSI0)	P41
SO1	2			Serial transmit data output (CSI1)	P33
TCLR10	100	Input	Provided	Timer clear input (TM10)	P35/INTP100/TI10
TCLR11	99			Timer clear input (TM11)	P36/INTP110/TI11
TI000	34	Input	None	External event/clock input (TM00)	P914/A14
TI001	35			External event input (TM00)	P915/A15
TI010	32			External event/clock input (TM01)	P912/A12
TI011	33			External event input (TM01)	P913/A13
TI020	30			External event/clock input (TM02)	P910/A10
TI021	31			External event input (TM02)	P911/A11
TI030	28			External event/clock input (TM03)	P98/A8
TI031	29			External event input (TM03)	P99/A9
TI10	100	Input	Provided	External clock input (TM10)	P35/INTP100/TCLR10
TI11	99			External clock input (TM11)	P36/INTP110/TCLR11
TI20	76	Input	Provided	External clock input (TM20)	P03/INTP2
TI21	10			External clock input (TM21)	P14/TO21
TO00	7	Output	Provided	Timer output (TM00)	P11/PWM1
TO01	8			Timer output (TM01)	P12/PWM2
TO02	36			Timer output (TM02)	P20

(3/3)

Pin Name	Pin No.	I/O	On-Chip Pull-up Resistor	Function	Alternate Function
TO03	37	Output	Provided	Timer output (TM03)	P21
TO10	74			Timer output (TM10)	P45/INTP101
TO11	75			Timer output (TM11)	P46/INTP111
TO20	9	Output	Provided	Timer output (TM20)	P13/PWM3
TO21	10	Output	Provided	Timer output (TM21)	P14/TI21
TXD0	4	Output	Provided	Serial transmit data output (UART0)	P31
TXD1	73			Serial transmit data output (UART1)	P44
V_{DD}	11	–	–	Positive power supply for internal	–
V_{SS}	12	–	–	Ground potential for internal	–
$\overline{\text{WAIT}}$	41	Input	None	External wait input	PCM0
$\overline{\text{WR0}}$	45	Output	None	Write strobe for external memory (lower 8 bits)	PCT0
$\overline{\text{WR1}}$	46			Write strobe for external memory (higher 8 bits)	PCT1
X1	13	Input	None	Connection of resonator for main clock	–
X2	14	–			–
XT1	16	Input	None	Connection of resonator for subclock	–
XT2	17	–			–

2.2 Pin Status

The operating status of each pin in each operation mode is shown below.

★ **Table 2-2. Operating Status of Each Pin in Each Operation Mode**

Bus Control Pins	Reset$^{\text{Note 1}}$	HALT Mode	IDLE and STOP Modes	Idle State$^{\text{Note 2}}$
D0 to D15	Hi-Z	Hi-Z	Hi-Z	Retained
A16 to A18	Hi-Z	Undefined	Hi-Z	Retained
A0 to A15	Hi-Z	Undefined	Hi-Z	Retained
$\overline{\text{WAIT}}$	Hi-Z	–	–	–
CLKOUT	Hi-Z	Operates	L	Operates
$\overline{\text{CS0}}$ to $\overline{\text{CS2}}$	Hi-Z	H	H	Retained
$\overline{\text{WR0}}$, $\overline{\text{WR1}}$	Hi-Z	H	H	H
$\overline{\text{RD}}$	Hi-Z	H	H	H

Notes 1. Because the bus control pins function alternately as port pins, they are initialized to the input mode (port mode) in the single chip mode.

Signals other than the CLKOUT signal are initialized to the control mode while the ROMless mode is reset.

2. Indicates the pin status in the idle state that is inserted after the T2 state.

Remark

Hi-Z:	High impedance
Retained:	Status in external bus cycle immediately before is retained.
L:	Low-level output
H:	High-level output
–:	Input not sampled (not acknowledged)

2.3 Types of Pin I/O Circuits, I/O Buffer Power Supplies, and Connection of Unused Pins

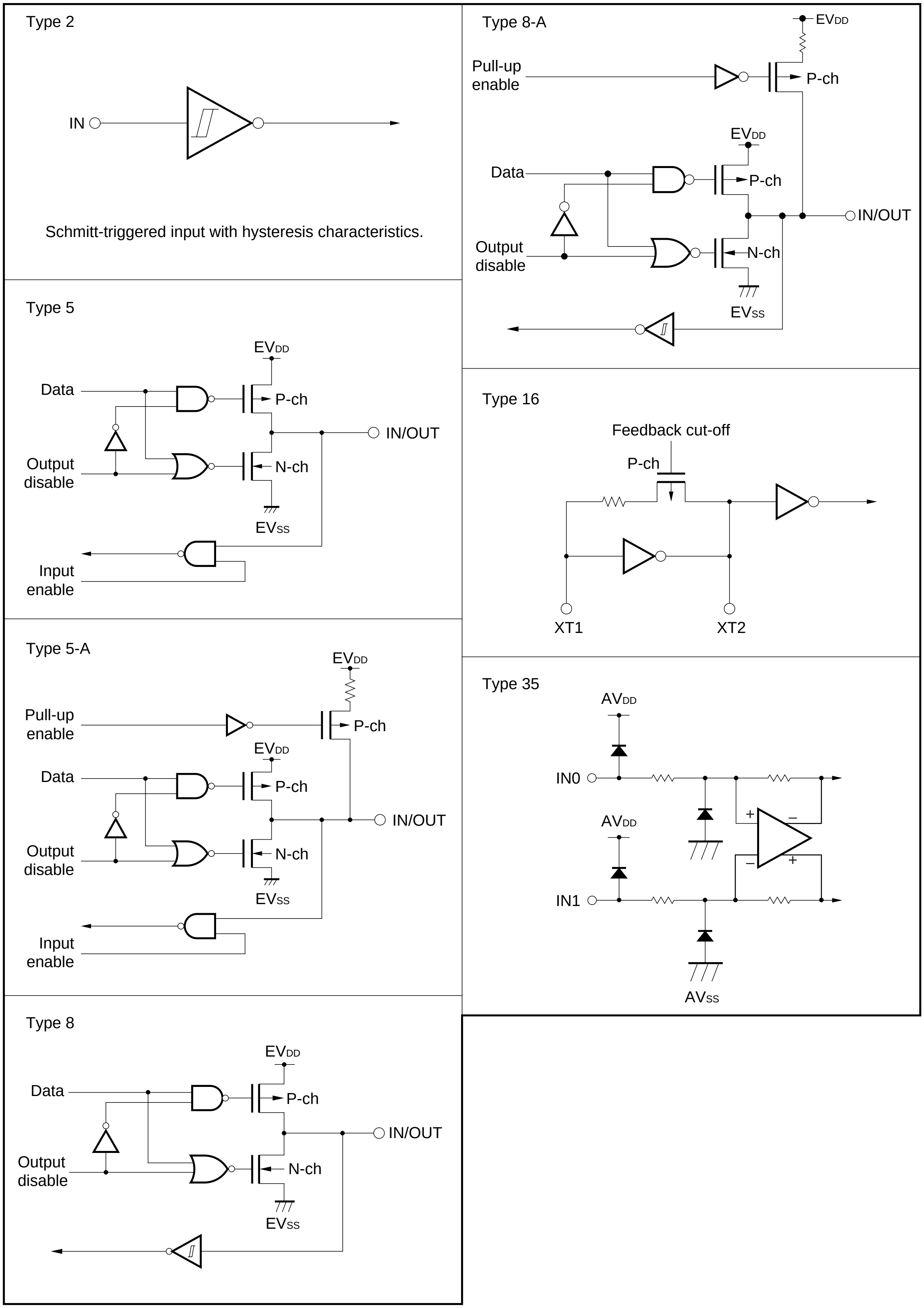
(1/2)

Pin	Alternate Function	Pin No.	I/O Circuit Type	Recommended Connection
P00	NMI	79	8-A	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P01, P02	INTP0, INTP1	78, 77		
P03	INTP2/TI20	76		
P10	PWM0	6	5-A	
P11	TO00/PWM1	7		
P12	TO01/PWM2	8		
P13	TO20/PWM3	9		
P14	TO21/TI21	10	8-A	
P20, P21	TO02, TO03	36, 37	5-A	
P30	RXD0	5	8-A	
P31	TXD0	4	5-A	
P32	SI1	3	8-A	
P33	SO1	2	5-A	
P34	$\overline{\text{SCK1}}$	1	8-A	
P35	INTP100/TI10/TCLR10	100		
P36	INTP110/TI11/TCLR11	99		
P40	SI0	69		
P41	SO0	70	5-A	
P42	$\overline{\text{SCK0}}$	71	8-A	
P43	RXD1	72		
P44	TXD1	73	5-A	
P45	INTP101/TO10	74	8-A	
P46	INTP111/TO11	75		
P90 to P97	A0 to A7	20 to 27	5	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
★ P98	A8/TI030	28	8	
★ P99	A9/TI031	29		
★ P910	A10/TI020	30		
★ P911	A11/TI021	31		
★ P912	A12/TI010	32		
★ P913	A13/TI011	33		
★ P914	A14/TI000	34		
★ P915	A15/TI001	35		
PCM0	$\overline{\text{WAIT}}$	41	5	
PCM1	CLKOUT	40		
PCS0 to PCS2	$\overline{\text{CS0}}$ to $\overline{\text{CS2}}$	66 to 68		
PCT0, PCT1	$\overline{\text{WR0}}$, $\overline{\text{WR1}}$	45, 46		
PCT4	$\overline{\text{RD}}$	47		
PDH0 to PDH2	A16 to A18	42 to 44		
PDL0 to PDL15	D0 to D15	48 to 63		

(2/2)

Pin	Alternate Function	Pin No.	I/O Circuit Type	Recommended Connection
ANI00, ANI01, ANI10, ANI11, ANI20, ANI21, ANI30, ANI31, ANI40, ANI41, ANI50, ANI51	–	82 to 87, 90 to 95	35	Connect to AV_{DD} or AV_{SS} via a resistor.
AV_{DD}	–	81	–	–
AV_{REFIN}	–	88	–	Connect to AV_{SS} via a resistor.
AV_{REFOUT}	–	89	–	Leave open.
AV_{SS}	–	97	–	–
EV_{DD}	–	19, 39, 65	–	–
EV_{SS}	–	18, 38, 64	–	–
IC0, IC1	–	96, 98	–	–
$\overline{\text{RESET}}$	–	15	2	–
MODE	–	80	2	–
V_{DD}	–	11	–	–
V_{SS}	–	12	–	–
X1	–	13	–	–
X2	–	14	–	–
XT1	–	15	16	Connect to V_{SS} via a resistor.
XT2	–	17	16	Leave open.

Figure 2-1. Pin I/O Circuits



★

CHAPTER 3 CPU FUNCTION

The CPU of the V850ES/PM1 is based on RISC architecture and executes almost all instructions with one clock by using a 5-stage pipeline.

3.1 Features

○ Minimum instruction execution time: 50 ns (at 20 MHz operation: 3.0 to 3.6 V)
100 ns (at 10 MHz operation: 2.7 to 3.6 V)
30.5 μs (with subclock (f_{XT}) = 32.768 kHz operation: 2.2 to 3.6 V)

○ Memory space Program (physical address) space: 64 MB linear
Data (logical address) space: 4 GB linear
• Memory block division function: 2, 2, 4 MB/total: 3 blocks
Each block can be accessed in 512 KB units.

○ General-purpose registers: 32 bits × 32 registers

○ Internal 32-bit architecture

○ 5-stage pipeline control

○ Multiplication/division instruction

○ Saturation operation instruction

○ 32-bit shift instruction: 1 clock

○ Load/store instruction with long/short format

○ Four types of bit manipulation instructions

- SET1
- CLR1
- NOT1
- TST1

3.2 CPU Register Set

The registers of the V850ES/PM1 can be classified into two types: general-purpose program registers and dedicated system registers. All the registers are 32 bits wide.

For details, refer to the **V850ES Architecture User's Manual**.

(1) Program register set

31	0
r0	(Zero register)
r1	(Assembler-reserved register)
r2	
r3	(Stack pointer (SP))
r4	(Global pointer (GP))
r5	(Text pointer (TP))
r6	
r7	
r8	
r9	
r10	
r11	
r12	
r13	
r14	
r15	
r16	
r17	
r18	
r19	
r20	
r21	
r22	
r23	
r24	
r25	
r26	
r27	
r28	
r29	
r30	(Element pointer (EP))
r31	(Link pointer (LP))

31	0
PC	(Program counter)

(2) System register set

31	0
EIPC	(Interrupt status saving register)
EIPSW	(Interrupt status saving register)

FEPC	(NMI status saving register)
FEPSW	(NMI status saving register)

ECR	(Interrupt source register)

PSW	(Program status word)

CTPC	(CALLT execution status saving register)
CTPSW	(CALLT execution status saving register)

DBPC	(Exception/debug trap status saving register)
DBPSW	(Exception/debug trap status saving register)

CTBP	(CALLT base pointer)

3.2.1 Program register set

The program registers include general-purpose registers and a program counter.

(1) General-purpose registers (r0 to r31)

Thirty-two general-purpose registers, r0 to r31, are available. Any of these registers can be used to store a data variable or an address variable.

However, r0 and r30 are implicitly used by instructions and care must be exercised when these registers are used. r0 always holds 0 and is used for an operation that uses 0 or addressing of offset 0. r30 is used by the SLD and SST instructions as a base pointer when these instructions access the memory. r1, r3 to r5, and r31 are implicitly used by the assembler and C compiler. When using these registers, save their contents for protection, and then restore the contents after using the registers. r2 is sometimes used by the real-time OS. If the real-time OS does not use r2, it can be used as a register for variables.

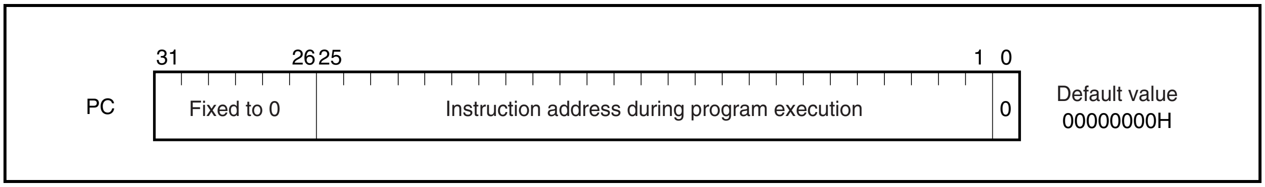
Table 3-1. Program Registers

Name	Usage	Operation
r0	Zero register	Always holds 0.
r1	Assembler-reserved register	Used as working register to create 32-bit immediate data
r2	Register for address/data variable (if real-time OS does not use r2)	
r3	Stack pointer	Used to create a stack frame when a function is called
r4	Global pointer	Used to access a global variable in the data area
r5	Text pointer	Used as register that indicates the beginning of a text area (area where program codes are located)
r6 to r29	Register for address/data variable	
r30	Element pointer	Used as base pointer to access memory
r31	Link pointer	Used when the compiler calls a function
PC	Program counter	Holds the instruction address during program execution

(2) Program counter (PC)

The program counter holds the instruction address during program execution. The lower 26 bits of this register are valid. Bits 31 to 26 are fixed to 0. A carry from bit 25 to 26 is ignored even if it occurs.

Bit 0 is fixed to 0. This means that execution cannot branch to an odd address.



3.2.2 System register set

The system registers control the status of the CPU and hold interrupt information.

These registers can be read or written by using system register load/store instructions (LDSR and STSR), using the system register numbers listed below.

Table 3-2. System Register Numbers

System Register Number	System Register Name	Operand Specification	
		LDSR instruction	STSR instruction
0	Interrupt status saving register (EIPC)[Note 1]	√	√
1	Interrupt status saving register (EIPSW)[Note 1]	√	√
2	NMI status saving register (FEPC)	√	√
3	NMI status saving register (FEPSW)	√	√
4	Interrupt source register (ECR)	×	√
5	Program status word (PSW)	√	√
6 to 15	Reserved for future function expansion (operation is not guaranteed if these registers are accessed)	×	×
16	CALLT execution status saving register (CTPC)	√	√
17	CALLT execution status saving register (CTPSW)	√	√
18	Exception/debug trap status saving register (DBPC)	√[Note 2]	√
19	Exception/debug trap status saving register (DBPSW)	√[Note 2]	√
20	CALLT base pointer (CTBP)	√	√
21 to 31	Reserved for future function expansion (operation is not guaranteed if these registers are accessed)	×	×

Notes 1. Because only one set of this register is available, the contents of this register must be saved by program if multiple interrupts are enabled.

2. These registers can be accessed only within the period from the DBTRAP instruction to the DBRETI instruction.

Caution **Even if EIPC or FEPC, or bit 0 of CTPC is set to 1 by the LDSR instruction, bit 0 is ignored when execution is returned to the main routine by the RETI instruction after interrupt servicing (this is because bit 0 of the PC is fixed to 0). Set an even value to EIPC, FEPC, and CTPC (bit 0 = 0).**

Remark √: Can be accessed
×: Access prohibited

(1) Interrupt status saving registers (EIPC and EIPSW)

EIPC and EIPSW are used to save the status when an interrupt occurs.

If a software exception or a maskable interrupt occurs, the contents of the program counter (PC) are saved to EIPC, and the contents of the program status word (PSW) are saved to EIPSW (these contents are saved to the NMI status saving registers (FEPC and FEPSW) if a non-maskable interrupt occurs).

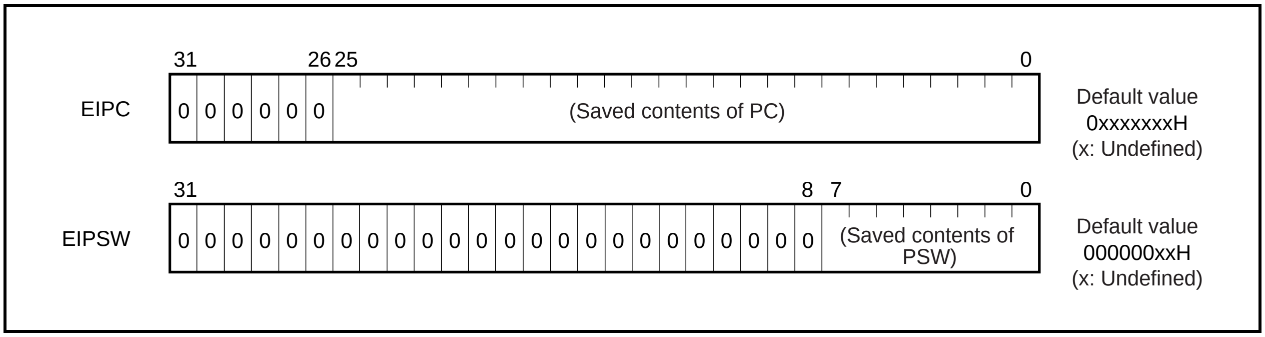
The address of the instruction next to the one of the instruction under execution, except some instructions (see **16.8 Periods in Which Interrupts Are Not Acknowledged by CPU**), is saved to EIPC when a software exception or a maskable interrupt occurs.

The current contents of the PSW are saved to EIPSW.

Because only one set of interrupt status saving registers is available, the contents of these registers must be saved by program when multiple interrupts are enabled.

Bits 31 to 26 of EIPC and bits 31 to 8 of EIPSW are reserved for future function expansion (these bits are always fixed to 0).

The values of EIPC and EIPSW are restored to the PC and PSW respectively by the RETI instruction.



(2) NMI status saving registers (FEPC and FEPSW)

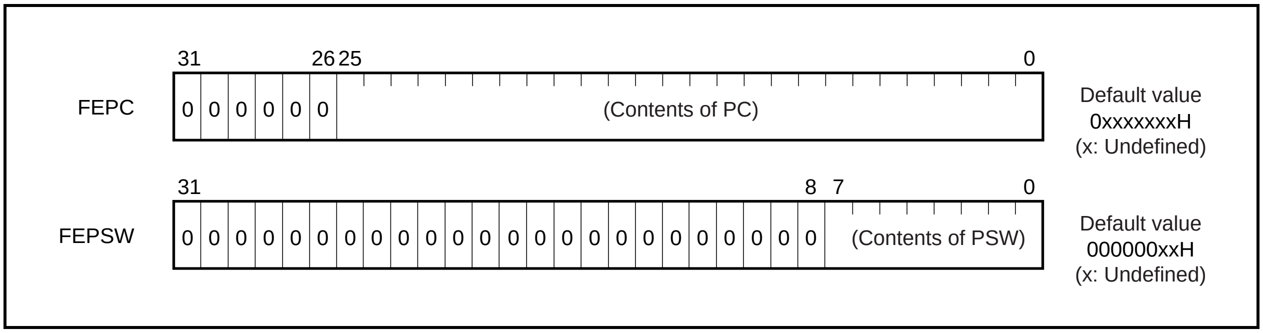
FEPC and FEPSW are used to save the status when a non-maskable interrupt (NMI) occurs.

If an NMI occurs, the contents of the program counter (PC) are saved to FEPC, and those of the program status word (PSW) are saved to FEPSW.

The address of the instruction next to the one of the instruction under execution, except some instructions, is saved to FEPC when an NMI occurs.

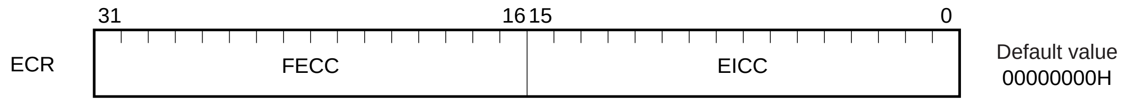
The current contents of the PSW are saved to FEPSW.

Bits 31 to 26 of FEPC and bits 31 to 8 of FEPSW are reserved for future function expansion (these bits are always fixed to 0).



(3) Interrupt source register (ECR)

The interrupt source register (ECR) holds the source of an exception or interrupt if an exception or interrupt occurs. This register holds the exception code of each interrupt source. Because this register is a read-only register, data cannot be written to this register using the LDSR instruction.



Bit position	Bit name	Meaning
31 to 16	FECC	Exception code of non-maskable interrupt (NMI)
15 to 0	EICC	Exception code of exception or maskable interrupt

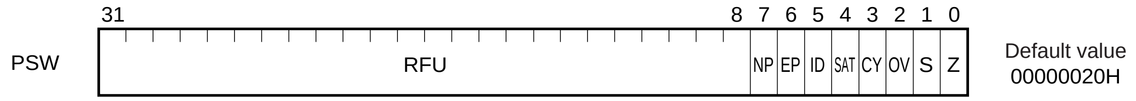
(4) Program status word (PSW)

The program status word (PSW) is a collection of flags that indicate the status of the program (result of instruction execution) and the status of the CPU.

If the contents of a bit of this register are changed by using the LDSR instruction, the new contents are validated immediately after completion of LDSR instruction execution. During the period in which the PSW is being accessed by the LDSR instruction, acknowledgment of interrupt requests is held pending.

Bits 31 to 8 of this register are reserved for future function expansion (these bits are fixed to 0).

(1/2)



Bit position	Flag name	Meaning
31 to 8	RFU	Reserved field. Fixed to 0.
7	NP	Indicates that a non-maskable interrupt (NMI) is being serviced. This bit is set to 1 when an NMI request is acknowledged, disabling multiple interrupts. 0: NMI is not being serviced. 1: NMI is being serviced.
6	EP	Indicates that an exception is being processed. This bit is set to 1 when an exception occurs. Even if this bit is set, interrupt requests are acknowledged. 0: Exception is not being processed. 1: Exception is being processed.
5	ID	Indicates whether a maskable interrupt can be acknowledged. 0: Interrupt enabled (EI) 1: Interrupt disabled (DI)

(2/2)

Bit position	Flag name	Meaning
4	SAT[Note]	Indicates that the result of a saturation operation has overflowed and is saturated. Because this is a cumulative flag, it is set to 1 when the result of a saturation operation instruction is saturated, and is not cleared to 0 even if the subsequent operation result is not saturated. Use the LDSR instruction to clear this bit. This flag is neither set to 1 nor cleared to 0 by execution of an arithmetic operation instruction. 0: Not saturated 1: Saturated
3	CY	Indicates whether a carry or a borrow occurs as a result of an operation. 0: Carry or borrow does not occur. 1: Carry or borrow occurs.
2	OV[Note]	Indicates whether an overflow occurs during operation. 0: Overflow does not occur. 1: Overflow occurs.
1	S[Note]	Indicates whether the result of an operation is negative. 0: The result is positive or 0. 1: The result is negative.
0	Z	Indicates whether the result of an operation is 0. 0: The result is not 0. 1: The result is 0.

Note The result of the operation that has performed saturation processing is determined by the contents of the OV and S flags. The SAT flag is set to 1 only when the OV flag is set to 1 when a saturation operation is performed.

Status of Operation Result	Flag Status			Result of Operation of Saturation Processing
	SAT	OV	S	
Maximum positive value is exceeded.	1	1	0	7FFFFFFFH
Maximum negative value is exceeded.	1	1	1	80000000H
Positive (maximum value is not exceeded)	Holds value before operation	0	0	Operation result itself
Negative (maximum value is not exceeded)			1	

(5) CALLT execution status saving registers (CTPC and CTPSW)

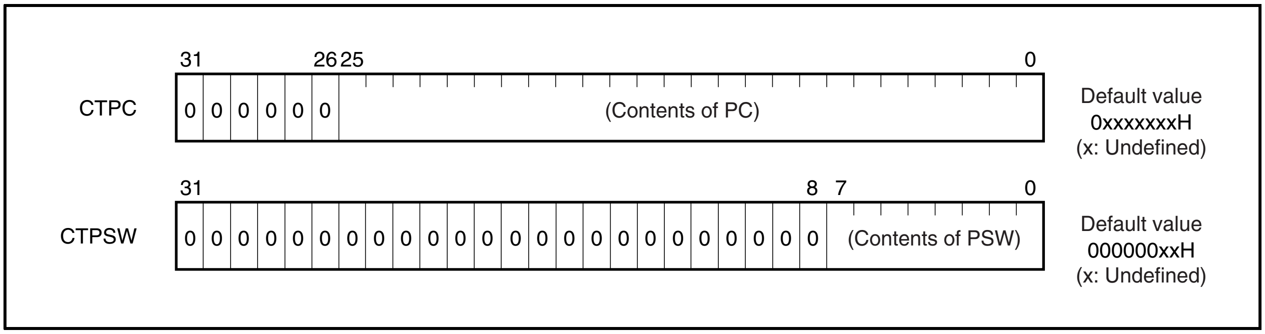
CTPC and CTPSW are CALLT execution status saving registers.

When the CALLT instruction is executed, the contents of the program counter (PC) are saved to CTPC, and those of the program status word (PSW) are saved to CTPSW.

The contents saved to CTPC are the address of the instruction next to CALLT.

The current contents of the PSW are saved to CTPSW.

Bits 31 to 26 of CTPC and bits 31 to 8 of CTPSW are reserved for future function expansion (fixed to 0).



(6) Exception/debug trap status saving registers (DBPC and DBPSW)

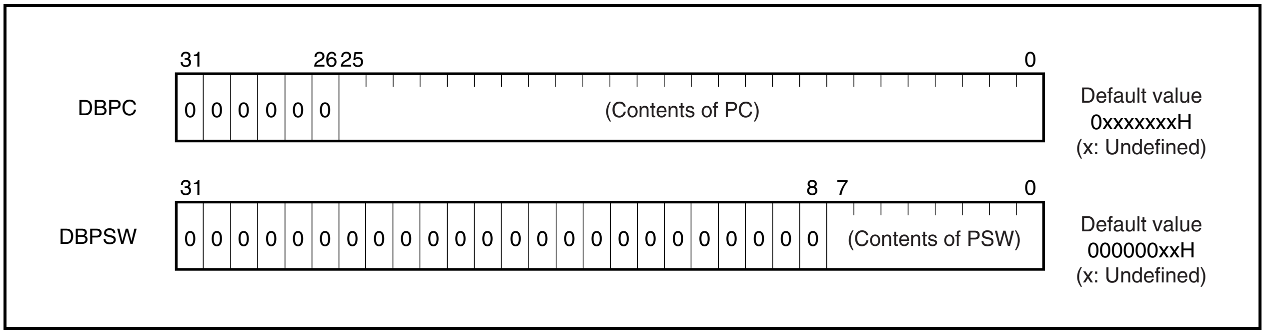
DBPC and DBPSW are exception/debug trap status saving registers.

If an exception trap or debug trap occurs, the contents of the program counter (PC) are saved to DBPC, and those of the program status word (PSW) are saved to DBPSW.

The contents to be saved to DBPC are the address of the instruction next to the one that is executed when an exception trap or debug trap occurs.

The current contents of the PSW are saved to DBPSW.

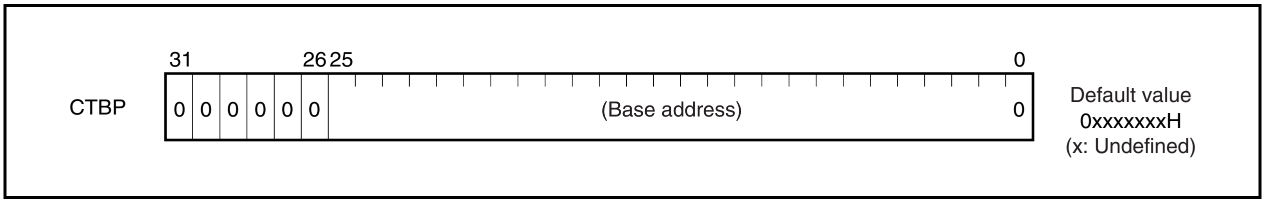
Bits 31 to 26 of DBPC and bits 31 to 8 of DBPSW are reserved for future function expansion (fixed to 0).



(7) CALLT base pointer (CTBP)

The CALLT base pointer (CTBP) is used to specify a table address or generate a target address (bit 0 is fixed to 0).

Bits 31 to 26 of this register are reserved for future function expansion (fixed to 0).



3.3 Operation Modes

3.3.1 Operation modes

The V850ES/PM1 has the following operation modes.

(1) Single-chip mode

In this mode, each pin related to the bus interface is set to the port mode after system reset has been released. Execution branches to the reset entry address of the internal ROM, and then instruction processing is started.

(2) ROMless mode

Of the pins related to the bus interface, only the PCM1 pin is set in the port mode and the other pins are set in the control mode after the system reset signal has been released. The program branches to the reset entry address of an external device (memory) and starts instruction processing. Instruction fetch or data access to the internal ROM is impossible.

The default value of the following registers differs depending on the mode.

Operation Mode	PMCDL	PMCDH	PMCCS	PMCCT	PMCCM
ROMless mode	FFFFH	07H	07H	13H	01H
Single-chip mode	0000H	00H	00H	00H	00H

3.3.2 Specifying operation mode

The operation mode is specified according to the status of the MODE pin. Fix the level of this pin in the application system and do not change it during operation, otherwise the operation will not be guaranteed.

MODE	Operation Mode	Remark
L	ROMless mode	16-bit data bus
H	Single-chip mode	–

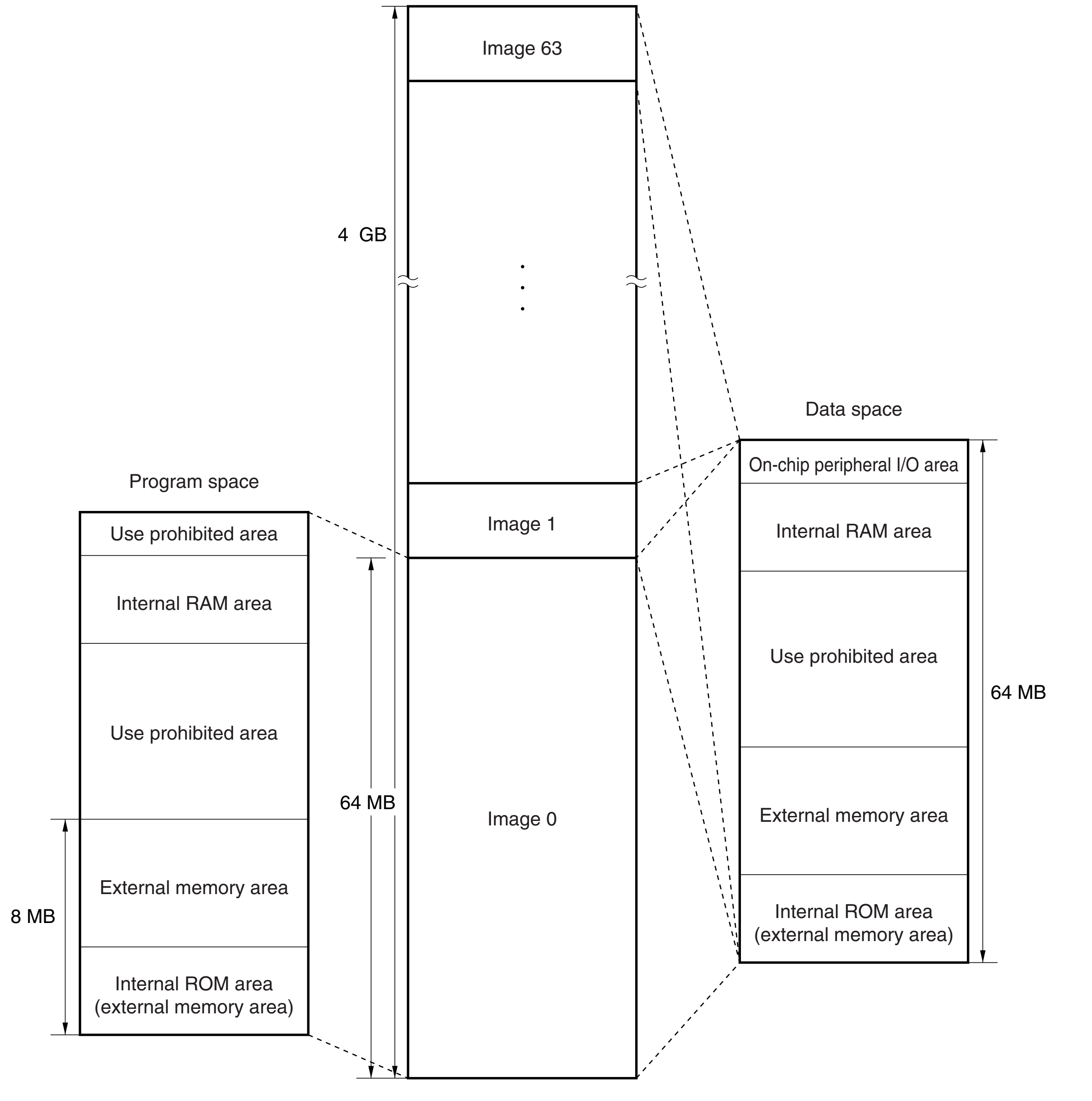
Remark L: Low-level input
H: High-level input

3.4 Address Space

3.4.1 CPU address space

For instruction addressing, up to 8 MB of linear address space (program space) and an internal RAM area are supported. Up to 4 GB of linear address space (data space) is supported for operand addressing (data access). In the 4 GB address space, it seems that there are sixty-four 64 MB physical address spaces. This means that the same 64 MB physical address space is accessed, regardless of the values of bits 31 to 26.

Figure 3-1. Image on Address Space



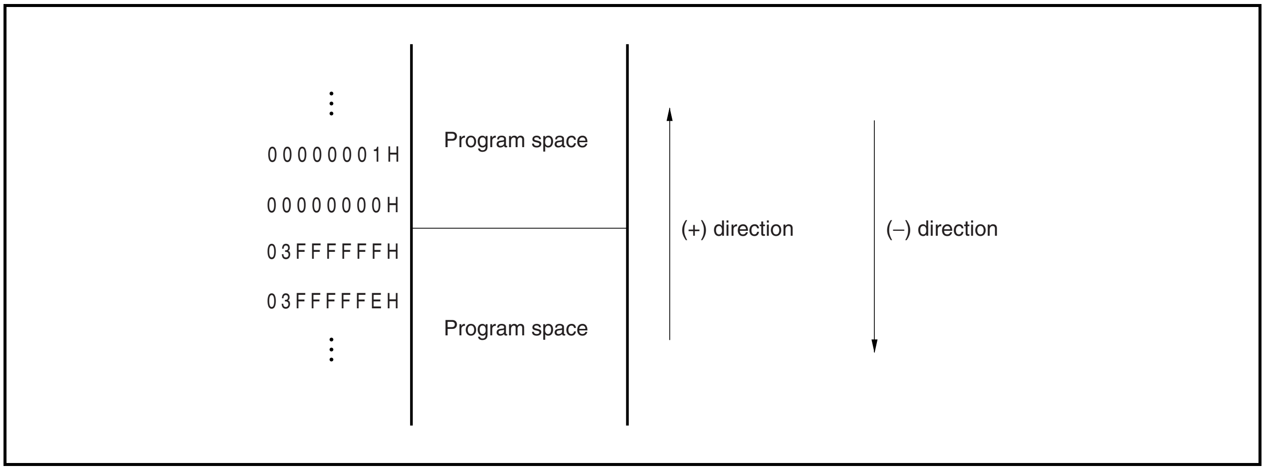
3.4.2 Wrap-around of CPU address space

(1) Program space

Of the 32 bits of the PC (program counter), the higher 6 bits are fixed to 0 and only the lower 26 bits are valid. The higher 6 bits ignore a carry or borrow from bit 25 to 26 during branch address calculation.

Therefore, the lowest address of the program space, 00000000H, and the highest address, 03FFFFFFH, are contiguous addresses. That the lowest address and the highest address of the program space are contiguous in this way is called wrap-around.

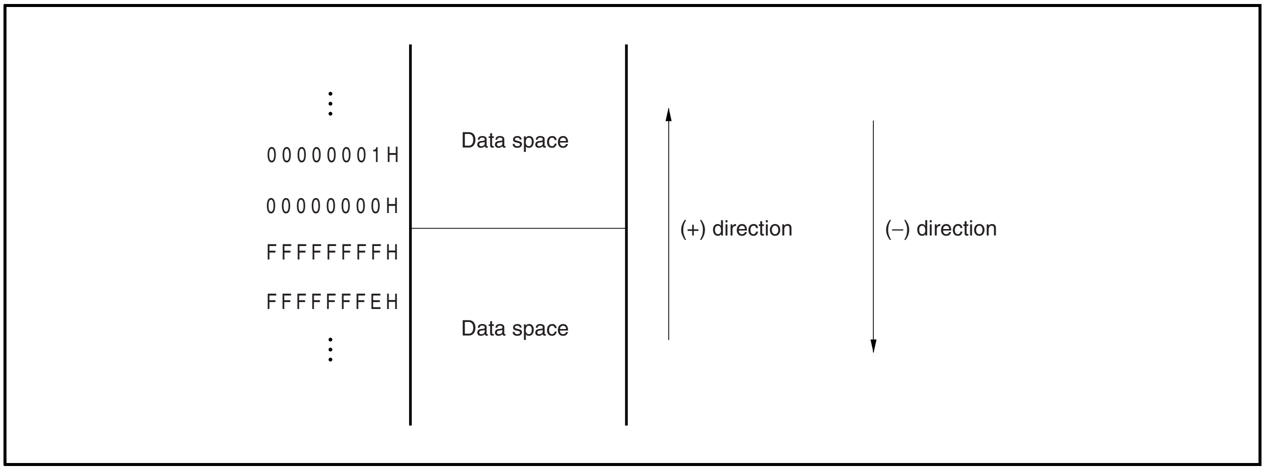
Caution Because the 4 KB area of addresses 03FFF000H to 03FFFFFFH is an on-chip peripheral I/O area, instructions cannot be fetched from this area. Therefore, do not execute an operation in which the result of a branch address calculation affects this area.



(2) Data space

The result of an operand address calculation operation that exceeds 32 bits is ignored.

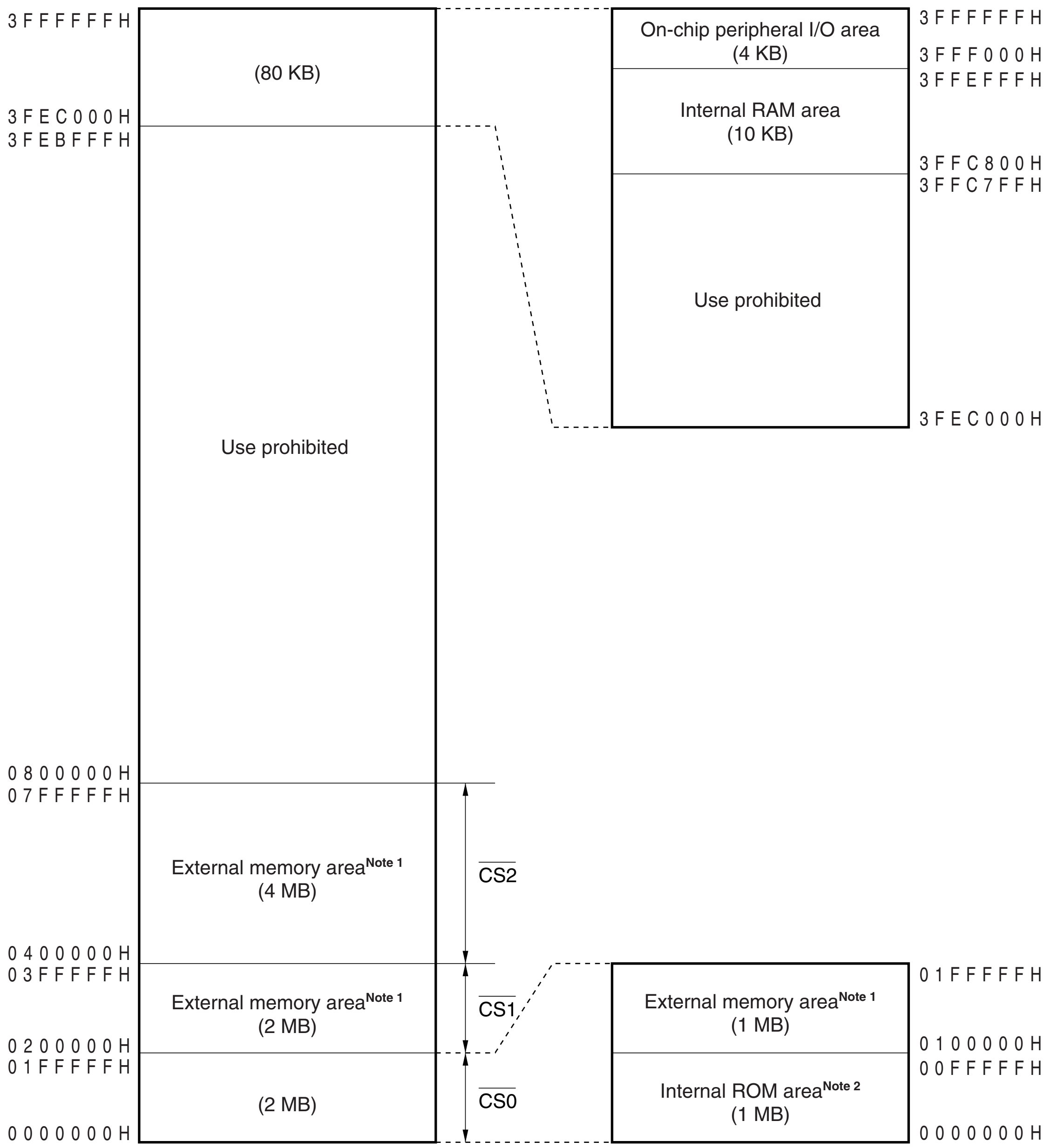
Therefore, the lowest address of the data space, 00000000H, and the highest address, FFFFFFFFH, are contiguous, and wrap-around occurs at the boundary of these addresses.



3.4.3 Memory map

The V850ES/PM1 reserves the areas shown in the following.

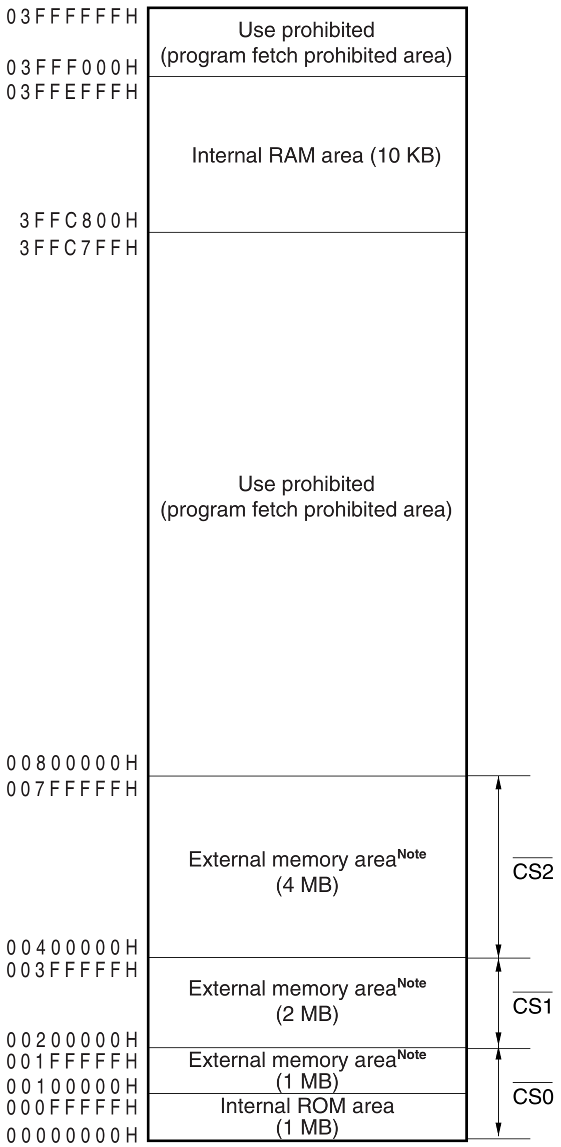
Figure 3-2. Data Memory Map (Physical Addresses)



Notes 1. Each of these areas is a 512 KB space of 0100000H to 017FFFFH, 0200000H to 027FFFFH, or 0400000H to 047FFFFH (0180000H to 01FFFFFH, 0280000H to 03FFFFFH, and 0480000H to 07FFFFFH are an image).

2. Fetch access and read access to addresses 0000000H to 00FFFFFH is made to the internal ROM area. However, data write access to these addresses is made to the external memory area.

Figure 3-3. Program Memory Map



Note Each of these areas is a 512 KB space of 0100000H to 017FFFFH, 0200000H to 027FFFFH, or 0400000H to 047FFFFH (0180000H to 01FFFFFH, 0280000H to 03FFFFFH, and 0480000H to 07FFFFFH are an image).

3.4.4 Areas

(1) Internal ROM area

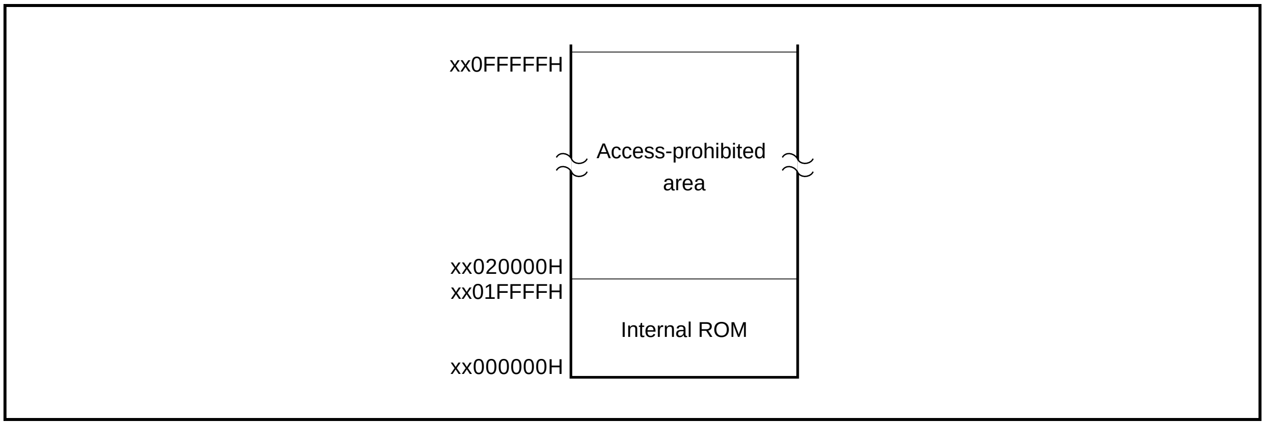
(a) Memory map

1 MB of addresses 0000000H to 00FFFFFH is reserved as the internal ROM area.

128 KB are mapped to addresses 000000H to 01FFFFH as the physical internal ROM (mask ROM).

Remark The internal ROM area is not available in the ROMless mode.

Figure 3-4. Internal ROM Area

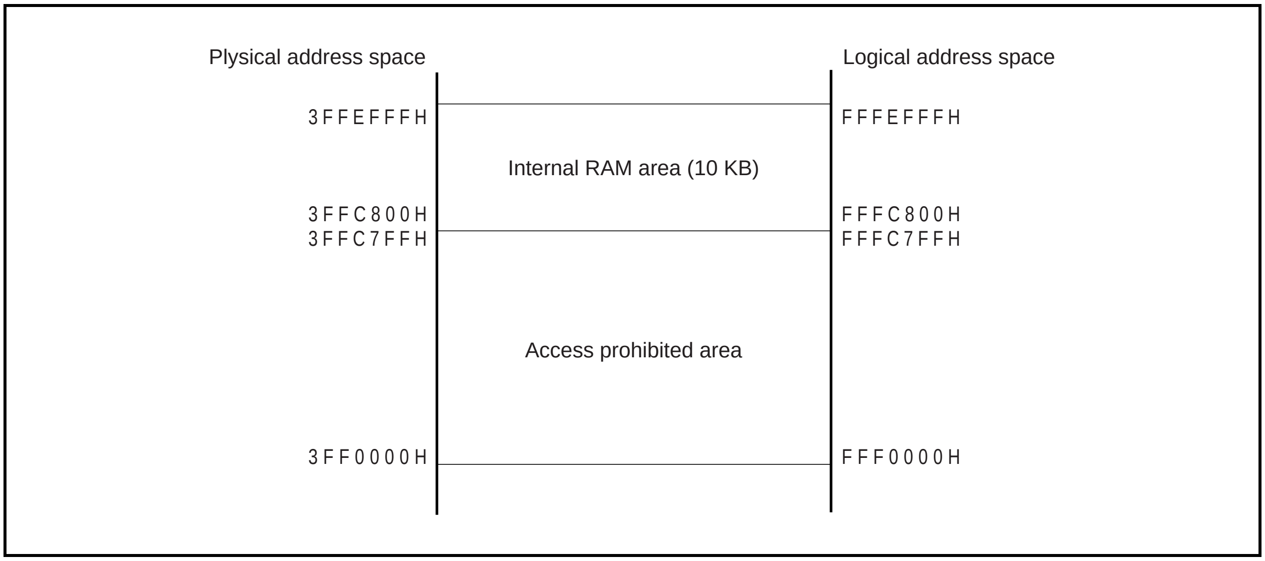


(2) Internal RAM area

60 KB of addresses 3FF0000H to 3FFEFFFH are reserved as the internal RAM area.

The V850ES/PM1 maps 10 KB of addresses 3FFC800H to 3FFEFFFH as physical internal RAM.

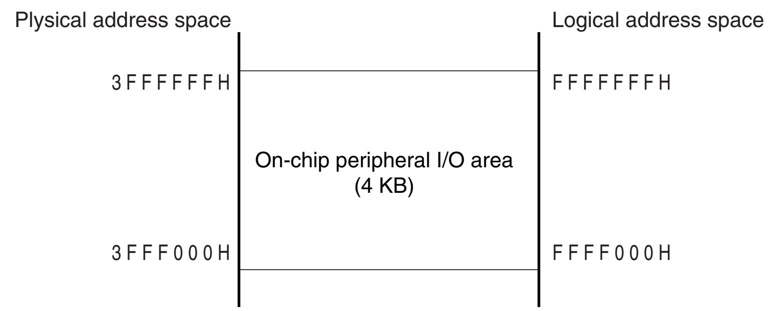
Figure 3-5. Internal RAM Area (10 KB)



(3) On-chip peripheral I/O area

4 KB of addresses 3FFF000H to 3FFFFFFH are allocated as the on-chip peripheral I/O area.

Figure 3-6. On-Chip Peripheral I/O Area



Peripheral I/O registers that have functions to specify the operation mode for and monitor the status of the on-chip peripheral I/O are mapped to the on-chip peripheral I/O area. Program cannot be fetched from this area.

Cautions 1. When a register is accessed in word units, a word area is accessed twice in halfword units in the order of lower area and higher area, with the lower 2 bits of the address ignored.

2. If a register that can be accessed in byte units is accessed in halfword units, the higher 8 bits are undefined when the register is read, and data is written to the lower 8 bits.

3. Addresses not defined as registers are reserved for future expansion. The operation is undefined and not guaranteed when these addresses are accessed.

(4) External memory area

7 MB (0100000H to 07FFFFFH) are allocated as the external memory area. For details, refer to **CHAPTER 5 BUS CONTROL FUNCTION**.

3.4.5 Recommended use of address space

The architecture of the V850ES/PM1 requires that a register that serves as a pointer be secured for address generation when operand data in the data space is accessed. The address stored in this pointer ±32 KB can be directly accessed by an instruction for operand data. Because the number of general-purpose registers that can be used as a pointer is limited, however, by keeping the performance from dropping during address calculation when a pointer value is changed, as many general-purpose registers as possible can be secured for variables, and the program size can be reduced.

(1) Program space

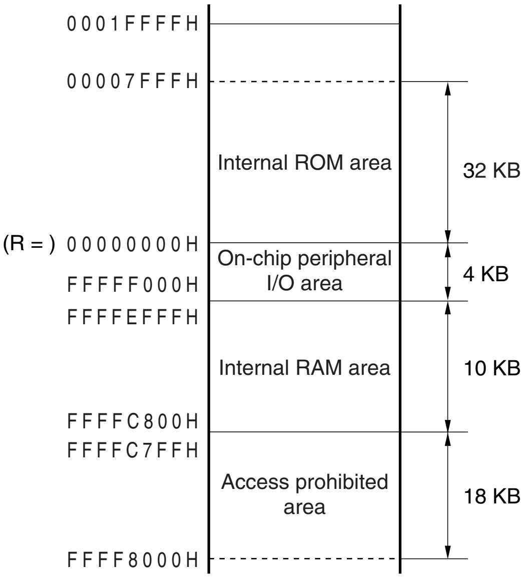
Of the 32 bits of the program counter (PC), the higher 6 bits are fixed to 0, and only the lower 26 bits are valid. Regarding the program space, therefore, a 64 MB space of contiguous addresses starting from 00000000H unconditionally corresponds to the memory map.

To use the internal RAM area as the program space, access addresses 3FFC800H to 3FFEFFFH.

(2) Data space

With the V850ES/PM1, it seems that there are sixty-four 64 MB address spaces on the 4 GB CPU address space. Therefore, the least significant bit (bit 25) of a 26-bit address is sign-extended to 32 bits and allocated as an address.

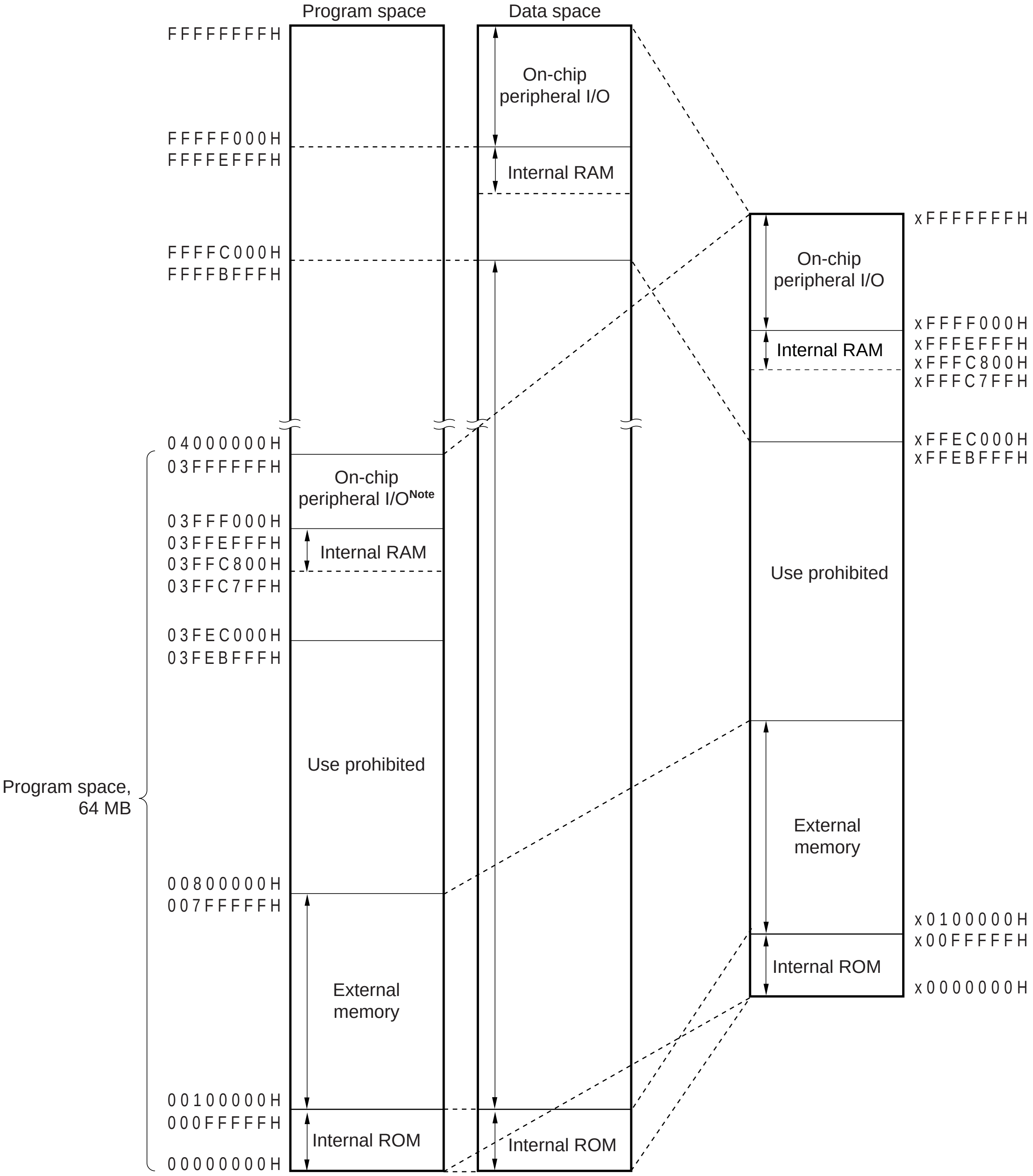
Example: An application example of wrap-around is shown below.



If R = r0 (zero register) is specified for the LD/ST disp16 [R] instruction, a range of addresses 00000000H ±32 KB can be addressed by sign-extended disp16. All the resources of the internal hardware can be addressed by one pointer.

The zero register (r0) is a register fixed to 0 by hardware, and practically eliminates the need for registers dedicated to pointers.

Figure 3-7. Recommended Memory Map



Note Access to this area is prohibited. To access the on-chip peripheral I/O in this area, specify addresses FFFF000H to FFFFFFFH.

Remark ↕ indicates the recommended area.

3.4.6 Peripheral I/O registers

(1/6)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF004H	Port DL	PDL	R/W			√	0000H[Note]
FFFFF004H	Port DLL	PDLL		√	√		00H[Note]
FFFFF005H	Port DLH	PDLH		√	√		
FFFFF006H	Port DH	PDH		√	√		
FFFFF008H	Port CS	PCS		√	√		
FFFFF00AH	Port CT	PCT		√	√		
FFFFF00CH	Port CM	PCM		√	√		
FFFFF024H	Port DL mode register	PMDL				√	FFFFH
FFFFF024H	Port DLL mode register	PMDLL		√	√		FFH
FFFFF025H	Port DLH mode register	PMDLH		√	√		
FFFFF026H	Port DH mode register	PMDH		√	√		
FFFFF028H	Port CS mode register	PMCS		√	√		
FFFFF02AH	Port CT mode register	PMCT		√	√		
FFFFF02CH	Port CM mode register	PMCM		√	√		
FFFFF044H	Port DL mode control register	PMCDL				√	0000H
FFFFF044H	Port DLL mode control register	PMCDLL		√	√		00H
FFFFF045H	Port DLH mode control register	PMCDLH		√	√		
FFFFF046H	Port DH mode control register	PMCDH		√	√		
FFFFF048H	Port CS mode control register	PMCCS		√	√		
FFFFF04AH	Port CT mode control register	PMCCT		√	√		
FFFFF04CH	Port CM mode control register	PMCCM		√	√		
FFFFF066H	Bus size configuration register	BSC				√	5555H
FFFFF06EH	System wait control register	VSWC			√		77H
FFFFF100H	Interrupt mask register 0	IMR0				√	FFFFH
FFFFF100H	Interrupt mask register 0L	IMR0L		√	√		FFH
FFFFF101H	Interrupt mask register 0H	IMR0H		√	√		
FFFFF102H	Interrupt mask register 1	IMR1				√	FFFFH
FFFFF102H	Interrupt mask register 1L	IMR1L		√	√		FFH
FFFFF103H	Interrupt mask register 1H	IMR1H		√	√		
FFFFF110H	Interrupt control register	WDTIC		√	√		47H
FFFFF112H	Interrupt control register	PIC0		√	√		
FFFFF114H	Interrupt control register	PIC1		√	√		
FFFFF116H	Interrupt control register	PIC2		√	√		
FFFFF118H	Interrupt control register	ADIC		√	√		
FFFFF11AH	Interrupt control register	RTCIC		√	√		
FFFFF11CH	Interrupt control register	TMIC000		√	√		
FFFFF11EH	Interrupt control register	TMIC001		√	√		
FFFFF120H	Interrupt control register	TMIC010		√	√		
FFFFF122H	Interrupt control register	TMIC011		√	√		
FFFFF124H	Interrupt control register	TMIC020		√	√		
FFFFF126H	Interrupt control register	TMIC021		√	√		

Note The value of the output latch is 00H or 0000H. When input, the status of the pin is read.

(2/6)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF128H	Interrupt control register	TMIC030	R/W	√	√		47H
FFFFF12AH	Interrupt control register	TMIC031		√	√		
FFFFF12CH	Interrupt control register	CCIC100		√	√		
FFFFF12EH	Interrupt control register	CCIC101		√	√		
FFFFF130H	Interrupt control register	OVFIC10		√	√		
FFFFF132H	Interrupt control register	CCIC110		√	√		
FFFFF134H	Interrupt control register	CCIC111		√	√		
FFFFF136H	Interrupt control register	OVFIC11		√	√		
FFFFF138H	Interrupt control register	TMIC20		√	√		
FFFFF13AH	Interrupt control register	TMIC21		√	√		
FFFFF13CH	Interrupt control register	CSIIC0		√	√		
FFFFF13EH	Interrupt control register	CSIIC1		√	√		
FFFFF140H	Interrupt control register	SREIC0		√	√		
FFFFF142H	Interrupt control register	SRIC0		√	√		
FFFFF144H	Interrupt control register	STIC0		√	√		
FFFFF146H	Interrupt control register	SREIC1		√	√		
FFFFF148H	Interrupt control register	SRIC1		√	√		
FFFFF14AH	Interrupt control register	STIC1		√	√		
FFFFF14CH	Interrupt control register	ROVIC		√	√		
FFFFF1FAH	In-service priority register	ISPR	R	√	√		00H
FFFFF1FCH	Command register	PRCMD	W		√		Undefined
FFFFF1FEH	Power save control register	PSC	R/W	√	√		00H
FFFFF200H	A/D converter mode register	ADM		√	√		
FFFFF201H	A/D clock delay setting register	ADLY			√		
FFFFF202H	High-pass filter control register 0	HPFC0		√	√		
FFFFF204H	A/D conversion result register 0	ADCR0	R			√	0000H
FFFFF206H	A/D conversion result register 1	ADCR1				√	
FFFFF208H	A/D conversion result register 2	ADCR2				√	
FFFFF20AH	A/D conversion result register 3	ADCR3				√	
FFFFF20CH	A/D conversion result register 4	ADCR4				√	
FFFFF20EH	A/D conversion result register 5	ADCR5				√	
FFFFF400H	Port 0	P0	R/W	√	√		00H[Note]
FFFFF402H	Port 1	P1		√	√		
FFFFF404H	Port 2	P2		√	√		
FFFFF406H	Port 3	P3		√	√		
FFFFF408H	Port 4	P4		√	√		
FFFFF412H	Port 9	P9				√	0000H[Note]
FFFFF412H	Port 9L	P9L		√	√		00H[Note]
FFFFF413H	Port 9H	P9H		√	√		
FFFFF420H	Port 0 mode register	PM0		√	√		FFH
FFFFF422H	Port 1 mode register	PM1		√	√		
FFFFF424H	Port 2 mode register	PM2		√	√		

Note The value of the output latch is 00H or 0000H. When input, the status of the pin is read.

(3/6)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF426H	Port 3 mode register	PM3	R/W	√	√		FFH
FFFFF428H	Port 4 mode register	PM4		√	√		
FFFFF432H	Port 9 mode register	PM9				√	FFFFH
FFFFF432H	Port 9 mode register L	PM9L		√	√		FFH
FFFFF433H	Port 9 mode register H	PM9H		√	√		
FFFFF440H	Port 0 mode control register	PMC0		√	√		00H
FFFFF442H	Port 1 mode control register	PMC1		√	√		
FFFFF444H	Port 2 mode control register	PMC2		√	√		
FFFFF446H	Port 3 mode control register	PMC3		√	√		
FFFFF448H	Port 4 mode control register	PMC4		√	√		
FFFFF452H	Port 9 mode control register	PMC9				√	0000H
FFFFF452H	Port 9 mode control register L	PMC9L		√	√		00H
FFFFF453H	Port 9 mode control register H	PMC9H		√	√		
FFFFF460H	Port 0 function control register	PFC0		√	√		
FFFFF462H	Port 1 function control register	PFC1		√	√		
FFFFF466H	Port 3 function control register	PFC3		√	√		
FFFFF468H	Port 4 function control register	PFC4		√	√		
FFFFF472H	Port 9 function control register	PFC9				√	0000H
FFFFF472H	Port 9 function control register L	PFC9L		√	√		00H
FFFFF473H	Port 9 function control register H	PFC9H		√	√		
FFFFF484H	Data wait control register 0	DWC0				√	7777H
FFFFF488H	Address wait control register	AWC				√	FFFFH
FFFFF48AH	Bus cycle control register	BCC				√	AAAAH
FFFFF5C0H	16-bit timer counter 00	TM00	R			√	0000H
FFFFF5C2H	16-bit timer capture/compare register 000	CR000	R/W			√	
FFFFF5C4H	16-bit timer capture/compare register 001	CR001				√	
FFFFF5C6H	16-bit timer mode control register 00	TMC00		√	√		00H
FFFFF5C7H	Prescaler mode register 00	PRM00		√	√		
FFFFF5C8H	Capture/compare control register 00	CRC00		√	√		
FFFFF5C9H	16-bit timer output control register 00	TOC00		√	√		
FFFFF5D0H	16-bit timer counter 01	TM01	R			√	0000H
FFFFF5D2H	16-bit timer capture/compare register 010	CR010	R/W			√	
FFFFF5D4H	16-bit timer capture/compare register 011	CR011				√	
FFFFF5D6H	16-bit timer mode control register 01	TMC01		√	√		00H
FFFFF5D7H	Prescaler mode register 01	PRM01		√	√		
FFFFF5D8H	Capture/compare control register 01	CRC01		√	√		
FFFFF5D9H	16-bit timer output control register 01	TOC01		√	√		
FFFFF5E0H	16-bit timer counter 02	TM02	R			√	0000H
FFFFF5E2H	16-bit timer capture/compare register 020	CR020	R/W			√	
FFFFF5E4H	16-bit timer capture/compare register 021	CR021				√	
FFFFF5E6H	16-bit timer mode control register 02	TMC02		√	√		00H
FFFFF5E7H	Prescaler mode register 02	PRM02		√	√		
FFFFF5E8H	Capture/compare control register 02	CRC02		√	√		

(4/6)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF5E9H	16-bit timer output control register 02	TOC02	R/W	√	√		00H
FFFFF5F0H	16-bit timer counter 03	TM03	R			√	0000H
FFFFF5F2H	16-bit timer capture/compare register 030	CR030	R/W			√	
FFFFF5F4H	16-bit timer capture/compare register 031	CR031				√	
FFFFF5F6H	16-bit timer mode control register 03	TMC03		√	√		00H
FFFFF5F7H	Prescaler mode register 03	PRM03		√	√		
FFFFF5F8H	Capture/compare control register 03	CRC03		√	√		
FFFFF5F9H	16-bit timer output control register 03	TOC03		√	√		
FFFFF600H	16-bit timer counter 10	TM10	R			√	0000H
FFFFF602H	16-bit timer capture/compare register 100	CC100	R/W			√	
FFFFF604H	16-bit timer capture/compare register 101	CC101				√	
FFFFF606H	16-bit timer mode control register 100	TMC100		√	√		00H
FFFFF608H	16-bit timer mode control register 101	TMC101		√	√		20H
FFFFF609H	Valid edge select register 10	SES10			√		00H
FFFFF610H	16-bit timer counter 11	TM11	R			√	0000H
FFFFF612H	16-bit timer capture/compare register 110	CC110	R/W			√	
FFFFF614H	16-bit timer capture/compare register 111	CC111				√	
FFFFF616H	16-bit timer mode control register 110	TMC110		√	√		00H
FFFFF618H	16-bit timer mode control register 111	TMC111		√	√		20H
FFFFF619H	Valid edge select register 11	SES11			√		00H
FFFFF640H	16-bit timer counter 2	TM2	R			√	0000H
FFFFF640H	8-bit timer counter 20	TM20			√		00H
FFFFF641H	8-bit timer counter 21	TM21			√		
FFFFF642H	16-bit timer compare register 2	CR2	R/W			√	0000H
FFFFF642H	8-bit timer compare register 20	CR20			√		00H
FFFFF643H	8-bit timer compare register 21	CR21			√		
FFFFF644H	Timer clock select register 2	TCL2				√	0000H
FFFFF644H	Timer clock select register 20	TCL20			√		00H
FFFFF645H	Timer clock select register 21	TCL21			√		
FFFFF646H	16-bit timer mode control register 2	TMC2				√	0000H
FFFFF646H	8-bit timer mode control register 20	TMC20		√	√		00H
FFFFF647H	8-bit timer mode control register 21	TMC21		√	√		
FFFFF680H	RTC control register	RTCC				√	8X80H
FFFFF680H	RTC control register 0	RTCC0		√	√		80H
FFFFF681H	RTC control register 1	RTCC1		√	√		8XH
FFFFF682H	Sub-count register	SUBC	R			√	Undefined
FFFFF682H	Sub-count register L	SUBCL			√		
FFFFF683H	Sub-count register H	SUBCH			√		
FFFFF684H	Minute/second count register	SECMIN				√	
FFFFF684H	Second count register	SEC			√		
FFFFF685H	Minute count register	MIN			√		

(5/6)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits				Default Value
				1	8	16	32	
FFFFF686H	Day/hour count register	HOURDAY	R			√		Undefined
FFFFF686H	Hour count register	HOUR			√			
FFFFF687H	Day count register	DAY			√			
FFFFF688H	Week count register	WEEK				√		
FFFFF688H	Week count register L	WEEKL			√			
FFFFF689H	Week count register H	WEEKH			√			
FFFFF68AH	Minute/second count setting register	SECMINB	W			√		0000H
FFFFF68AH	Second count setting register	SECB			√			00H
FFFFF68BH	Minute count setting register	MINB			√			
FFFFF68CH	Day/hour count setting register	HOURDAYB				√		0000H
FFFFF68CH	Hour count setting register	HOURB			√			00H
FFFFF68DH	Day count setting register	DAYB			√			
FFFFF68EH	Week count setting register	WEEKB				√		0000H
FFFFF68EH	Week count setting register L	WEEKBL			√			00H
FFFFF68FH	Week count setting register H	WEEKBH			√			
FFFFF6C0H	Oscillation stabilization time select register	OSTS	R/W		√			04H
FFFFF6C1H	Watchdog timer clock select register	WDCS			√			00H
FFFFF6C2H	Watchdog timer mode register	WDTM		√	√			
FFFFF802H	System status register	SYS		√	√			
FFFFF820H	Power save mode register	PSMR		√	√			
FFFFF828H	Processor clock control register	PCC		√	√			03H
FFFFF82AH	WDT reset status register	WDRES						Undefined
FFFFF840H	Correction address register 0	CORAD0	R/W				√	00000000H
FFFFF840H	Correction address register 0L	CORAD0L				√		0000H
FFFFF842H	Correction address register 0H	CORAD0H				√		
FFFFF844H	Correction address register 1	CORAD1					√	00000000H
FFFFF844H	Correction address register 1L	CORAD1L				√		0000H
FFFFF846H	Correction address register 1H	CORAD1H				√		
FFFFF848H	Correction address register 2	CORAD2					√	00000000H
FFFFF848H	Correction address register 2L	CORAD2L				√		0000H
FFFFF84AH	Correction address register 2H	CORAD2H				√		
FFFFF84CH	Correction address register 3	CORAD3					√	00000000H
FFFFF84CH	Correction address register 3L	CORAD3L				√		0000H
FFFFF84EH	Correction address register 3H	CORAD3H				√		
FFFFF880H	Correction control register	CORCN		√	√			00H
FFFFFA00H	Asynchronous serial interface mode register 0	ASIM0		√	√			01H
FFFFFA02H	Receive buffer register 0	RXB0	R		√			FFH
FFFFFA03H	Asynchronous serial interface status register 0	ASIS0			√			00H
FFFFFA04H	Transmit buffer register 0	TXB0	R/W		√			FFH
FFFFFA05H	Asynchronous serial interface transmit status register 0[Note]	ASIF0	R	√	√			00H
FFFFFA06H	Clock select register 0	CKSR0	R/W		√			
FFFFFA07H	Baud rate generator control register 0	BRGC0			√			FFH

Note Although these registers can be manipulated in 8-bit units, it is recommended to manipulate them using a bit manipulation instruction.

(6/6)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits				Default Value
				1	8	16	32	
FFFFFA10H	Asynchronous serial interface mode register 1	ASIM1	R/W	√	√			01H
FFFFFA12H	Receive buffer register 1	RXB1	R		√			FFH
FFFFFA13H	Asynchronous serial interface status register 1	ASIS1			√			00H
FFFFFA14H	Transmit buffer register 1	TXB1	R/W		√			FFH
FFFFFA15H	Asynchronous serial interface transmit status register 1[Note]	ASIF1	R	√	√			00H
FFFFFA16H	Clock select register 1	CKSR1	R/W		√			
FFFFFA17H	Baud rate generator compare register 1	BRGC1			√			FFH
FFFFFB00H	PWM control register 0	PWMC0		√	√			40H
FFFFFB02H	PWM buffer register 0	PWMB0				√		0000H
FFFFFB10H	PWM control register 1	PWMC1		√	√			40H
FFFFFB12H	PWM buffer register 1	PWMB1				√		0000H
FFFFFB20H	PWM control register 2	PWMC2		√	√			40H
FFFFFB22H	PWM buffer register 2	PWMB2				√		0000H
FFFFFB30H	PWM control register 3	PWMC3		√	√			40H
FFFFFB32H	PWM buffer register 3	PWMB3				√		0000H
FFFFFC00H	External interrupt falling edge specification register 0	INTF0		√	√			00H
FFFFFC20H	External interrupt rising edge specification register 0	INTR0		√	√			
FFFFFC40H	Pull-up resistor option register 0	PU0		√	√			
FFFFFC42H	Pull-up resistor option register 1	PU1		√	√			
FFFFFC44H	Pull-up resistor option register 2	PU2		√	√			
FFFFFC46H	Pull-up resistor option register 3	PU3		√	√			
FFFFFC48H	Pull-up resistor option register 4	PU4		√	√			
FFFFFD00H	Clocked serial interface mode register 0	CSIM0		√	√			00H
FFFFFD01H	Clocked serial interface clock select register 0	CSIC0			√			
FFFFFD02H	Serial I/O shift register 0	SIO0	R		√			
FFFFFD03H	Reception-only serial I/O shift register 0	SIOE0			√			
FFFFFD04H	Clocked serial interface transmit buffer register 0	SOTB0	R/W		√			
FFFFFD10H	Clocked serial interface mode register 1	CSIM1		√	√			
FFFFFD11H	Clocked serial interface clock select register 1	CSIC1			√			
FFFFFD12H	Serial I/O shift register 1	SIO1	R		√			
FFFFFD13H	Reception-only serial I/O shift register 1	SIOE1			√			
FFFFFD14H	Clocked serial interface transmit buffer register 1	SOTB1	R/W		√			

Note Although these registers can be manipulated in 8-bit units, it is recommended to manipulate them using a bit manipulation instruction.

3.4.7 Special registers

Special registers are registers that are protected from being written with illegal data due to a program hang-up.
The V850ES/PM1 has the following four special registers.

- Power save control register (PSC)
- Processor clock control register (PCC)
- Watchdog timer mode register (WDTM)
- WDT reset register (WDRES)

In addition, a command register (PRCDM) is provided to protect against a write access to the special registers so that the application system does not inadvertently stop due to a program hang-up. A write access to the special registers is made in a specific sequence, and an illegal store operation is reported to the system status register (SYS).

(1) Setting data to special registers

Set data to the special registers in the following sequence:

<1> Prepare data to be set to the special register in a general-purpose register.
<2> Write the data prepared in <1> to the PRCMD register.
<3> Write the setting data to the special register (by using the following instructions).

- Store instruction (ST/SST instruction)
- Bit manipulation instruction (SET1/CLR1/NOT1 instruction)

[Example] With PSC register

```
   ST.B r11, PSMR[r0]   ; Set PSMR register.
<1>MOV  0x02, r10
<2>ST.B r10, PRCMD[r0]  ; Write PRCMD register.
<3>ST.B r10, PSC[r0]    ; Set PSC register.
<4>NOP                  ; Dummy instruction
<5>NOP                  ; Dummy instruction
<6>NOP                  ; Dummy instruction
<7>NOP                  ; Dummy instruction
<8>NOP                  ; Dummy instruction
(next instruction)
```

There is no special sequence to read a special register.

Cautions 1. When a store instruction is executed to store data in the command register, an interrupt is not acknowledged. This is because it is assumed that steps <2> and <3> above are performed by successive store instructions. If another instruction is placed between <2> and <3>, and if an interrupt is acknowledged by that instruction, the above sequence may not be established, causing malfunction.

2. Although dummy data is written to the PRCMD register, use the same general-purpose register used to set the special register (<3> in Example) to write data to the PRCMD register (<2> in Example). The same applies when a general-purpose register is used for addressing.

3. Five NOP instructions or more must be inserted immediately after setting the IDLE mode or software STOP mode (by setting the PSC.STP bit to 1).

(2) Command register (PRCMD)

The PRCMD register is an 8-bit register that protects the registers that may seriously affect the application system from being written, so that the system does not inadvertently stop due to a program hang-up. The first write access to a special register (the PCC, PSC, WDRES, or WDTM register) is valid after data has been written in advance to the PRCMD register. In this way, the value of the special register can be rewritten only in a specific sequence, so as to protect the register from an illegal write access.

The PRCMD register is write-only, in 8-bit units (undefined data is read when this register is read).

The values are undefined after reset.

After reset: Undefined W Address: FFFFF1FCH

	7	6	5	4	3	2	1	0
PRCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0

(3) System status register (SYS)

Status flags that indicate the operation status of the overall system are allocated to the SYS register.

This register can be read or written in 8-bit or 1-bit units.

This register is set to 00H after reset.

After reset: 00H R/W Address: FFFFF802H

	7	6	5	4	3	2	1	<0>
SYS	0	0	0	0	0	0	0	PRERR

PRERR	Detects protection error
0	Protection error did not occur.
1	Protection error occurred.

The PRERR flag operates under the following conditions.

(a) Set condition (PRERR flag = 1)

(i) When data is written to a special register without writing anything to the PRCMD register (when <3> is executed without executing <2> in **3.4.7 (1) Setting data to special registers**)

(ii) When data is written to a peripheral I/O register other than a special register (including execution of a bit manipulation instruction) after writing data to the PRCMD register (if <3> in **3.4.7 (1) Setting data to special registers** is not the setting of a special register)

Remark Even if a peripheral I/O register is read (excluding execution of a bit manipulation instruction) between a write access to the PRCMD register and a write access to a special register other than the WDTM register (PCC, PSC, and WDRES registers) (such as an access to the internal RAM), the PRERR flag is not set and data can be written to the special register.

(b) Clear condition (PRERR flag = 0)

(i) When 0 is written to the SYS.PRERR flag

(ii) When the system is reset

Cautions 1. If 0 is written to the PRERR bit of the SYS register, which is not a special register, immediately after a write access to the PRCMD register, the PRERR bit is cleared to 0 (the write access takes precedence).

2. If data is written to the PRCMD register, which is not a special register, immediately after a write access to the PRCMD register, the PRERR bit is set to 1.

3.4.8 Cautions

(1) System wait control register (VSWC)

Be sure to set the VSWC register first when using the V850ES/PM1.

After setting the VSWC register, set the other registers as necessary.

When using the external bus, perform the following initial settings after setting the above register.

- Set each pin to the control mode by using the port-related registers.

The VSWC register controls wait of bus access to the on-chip peripheral I/O registers.

Three clocks are required to access an on-chip peripheral I/O register (without a wait cycle). The V850ES/PM1 requires wait cycles according to the operating frequency. Set the following value to the VSWC register in accordance with the frequency used.

The VSWC register can be read or written in 8-bit units (address: FFFFF06EH, default value: 77H).

Operating Frequency (f_{CLK})	Set Value of VSWC
2 MHz ≤ f_{CLK} ≤ 10 MHz	00H
10 MHz < f_{CLK} ≤ 20 MHz	02H

★ **(2) Access to special on-chip peripheral I/O registers**

The V850ES/PM1 has two types of internal system buses.

One is a CPU bus and the other is a peripheral bus that interfaces with low-speed peripheral hardware.

Because the clocks for the CPU bus and for the peripheral bus are asynchronous, if a conflict between the access to the CPU and access to the peripheral hardware occurs, unexpected invalid data may be communicated. Therefore, during an access to the peripheral hardware that may cause a conflict, the number of access cycles for the CPU is changed so that data can be communicated correctly. As a result, the CPU does not shift to the next instruction processing and the CPU processing is in the wait status. The number of clocks for instruction execution is therefore longer by the wait clocks shown below if this wait is generated.

Note this caution when real-time processing is required.

During access to specific on-chip peripheral I/O registers, there are cases in which waits other than those set in the VSWC register are required.

The following shows the access method and how to calculate the number waits to be inserted (number of CPU clocks) in such cases.

Peripheral Function	Register Name	Access Method	k
Watchdog timer (WDT)	WDTM	Write	1 to 17
	<Equation to calculate maximum of number waits> $\{(16/f_{XX} \times 2/((2 + m)/f_{CPU})\} + 1$ f_{XX}: Main clock frequency		
16-bit timer/event counters 00 to 03 (TM00 to TM03)	TMC00 to TMC03	Read, modify, write	1 (fixed) A wait is generated during write
16-bit timer/event counters (TM10, TM11)	TM10, TM11	Read	1 and 2
	CC100, CC101 CC110, CC111	Read (in capture mode) Write (in compare mode)	1 and 2
	TMC100, TMC110	Write	1 and 2
	<Equation to calculate maximum of number waits> $\{(1/f_{XX} \times 2/((2 + m)/f_{CPU})\} + 1$ In case the TMC1n0.TM1CAEn bit is set to 1 f_{XX}: Main clock frequency		
	TMC100, TMC110	Read, modify, write	1 to 3
	<Equation to calculate maximum of number waits> $\{(1/f_{XX} \times 2/((2 + m)/f_{CPU})\} + 1$ f_{XX}: Main clock frequency		
PWM	PWMB0 to PWMB3	Write	3 to 35
	<Equation to calculate maximum of number waits> $\{(1/f_{PWM}/((2 + m)/f_{CPU})\} + 1$ While PWMCn is operating f_{PWM}: Clock frequency specified for PWM (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$) For details, refer to **13.3 (1) PWM control register n (PWMCn)**.		
Asynchronous serial interface (UART0, UART1)	ASIS0, ASIS1	Read	1 (fixed)

Number of clocks increased by wait = (2 + m) × k [clocks] (k: Maximum number of waits)

Caution While the CPU is operating on the subclock and when a clock is not input to X1 or the main oscillator is stopped, do not access the registers that cause a wait (excluding the TMC00 to TMC03, ASIS0 and ASIS1 registers) using an access method that causes a wait. If a wait is generated, only a reset can release the wait.

Remark In the equation to calculate waits, the following applies.

f_{CPU}:	CPU clock frequency
m:	Set values of bits 2 to 0 of the VSWC register
f_{CLK}:	Internal system clock

When $f_{CLK} \leq 10.0$ MHz: m = 0
When $f_{CLK} > 10.0$ MHz: m = 2

If the product of the decimal places of the solution multiplied by ($1/f_{CPU}$) is equal to ($1/f_{CPU}$)/(2 + m) or less, round off the decimal places, and if the product is more than ($1/f_{CPU}$)/(2 + m), round up the decimal places.

CHAPTER 4 PORT FUNCTIONS

4.1 Features

○ I/O ports: 68 pins
○ I/O pins function alternately as other peripheral functions
○ Can be set to input or output mode in 1-bit units.

4.2 Basic Configuration of Port

The V850ES/PM1 has a total of 68 input/output port pins: ports 0 to 4, 9, CM, CS, CT, DH, and DL. The port configuration is shown below.

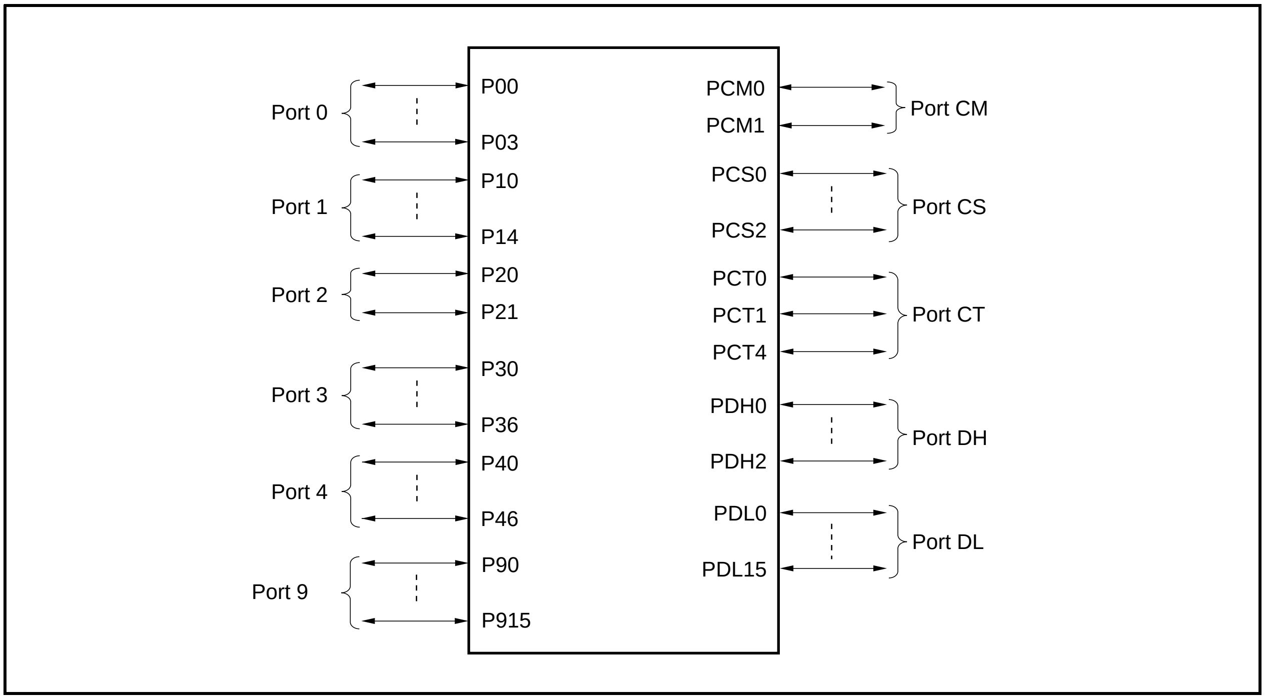


Table 4-1. I/O Buffer Power Supply for Each Pin

Power Supply	Corresponding Pin
AV_{DD}	ANIn0, ANIn1 (n = 0 to 5)
EV_{DD}	Ports 0 to 4, 9, CM, CS, CT, DH, DL, $\overline{\text{RESET}}$

4.3 Port Configuration

Table 4-2. Port Configuration

Item	Configuration
Control registers	Port n register (Pn: n = 0 to 4, 9, CM, CS, CT, DH, DL) Port n mode register (PMn: n = 0 to 4, 9, CM, CS, CT, DH, DL) Port n mode control register (PMCn: n = 0 to 4, 9, CM, CS, CT, DH, DL) Port n function control register (PFCn: n = 0, 1, 3, 4, 9) Pull-up resistor option register n (PUn: n = 0 to 4)
Ports	I/O: 68 pins
Pull-up resistor	Software-controlled: 25 resistors

(1) Port n register (Pn)

Data I/O with external devices is performed by writing to and reading from the Pn register. The Pn register is configured by a port latch that retains the output data and a circuit that reads the status of pins.

Each bit of the Pn register corresponds to one pin of port n. The Pn register can be read/written in 1-bit units.

After reset: 00H (output latch) R/W

	7	6	5	7	3	2	1	0
Pn	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	Pn0

Pnm	Control of output data (in output mode)
0	Output 0.
1	Output 1.

Writing to/reading from the Pn register is performed as follows regardless of the setting of the port n mode register (PMCn).

Table 4-3. Writing to/Reading from Port n Register (Pn)

Setting in PMn Register	Writing to Pn Register	Reading from Pn Register
Output mode (PMnm = 0)	The value is written to the output latch[Note]. In the port mode (PMCn = 0), the contents of the output latch are output from the pin.	The value in the output latch is read.
Input mode (PMnm = 1)	The value is written to the output latch. The status of the pin is not affected[Note].	The status of the pin is read.

Note The value written to the output latch is retained until another value is written to the output latch.

(2) Port n mode register (PMn)

The port n mode register specifies the input mode/output mode of the port.

Each bit of the port PMn mode register corresponds to one pin of port n. The port n mode register can be specified in 1-bit units.

After reset: FFH R/W

	7	6	5	4	3	2	1	0
PMn	PMn7	PMn6	PMn5	PMn4	PMn3	PMn2	PMn1	PMn0

PMnm	Controls I/O mode
0	Output mode
1	Input mode

(3) Port n mode control register (PMCn)

The port n mode control register specifies the port mode/alternate function.

Each bit of the port PMCn mode control register corresponds to one pin of port n. The port n mode control register can be specified in 1-bit units.

After reset: 00H R/W

	7	6	5	4	3	2	1	0
PMCn	PMCn7	PMCn6	PMCn5	PMCn4	PMCn3	PMCn2	PMCn1	PMCn0

PMCnm	Specifies of operation mode
0	Port mode
1	Alternate-function mode

(4) Port n function control register (PFCn)

The port n function control register specifies the alternate function to be used when a pin has two or more alternate functions.

Each bit of the port PFCn function control register corresponds to one pin of port n. The port n function control register can be specified in 1-bit units.

After reset: 00H R/W

	7	6	5	4	3	2	1	0
PFCn	PFCn7	PFCn6	PFCn5	PFCn4	PFCn3	PFCn2	PFCn1	PFCn0

PFCnm	Specifies of alternate function
0	Alternate function 1
1	Alternate function 2

(5) Pull-up resistor option register n (PUn)

Pull-up resistor option register n specifies the connection of an on-chip pull-up resistor.

Each bit of the PUn register corresponds to one pin of port n. Pull-up resistor option register n can be specified in 1-bit units.

After reset: 00H R/W

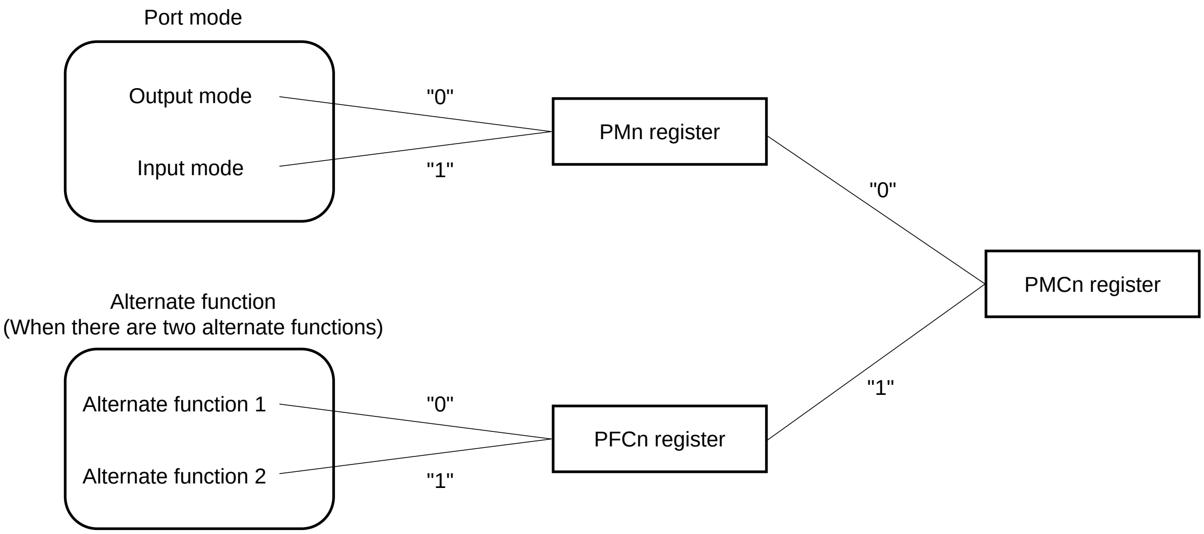
	7	6	5	4	3	2	1	0
PUn	PUn7	PUn6	PUn5	PUn4	PUn3	PUn2	PUn1	PUn0

PUnm	Controls on-chip pull-up resistor connection
0	Not connected
1	Connected

(6) Port setting

Set the ports as follows.

Figure 4-1. Setting of Each Register and Functions of Pins



Remark Follow the procedure below to switch to the alternate function.

<1> Set the PFCn register.
<2> Set the PMCn register.
<3> Set external interrupt rising/falling edge specification register n (INTRn/INTFn) (when setting the external interrupt pin).

If the PMCn register is set first, an unintended alternate function may be set while the PFCn register is being set.

4.3.1 Port 0

Port 0 is a 4-bit I/O port that can be set to the input or output mode in 1-bit units.

Port 0 has an alternate function as the following pins.

Table 4-4. Alternate-Function Pins of Port 0

Pin Name		Alternate-Function Pin	I/O	PULL[Note]	Remark	Block Type
Port 0	P00	NMI	Input	Provided	–	A-3
	P01	INTP0	Input			A-3
	P02	INTP1	Input			A-3
	P03	INTP2/TI20	Input			A-5

Note Software pull-up function

(1) Registers

(a) Port 0 register (P0)

After reset: 00H (output latch) R/W Address: FFFFF400H

	7	6	5	4	3	2	1	0
P0	0	0	0	0	P03	P02	P01	P00

P0n	Controls output data (in output mode) (n = 0 to 3)
0	Output 0.
1	Output 1.

(b) Port 0 mode register (PM0)

After reset: FFH R/W Address: FFFFF420H

	7	6	5	4	3	2	1	0
PM0	1	1	1	1	PM03	PM02	PM01	PM00

PM0n	Controls input/output mode (n = 0 to 3)
0	Output mode
1	Input mode

(c) Port 0 mode control register (PMC0)

After reset: 00H R/W Address: FFFFF440H

	7	6	5	4	3	2	1	0
PMC0	0	0	0	0	PMC03	PMC02	PMC01	PMC00

PMC03	Specifies operation mode of P03 pin
0	I/O port
1	INTP2/TI20 input

PMC02	Specifies operation mode of P02 pin
0	I/O port
1	INTP1 input

PMC01	Specifies operation mode of P01 pin
0	I/O port
1	INTP0 input

PMC00	Specifies operation mode of P00 pin
0	I/O port
1	NMI input

(d) Port 0 function control register (PFC0)

After reset: 00H R/W Address: FFFFF460H

	7	6	5	4	3	2	1	0
PFC0	0	0	0	0	PFC03	0	0	0

PFC03	Specifies operation mode of P03 pin in control mode
0	INTP2 input
1	TI20 input

Caution When using port 0 to input an external interrupt, specify the valid edge of the interrupt request by using the INTR0 and INTF0 registers. When using the port for timer input, specify the valid edge of TI20 by using the TCL20 register.

- **INTR0: External interrupt rising edge specification register 0 (refer to 16.4.2 (1))**
- **INTF0: External interrupt falling edge specification register 0 (refer to 16.4.2 (1))**
- **TCL20: Timer clock select register 20 (refer to CHAPTER 9 8-BIT TIMER/EVENT COUNTERS 20 AND 21**

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

After reset: 00H R/W Address: FFFFFC40H

	7	6	5	4	3	2	1	0
PU0	0	0	0	0	PU03	PU02	PU01	PU00

PU0n	Controls connection of on-chip pull-up resistor (n = 0 to 3)
0	Not connected
1	Connected

4.3.2 Port 1

Port 1 is a 5-bit I/O port that can be set to the input or output mode in 1-bit units.

Port 1 has an alternate function as the following pins.

Table 4-5. Alternate-Function Pins of Port 1

Pin Name		Alternate-Function Pin	I/O	PULL[Note]	Remark	Block Type
Port 1	P10	PWM0	Output	Provided	–	B-3
	P11	TO00/PWM1	Output			B-4
	P12	TO01/PWM2	Output			B-4
	P13	TO20/PWM3	Output			B-4
	P14	TO21/TI21	I/O			D-1

Note Software pull-up function

(1) Registers

(a) Port 1 register (P1)

P1 is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

After reset: 00H (output latch) R/W Address: FFFFF402H

	7	6	5	4	3	2	1	0
P1	0	0	0	P14	P13	P12	P11	P10

P1n	Controls output data (in output mode) (n = 0 to 4)
0	Output 0.
1	Output 1.

(b) Port 1 mode register (PM1)

After reset: FFH R/W Address: FFFFF422H

	7	6	5	4	3	2	1	0
PM1	1	1	1	PM14	PM13	PM12	PM11	PM10

PM1n	Controls input/output mode (n = 0 to 4)
0	Output mode
1	Input mode

(c) Port 1 mode control register (PMC1)

After reset: 00H R/W Address: FFFFF442H

	7	6	5	4	3	2	1	0
PMC1	0	0	0	PMC14	PMC13	PMC12	PMC11	PMC10

PMC14	Specifies operation mode of P14 pin
0	I/O port
1	TO21/TI21 I/O

PMC13	Specifies operation mode of P13 pin
0	I/O port
1	TO20/PWM3 output

PMC12	Specifies operation mode of P12 pin
0	I/O port
1	TO01/PWM2 output

PMC11	Specifies operation mode of P11 pin
0	I/O port
1	TO00/PWM1 output

PMC10	Specifies operation mode of P10 pin
0	I/O port
1	PWM0 output

(d) Port 1 function control register (PFC1)

After reset: 00H   R/W   Address: FFFFF442H

	7	6	5	4	3	2	1	0
PFC1	0	0	0	PFC14	PFC13	PFC12	PFC11	0

PFC14	Specifies operation mode of P14 pin in control mode
0	TO21 output
1	TI21 input

PFC13	Specifies operation mode of P13 pin in control mode
0	TO20 output
1	PWM3 output

PFC12	Specifies operation mode of P12 pin in control mode
0	TO01 output
1	PWM2 output

PFC11	Specifies operation mode of P11 pin in control mode
0	TO00 output
1	PWM1 output

(e) Pull-up resistor option register 1 (PU1)

After reset: 00H   R/W   Address: FFFFFC42H

	7	6	5	4	3	2	1	0
PU1	0	0	0	PU14	PU13	PU12	PU11	PU10

PU1n	Controls connection of on-chip pull-up resistor (n = 0 to 4)
0	Not connected
1	Connected

4.3.3 Port 2

Port 2 is a 2-bit I/O port that can be set to the input or output mode in 1-bit units.

Port 2 has an alternate function as the following pins.

Table 4-6. Alternate-Function Pins of Port 2

Pin Name		Alternate-Function Pin	I/O	PULL[Note]	Remark	Block Type
Port 2	P20	TO02	Output	Provided	–	B-3
	P21	TO03	Output			B-3

Note Software pull-up function

(1) Registers

(a) Port 2 register (P2)

After reset: 00H (output latch) R/W Address: FFFFF404H

	7	6	5	4	3	2	1	0
P2	0	0	0	0	0	0	P21	P20

P2n	Controls output data (in output mode) (n = 0, 1)
0	Output 0.
1	Output 1.

(b) Port 2 mode register (PM2)

After reset: FFH R/W Address: FFFFF424H

	7	6	5	4	3	2	1	0
PM2	1	1	1	1	1	1	PM21	PM20

PM2n	Controls input/output mode (n = 0, 1)
0	Output mode
1	Input mode

(c) Port 2 mode control register (PMC2)

After reset: 00H R/W Address: FFFFF444H

	7	6	5	4	3	2	1	0
PMC2	0	0	0	0	0	0	PMC21	PMC20

PMC21	Specifies operation mode of P21 pin
0	I/O port
1	TO03 output

PMC20	Specifies operation mode of P20 pin
0	I/O port
1	TO02 output

(d) Pull-up resistor option register 2 (PU2)

After reset: 00H R/W Address: FFFFFC44H

	7	6	5	4	3	2	1	0
PU2	0	0	0	0	0	0	PU21	PU20

PU2n	Controls connection of internal pull-up resistor (n = 0, 1)
0	Not connected
1	Connected

4.3.4 Port 3

Port 3 is a 7-bit I/O port that can be set to the input or output mode in 1-bit units.

Port 3 has an alternate function as the following pins.

Table 4-7. Alternate-Function Pins of Port 3

Pin Name		Alternate-Function Pin	I/O	PULL[Note]	Remark	Block Type
Port 3	P30	RXD0	Input	Provided	–	A-6
	P31	TXD0	Output			B-3
	P32	SI1	Input			A-2 (without noise elimination)
	P33	SO1	Output			B-3
	P34	$\overline{\text{SCK1}}$	I/O			C-2
	P35	INTP100/TI10/TCLR10	Input			A-2 (with noise elimination)
	P36	INTP110/TI11/TCLR11	Input			A-2 (with noise elimination)

Note Software pull-up function

(1) Registers

(a) Port 3 register (P3)

After reset: FFH R/W Address: FFFFF426H

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

PM3n	Controls input/output mode (n = 0 to 6)
0	Output mode
1	Input mode

(b) Port 3 mode register (PM3)

After reset: FFH R/W Address: FFFFF426H

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

PM3n	Controls input/output mode (n = 0 to 6)
0	Output mode
1	Input mode

(c) Port 3 mode control register (PMC3)

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	0	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

PMC36	Specifies operation mode of P36 pin
0	I/O port
1	INTP110/TI11/TCLR11 input

PMC35	Specifies operation mode of P35 pin
0	I/O port
1	INTP100/TI10/TCLR10 input

PMC34	Specifies operation mode of P34 pin
0	I/O port
1	$\overline{\text{SCK1}}$ I/O

PMC33	Specifies operation mode of P33 pin
0	I/O port
1	SO1 output

PMC32	Specifies operation mode of P32 pin
0	I/O port
1	SI1 input

PMC31	Specifies operation mode of P31 pin
0	I/O port
1	TXD0 output

PMC30	Specifies operation mode of P30 pin
0	I/O port
1	RXD0 input

Caution When PMC35 and PMC36 bits = 1, perform the following setting:

<1> To use INTPn0:

- **CMSn0 bit of TMCn1 register = 0, ETIn bit = 0, and ECLRn bit = 0**
- **Setting of valid edge by SESn register**

<2> To use TIn:

- **CMSn0 bit of TMCn1 register = 1, ETIn bit = 1, and ECLRn bit = 0**
- **Setting of valid edge by SESn register**

<3> To use TCLRn:

- **CMSn0 bit of TMCn1 register = 1, ETIn bit = 0, and ECLRn bit = 1**
- **Setting of valid edge by SESn register**

(d) Port 3 function control register (PFC3)

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	0	0	0	0	0	0	0	PFC30

PFC30	Specifies operation mode of P30 pin in control mode
0	RXD0 input
1	Reversed RXD0 input (Reverses the value of the RXD0 pin and supplies it to UART0.)

Caution The PFC30 bit is valid only when the PMC30 bit = 1.

(e) Pull-up resistor option register 3 (PU3)

After reset: 00H R/W Address: FFFFFC46H

	7	6	5	4	3	2	1	0
PU3	0	PU36	PU35	PU34	PU33	PU32	PU31	PU30

PU3n	Controls connection of internal pull-up resistor (n = 0 to 6)
0	Not connected.
1	Connected.

4.3.5 Port 4

Port 4 is a 7-bit I/O port that can be set to the input or output mode in 1-bit units.

Port 4 has an alternate function as the following pins.

Table 4-8. Alternate-Function Pins of Port 4

Pin Name		Alternate-Function Pin	I/O	PULL[Note]	Remark	Block Type
Port 4	P40	SI0	Input	Provided	–	A-2 (without noise elimination)
	P41	SO0	Output			B-3
	P42	$\overline{\text{SCK0}}$	I/O			C-2
	P43	RXD1	Input			A-6
	P44	TXD1	Output			B-3
	P45	INTP101/TO10	I/O			D-2
	P46	INTP111/TO11	I/O			D-2

Note Software pull-up function

(1) Registers

(a) Port 4 register (P4)

After reset: 00H (output latch) R/W Address: FFFFF408H

	7	6	5	4	3	2	1	0
P4	0	P46	P45	P44	P43	P42	P41	P40

P4n	Controls output data (in output mode) (n = 0 to 6)
0	Output 0.
1	Output 1.

(b) Port 4 mode register (PM4)

After reset: FFH R/W Address: FFFFF428H

	7	6	5	4	3	2	1	0
PM4	1	PM46	PM45	PM44	PM43	PM42	PM41	PM40

PM4n	Controls input/output mode (n = 0 to 6)
0	Output mode
1	Input mode

(c) Port 4 mode control register (PMC4)

After reset: 00H R/W Address: FFFFF448H

	7	6	5	4	3	2	1	0
PMC4	0	PMC46	PMC45	PMC44	PMC43	PMC42	PMC41	PMC40

PMC46	Specifies operation mode of P46 pin
0	I/O port
1	INTP111/TO11 I/O

PMC45	Specifies operation mode of P45 pin
0	I/O port
1	INTP101/TO10 I/O

PMC44	Specifies operation mode of P44 pin
0	I/O port
1	TXD1 output

PMC43	Specifies operation mode of P43 pin
0	I/O port
1	RXD1 input

PMC42	Specifies operation mode of P42 pin
0	I/O port
1	$\overline{\text{SCK0}}$ I/O

PMC41	Specifies operation mode of P41 pin
0	I/O port
1	SO0 output

PMC40	Specifies operation mode of P40 pin
0	I/O port
1	SI0 input

(d) Port 4 function control register (PFC4)

After reset: 00H R/W Address: FFFFF468H

	7	6	5	4	3	2	1	0
PFC4	0	PFC46	PFC45	0	PFC43	0	0	0

PFC46	Specifies operation mode of P46 pin in control mode
0	INTP111 input
1	TO11 output[Note]

PFC45	Specifies operation mode of P45 pin in control mode
0	INTP101 input
1	TO10 output[Note]

PFC43	Specifies operation mode of P43 pin in control mode
0	RXD1 input
1	Reversed RXD1 input (Reverses the value of the RXD1 pin and supplies it to UART1.)

Note Setting of the PFC45 and PFC46 bits to 1 is enabled only when TO1n output is enabled (TMC1n1.ENTO1n bit = 1)(n = 0, 1). Otherwise, this setting is prohibited.

Caution The PFC4n bit is valid only when the PMC4n bit = 1 (n = 3, 5, or 6).

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

After reset: 00H R/W Address: FFFFFC48H

	7	6	5	4	3	2	1	0
PU4	0	PU46	PU45	PU44	PU43	PU42	PU41	PU40

PU4n	Controls connection of internal pull-up resistor (n = 0 to 6)
0	Not connected.
1	Connected.

4.3.6 Port 9

Port 9 is a 16-bit I/O port that can be set to the input or output mode in 1-bit units.

Port 9 has an alternate function as the following pins.

Table 4-9. Alternate-Function Pins of Port 9

Pin Name		Alternate-Function Pin	I/O	PULL[Note]	Remark	Block Type
Port 9	P90	A0	Output	None	–	B-2
	P91	A1	Output			B-2
	P92	A2	Output			B-2
	P93	A3	Output			B-2
	P94	A4	Output			B-2
	P95	A5	Output			B-2
	P96	A6	Output			B-2
	P97	A7	Output			B-2
	P98	A8/TI030	I/O			A-4
	P99	A9/TI031	I/O			A-4
	P910	A10/TI020	I/O			A-4
	P911	A11/TI021	I/O			A-4
	P912	A12/TI010	I/O			A-4
	P913	A13/TI011	I/O			A-4
	P914	A14/TI000	I/O			A-4
	P915	A15/TI001	I/O			A-4

Note Software pull-up function

(1) Registers

(a) Port 9 register (P9)

After reset: 0000H (output latch) R/W Address: FFFFF412H, FFFFF413H

	15	14	13	12	11	10	9	8
P9	P915	P914	P913	P912	P911	P910	P99	P98

7	6	5	4	3	2	1	0
P97	P96	P95	P94	P93	P92	P91	P90

P9n	Controls output data (in output mode) (n = 0 to 15)
0	Output 0.
1	Output 1.

Remark The port 9 register (P9) can only be read or written in 16-bit units.
If the higher 8 bits of the P9 register are used as P9H and the lower 8 bits as P9L, however, P9H and P9L can be manipulated in 8-bit or 1-bit units.

(b) Port 9 mode register (PM9)

After reset: FFFFH R/W Address: FFFFF432H, FFFFF433H

	15	14	13	12	11	10	9	8
PM9	PM915	PM914	PM913	PM912	PM911	PM910	PM99	PM98

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

PM9n	Controls input/output mode (n = 0 to 15)
0	Output mode
1	Input mode

Remark The PM9 register can only be read or written in 16-bit units.
If the higher 8 bits of the PM9 register are used as PM9H and the lower 8 bits as PM9L, however, PM9H and PM9L can be manipulated in 8-bit or 1-bit units.

(c) Port 9 mode control register (PMC9)

(1/2)

After reset: 0000H R/W Address: FFFFF452H, FFFFF453H

	15	14	13	12	11	10	9	8
PMC9	PMC915	PMC914	PMC913	PMC912	PMC911	PMC910	PMC99	PMC98

7	6	5	4	3	2	1	0
PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90

PMC915	Specifies operation mode of P915 pin
0	I/O port
1	A15/TI001 I/O

PMC914	Specifies operation mode of P914 pin
0	I/O port
1	A14/TI000 I/O

PMC913	Specifies operation mode of P913 pin
0	I/O port
1	A13/TI011 I/O

PMC912	Specifies operation mode of P912 pin
0	I/O port
1	A12/TI010 I/O

PMC911	Specifies operation mode of P911 pin
0	I/O port
1	A11/TI021 I/O

PMC910	Specifies operation mode of P910 pin
0	I/O port
1	A10/TI020 I/O

PMC99	Specifies operation mode of P99 pin
0	I/O port
1	A9/TI031 I/O

PMC98	Specifies operation mode of P98 pin
0	I/O port
1	A8/TI030 I/O

Remark The PMC9 register can only be read or written in 16-bit units.
If the higher 8 bits of the PMC9 register are used as PMC9H and the lower 8 bits as PMC9L, however, PMC9H and PMC9L can be manipulated in 8-bit or 1-bit units.

(2/2)

PMC97	Specifies operation mode of P97 pin
0	I/O port
1	A7 output

PMC96	Specifies operation mode of P96 pin
0	I/O port
1	A6 output

PMC95	Specifies operation mode of P95 pin
0	I/O port
1	A5 output

PMC94	Specifies operation mode of P94 pin
0	I/O port
1	A4 output

PMC93	Specifies operation mode of P93 pin
0	I/O port
1	A3 output

PMC92	Specifies operation mode of P92 pin
0	I/O port
1	A2 output

PMC91	Specifies operation mode of P91 pin
0	I/O port
1	A1 output

PMC90	Specifies operation mode of P90 pin
0	I/O port
1	A0 output

(d) Port 9 function control register (PFC9)

Caution To perform address bus output (A0 to A15), set the PFC9 register to 0000H, and then set the PMC9 register to FFFFH in 16-bit units.

After reset: 0000H R/W Address: FFFFF472H, FFFFF473H

	15	14	13	12	11	10	9	8
PFC9	PFC915	PFC914	PFC913	PFC912	PFC911	PFC910	PFC99	PFC98

	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

PFC915	Specifies operation mode of P915 pin in control mode
0	A15 output
1	TI001 input

PFC914	Specifies operation mode of P914 pin in control mode
0	A14 output
1	TI000 input

PFC913	Specifies operation mode of P913 pin in control mode
0	A13 output
1	TI011 input

PFC912	Specifies operation mode of P912 pin in control mode
0	A12 output
1	TI010 input

PFC911	Specifies operation mode of P911 pin in control mode
0	A11 output
1	TI021 input

PFC910	Specifies operation mode of P910 pin in control mode
0	A10 output
1	TI020 input

PFC99	Specifies operation mode of P99 pin in control mode
0	A9 output
1	TI031 input

PFC98	Specifies operation mode of P98 pin in control mode
0	A8 output
1	TI030 input

Remark The PFC9 register can only be read or written in 16-bit units.
If the higher 8 bits of the PFC9 register are used as PFC9H and the lower 8 bits as PFC9L, however, PFC9H and PFC9L can be manipulated in 8-bit or 1-bit units. However, this register is read-only if the PFC9L register is used in 1-bit units.

4.3.7 Port CM

Port CM is a 2-bit I/O port that can be set to the input or output mode in 1-bit units.

Port CM has an alternate function as the following pins.

Table 4-10. Alternate-Function Pins of Port CM

Pin Name		Alternate-Function Pin	I/O	PULL[Note]	Remark	Block Type
Port CM	PCM0	$\overline{\text{WAIT}}$	Input	None	–	A-1
	PCM1	CLKOUT	Output			B-1

Note Software pull-up function

(1) Registers

(a) Port CM register (PCM)

After reset: 00H (output latch) R/W Address: FFFFF00CH

	7	6	5	4	3	2	1	0
PCM	0	0	0	0	0	0	PCM1	PCM0

PCMn	Controls output data (in output mode) (n = 0, 1)
0	Output 0.
1	Output 1.

(b) Port CM mode register (PMCM)

After reset: FFH R/W Address: FFFFF02CH

	7	6	5	4	3	2	1	0
PMCM	1	1	1	1	1	1	PMCM1	PMCM0

PMCMn	Controls input/output mode (n = 0, 1)
0	Output mode
1	Input mode

(c) Port CM mode control register (PMCCM)

After reset: **Note** R/W Address: FFFFF04CH

	7	6	5	4	3	2	1	0
PMCCM	0	0	0	0	0	0	PMCCM1	PMCCM0

PMCCM1	Specifies operation mode of PCM1 pin
0	I/O port
1	CLKOUT output

PMCCM0	Specifies operation mode of PCM0 pin
0	I/O port
1	$\overline{\text{WAIT}}$ input

Note In single-chip mode: 00H
In ROMless mode: 01H

4.3.8 Port CS

Port CS is a 3-bit I/O port that can be set to the input or output mode in 1-bit units.

Port CS has an alternate function as the following pins.

Table 4-11. Alternate-Function Pins of Port CS

Pin Name		Alternate-Function Pin	I/O	PULL[Note]	Remark	Block Type
Port CS	PCS0	$\overline{\text{CS0}}$	Output	None	–	B-1
	PCS1	$\overline{\text{CS1}}$	Output			B-1
	PCS2	$\overline{\text{CS2}}$	Output			B-1

Note Software pull-up function

(1) Registers

(a) Port CS register (PCS)

After reset: 00H (output latch) R/W Address: FFFFF008H

	7	6	5	4	3	2	1	0
PCS	0	0	0	0	0	PCS2	PCS1	PCS0

PCSn	Controls output data (in output mode) (n = 0 to 2)
0	Output 0.
1	Output 1.

(b) Port CS mode register (PMCS)

After reset: FFH R/W Address: FFFFF028H

	7	6	5	4	3	2	1	0
PMCS	1	1	1	1	1	PMCS2	PMCS1	PMCS0

PMCSn	Controls input/output mode (n = 0 to 2)
0	Output mode
1	Input mode

(c) Port CS mode control register (PMCCS)

After reset: **Note** R/W Address: FFFFF048H

	7	6	5	4	3	2	1	0
PMCCS	0	0	0	0	0	PMCCS2	PMCCS1	PMCCS0

PMCCSn	Specifies operation mode of PCSn pin (n = 0 to 2)
0	I/O port
1	$\overline{\text{CSn}}$ output

Note In single-chip mode: 00H
In ROMless mode: 07H

4.3.9 Port CT

Port CT is a 3-bit I/O port that can be set to the input or output mode in 1-bit units.

Port CT has an alternate function as the following pins.

Table 4-12. Alternate-Function Pins of Port CT

Pin Name		Alternate-Function Pin	I/O	PULL[Note]	Remark	Block Type
Port CT	PCT0	$\overline{\text{WR0}}$	Output	None	–	B-1
	PCT1	$\overline{\text{WR1}}$	Output			B-1
	PCT4	$\overline{\text{RD}}$	Output			B-1

Note Software pull-up function

(1) Registers

(a) Port CT register (PCT)

After reset: 00H (output) R/W Address: FFFFF00AH

	7	6	5	4	3	2	1	0
PCT	0	0	0	PCT4	0	0	PCT1	PCT0

PCTn	Controls output data (in output mode) (n = 0, 1, 4)
0	Output 0.
1	Output 1.

(b) Port CT mode register (PMCT)

After reset: FFH R/W Address: FFFFF02AH

	7	6	5	4	3	2	1	0
PMCT	1	1	1	PMCT4	1	1	PMCT1	PMCT0

PMCTn	Controls input/output mode (n = 0, 1, 4)
0	Output mode
1	Input mode

(c) Port CT mode control register (PMCCT)

After reset: **Note** R/W Address: FFFFF04AH

	7	6	5	4	3	2	1	0
PMCCT	0	0	0	PMCCT4	0	0	PMCCT1	PMCCT0

PMCCT4	Specifies operation mode of PCT4 pin
0	I/O port
1	$\overline{\text{RD}}$ output

PMCCT1	Specifies operation mode of PCT1 pin
0	I/O port
1	$\overline{\text{WR1}}$ output

PMCCT0	Specifies operation mode of PCT0 pin
0	I/O port
1	$\overline{\text{WR0}}$ output

Note In single-chip mode: 00H
In ROMless mode: 13H

4.3.10 Port DH

Port DH is a 3-bit I/O port that can be set to the input or output mode in 1-bit units.

Port DH has an alternate function as the following pins.

Table 4-13. Alternate-Function Pins of Port DH

Pin Name		Alternate-Function Pin	I/O	PULL[Note]	Remark	Block Type
Port DH	PDH0	A16	Output	None	–	B-2
	PDH1	A17	Output			B-2
	PDH2	A18	Output			B-2

Note Software pull-up function

(1) Registers

(a) Port DH register (PDH)

After reset: 00H (output latch) R/W Address: FFFFF006H

	7	6	5	4	3	2	1	0
PDH	0	0	0	0	0	PDH2	PDH1	PDH0

PDHn	Controls output data (in output mode) (n = 0 to 2)
0	Output 0.
1	Output 1.

(b) Port DH mode register (PMDH)

After reset: FFH R/W Address: FFFFF026H

	7	6	5	4	3	2	1	0
PMDH	1	1	1	1	1	PMDH2	PMDH1	PMDH0

PMDHn	Controls input/output mode (n = 0 to 2)
0	Output mode
1	Input mode

(c) Port DH mode control register (PMCDH)

After reset: **Note** R/W Address: FFFFF046H

	7	6	5	4	3	2	1	0
PMCDH	0	0	0	0	0	PMCDH2	PMCDH1	PMCDH0

PMCDHn	Specifies operation mode of PDHn pin (n = 0 to 2)
0	I/O port
1	Am output (address bus output) (m = 16 to 18)

Note In single-chip mode: 00H
In ROMless mode: 07H

4.3.11 Port DL

Port DL is a 16-bit I/O port that can be set to the input or output mode in 1-bit units.

Port DL has an alternate function as the following pins.

Table 4-14. Alternate-Function Pins of Port DL

Pin Name		Alternate-Function Pin	I/O	PULL[Note]	Remark	Block Type
Port DL	PDL0	D0	I/O	None	–	C-1
	PDL1	D1	I/O			C-1
	PDL2	D2	I/O			C-1
	PDL3	D3	I/O			C-1
	PDL4	D4	I/O			C-1
	PDL5	D5	I/O			C-1
	PDL6	D6	I/O			C-1
	PDL7	D7	I/O			C-1
	PDL8	D8	I/O			C-1
	PDL9	D9	I/O			C-1
	PDL10	D10	I/O			C-1
	PDL11	D11	I/O			C-1
	PDL12	D12	I/O			C-1
	PDL13	D13	I/O			C-1
	PDL14	D14	I/O			C-1
	PDL15	D15	I/O			C-1

Note Software pull-up function

(1) Registers

(a) Port DL register (PDL)

After reset: Undefined R/W Address: FFFFF004H, FFFFF005H

	15	14	13	12	11	10	9	8
PDL	PDL15	PDL14	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8

	7	6	5	4	3	2	1	0
	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0

PDLn	Controls output data (in output mode) (n = 0 to 15)
0	Output 0.
1	Output 1.

Remark The port DL register (PDL) can only be read or written in 16-bit units.

If the higher 8 bits of the PDL register are used as PDLH, and the lower 8 bits as PDLL, however, PDLH and PDLL can be used as an 8-bit I/O port whose input or output can be manipulated in 8-bit or 1-bit units.

(b) Port DL mode register (PMDL)

After reset: FFFFH R/W Address: FFFFF024H, FFFFF025H

	15	14	13	12	11	10	9	8
PMDL	PMDL15	PMDL14	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8

	7	6	5	4	3	2	1	0
	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0

PMDLn	Controls input/output mode (n = 0 to 15)
0	Output mode
1	Input mode

Remark The PMDL register can only be read or written in 16-bit units.

If the higher 8 bits of the PMDL register are used as PMDLH, and the lower 8 bits as PMDLL, however, PMDLH and PMDLL can be read or written in 8-bit or 1-bit units.

(c) Port DL mode control register (PMCDL)

After reset: **Note** R/W Address: FFFFF044H, FFFFF045H

	15	14	13	12	11	10	9	8
PMCDL	PMCDL15	PMCDL14	PMCDL13	PMCDL12	PMCDL11	PMCDL10	PMCDL9	PMCDL8

7	6	5	4	3	2	1	0
PMCDL7	PMCDL6	PMCDL5	PMCDL4	PMCDL3	PMCDL2	PMCDL1	PMCDL0

PMCDLn	Specifies operation mode of PDLn pin (n = 0 to 15)
0	I/O port
1	Dn I/O (data bus I/O)

Note In single-chip mode: 0000H
In ROMless mode: FFFFH

Caution **Do not specify D8 to D15 when the BS20 to BS00 bits of the BSC register = 0 (8-bit bus width).**

Remark The PMCDL register can only be read or written in 16-bit units.
If the higher 8 bits of the PMCDL register are used as PMCDLH, and the lower 8 bits as PMCDLL, however, PMCDLH and PMCDLL can be read or written in 8-bit or 1-bit units.

★

4.4 Block Diagram

Figure 4-2. Block Diagram of Type A-1

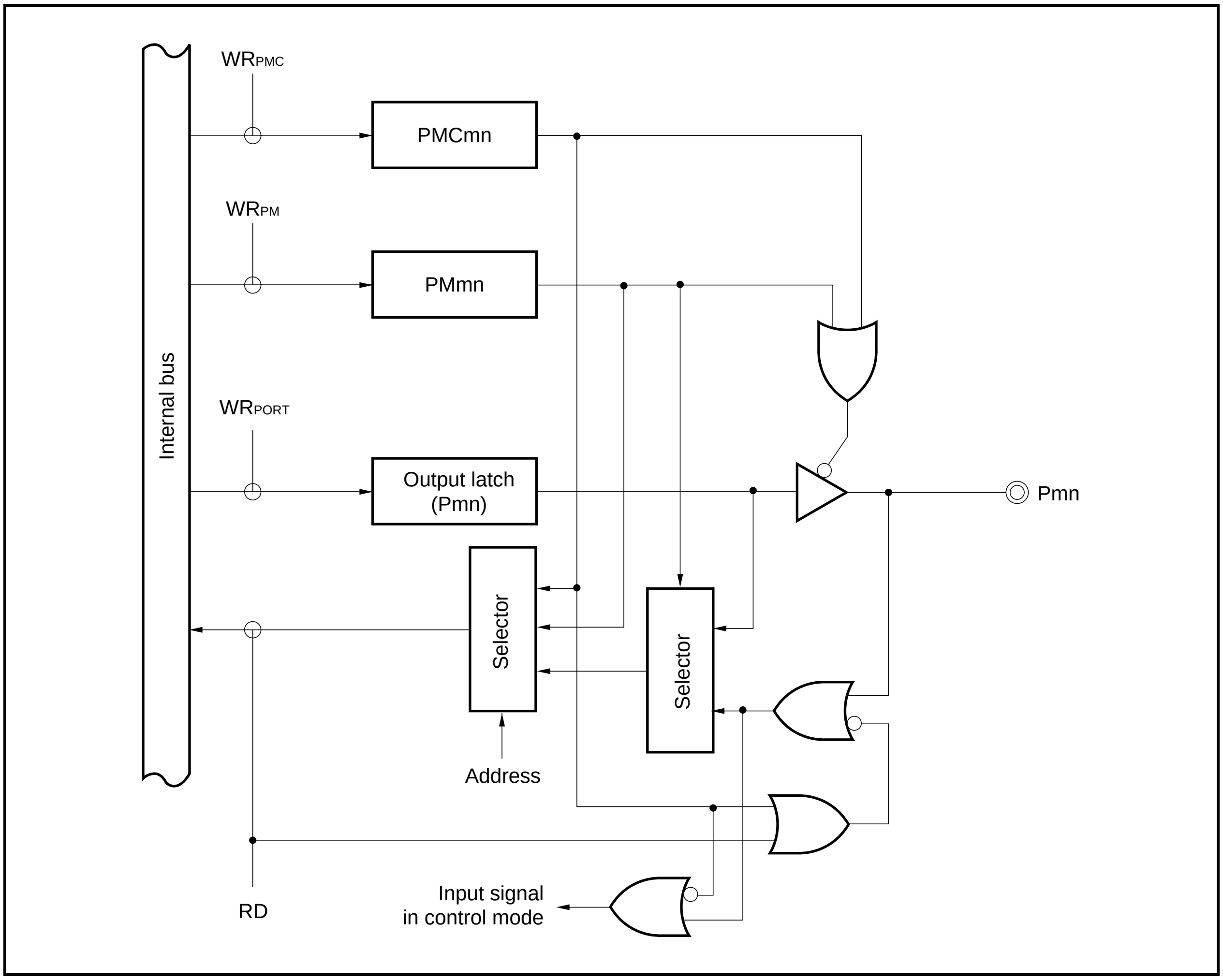


Figure 4-3. Block Diagram of Type A-2 (1/2)

(a) Without noise elimination

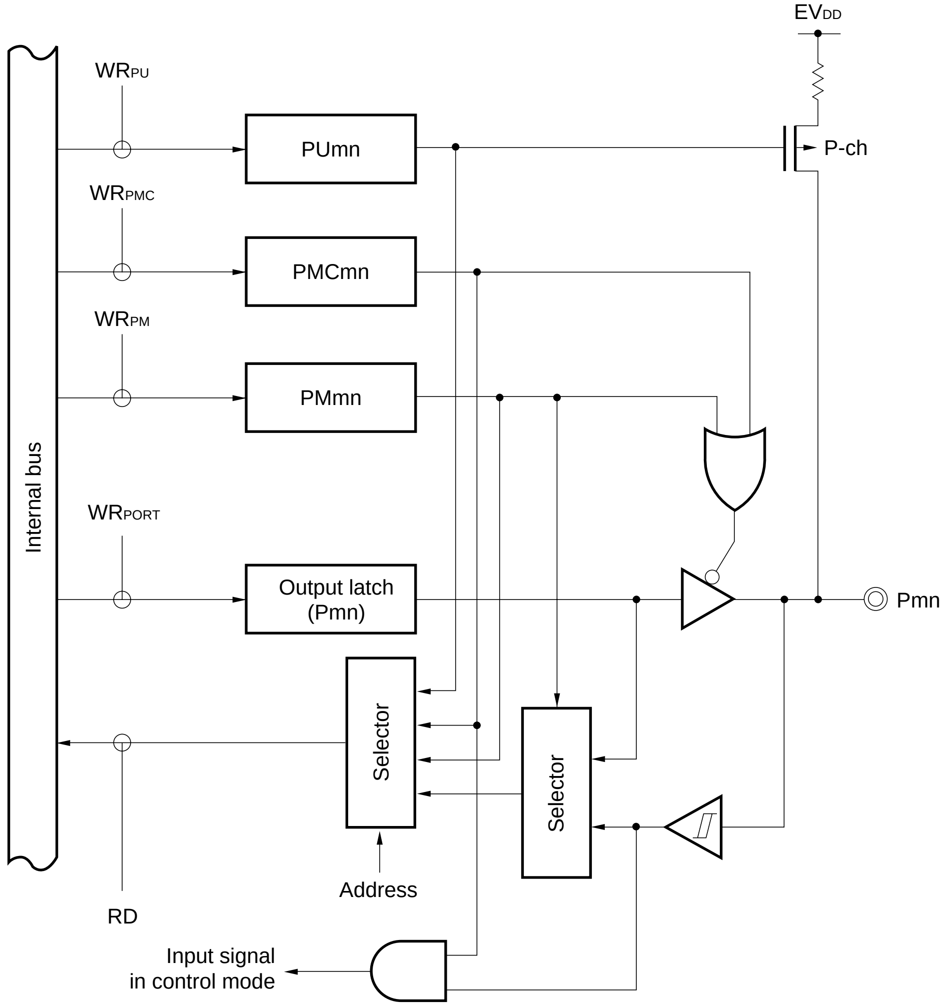


Figure 4-3. Block Diagram of Type A-2 (2/2)

(b) With noise elimination

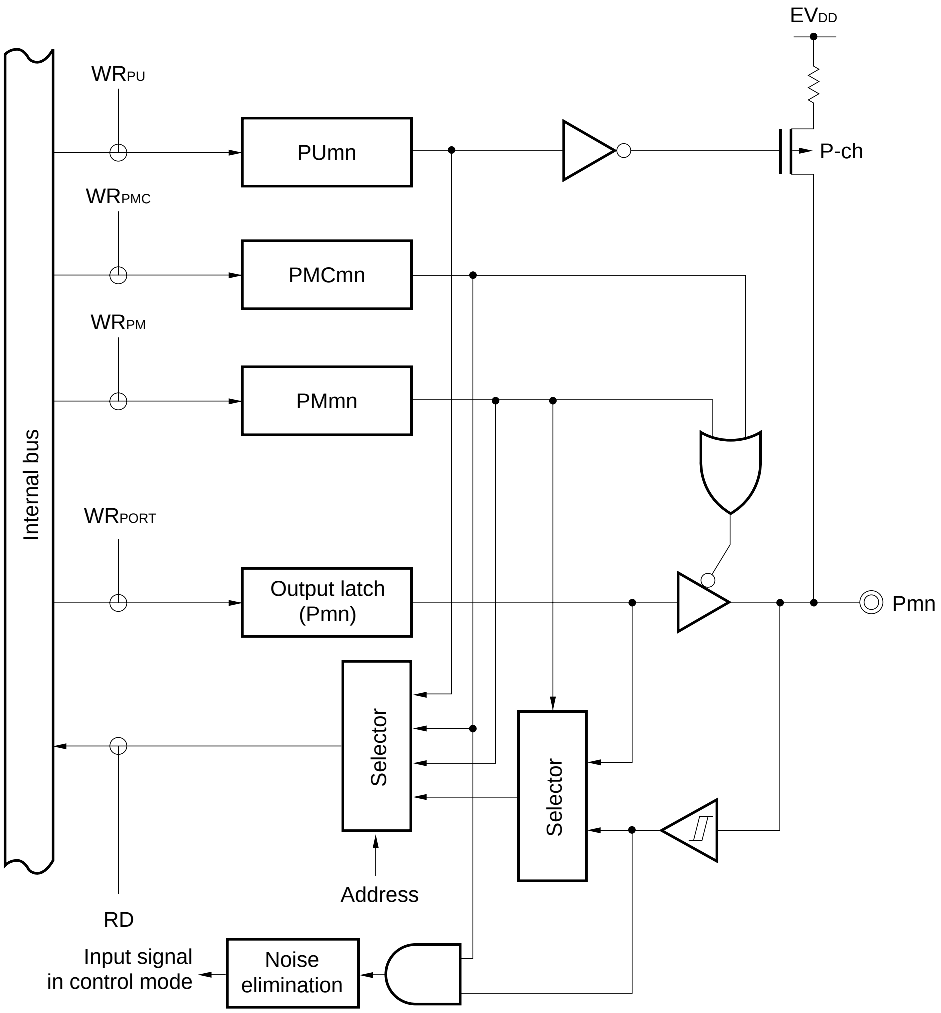


Figure 4-4. Block Diagram of Type A-3

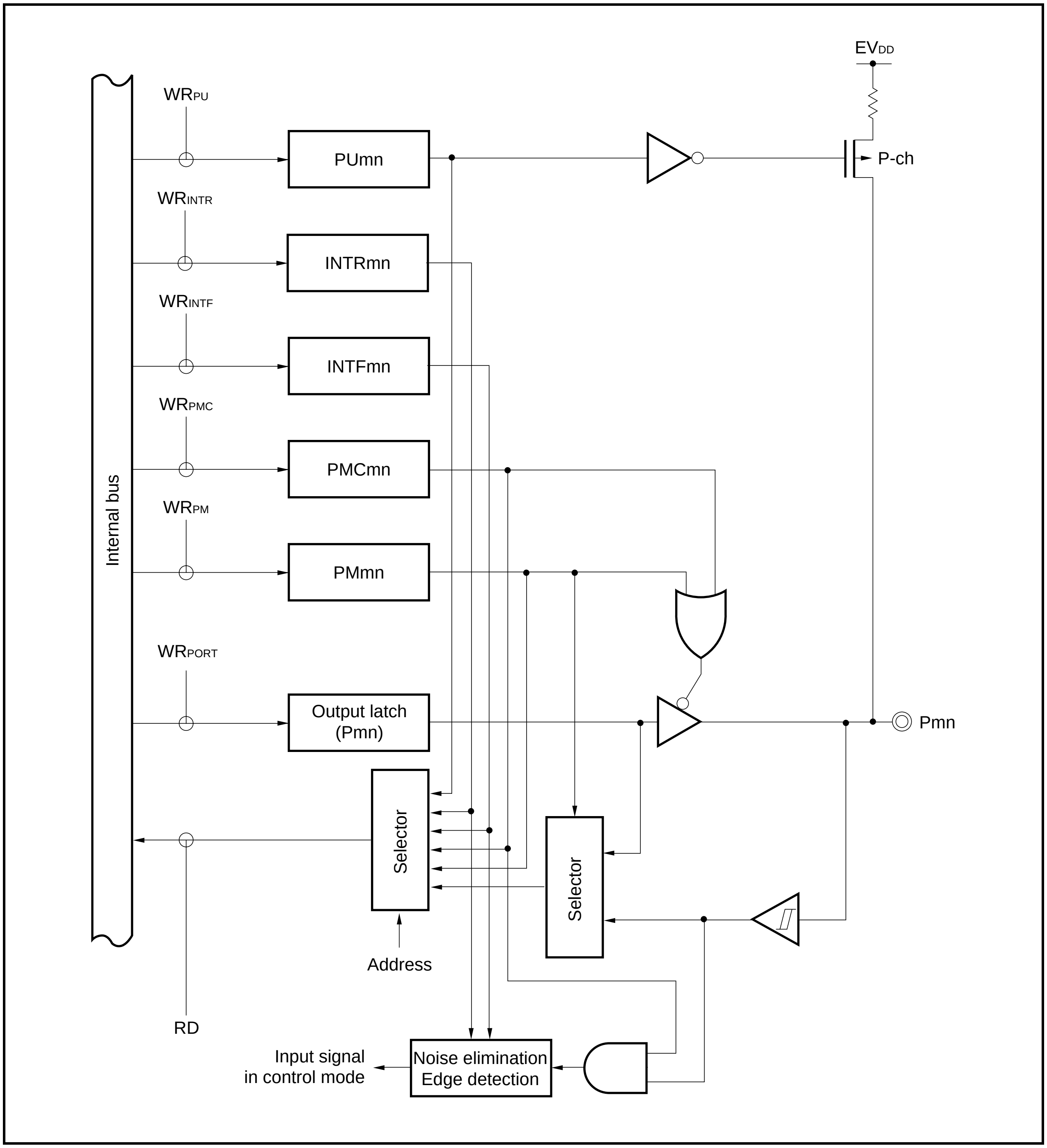


Figure 4-5. Block Diagram of Type A-4

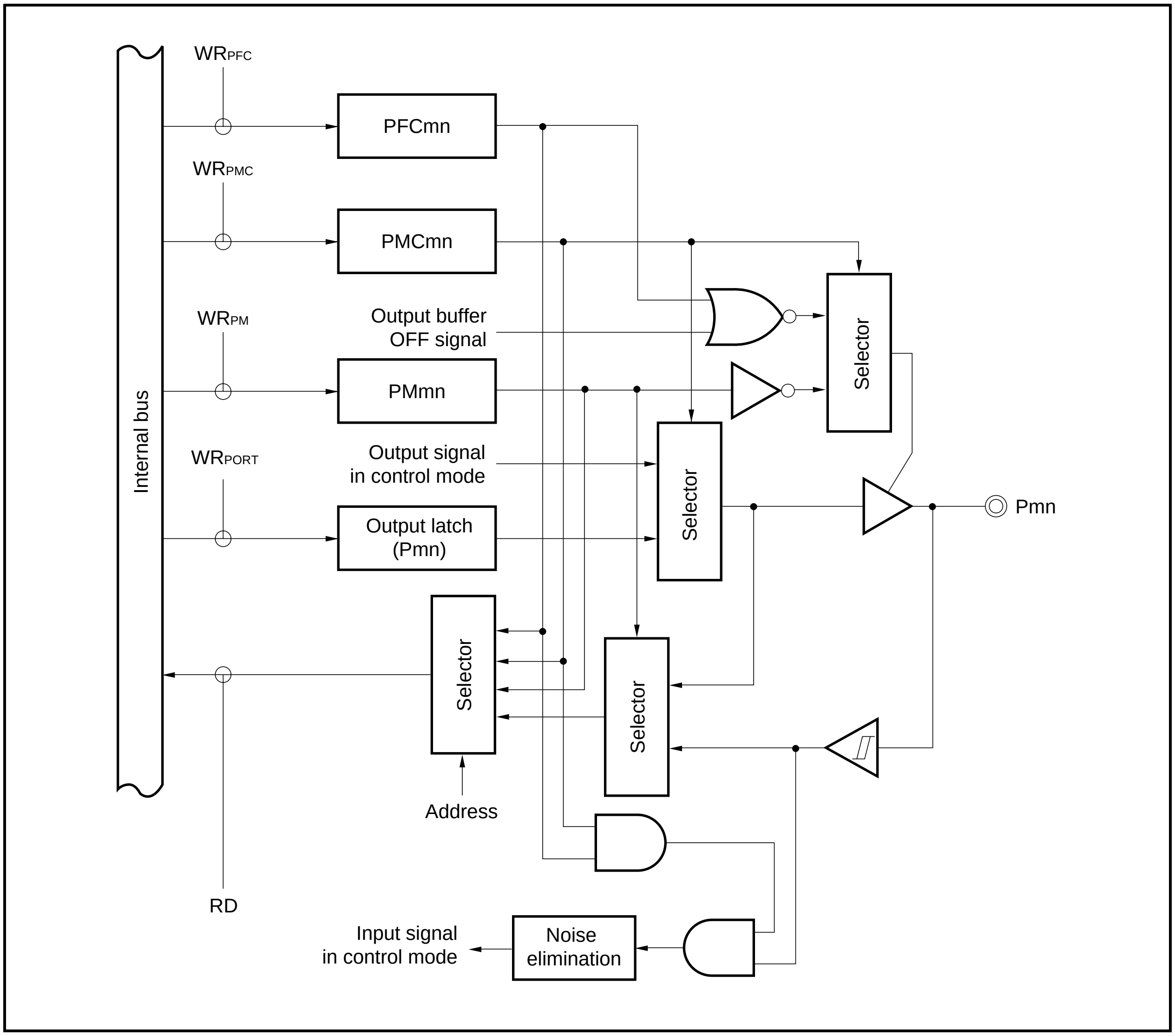


Figure 4-6. Block Diagram of Type A-5

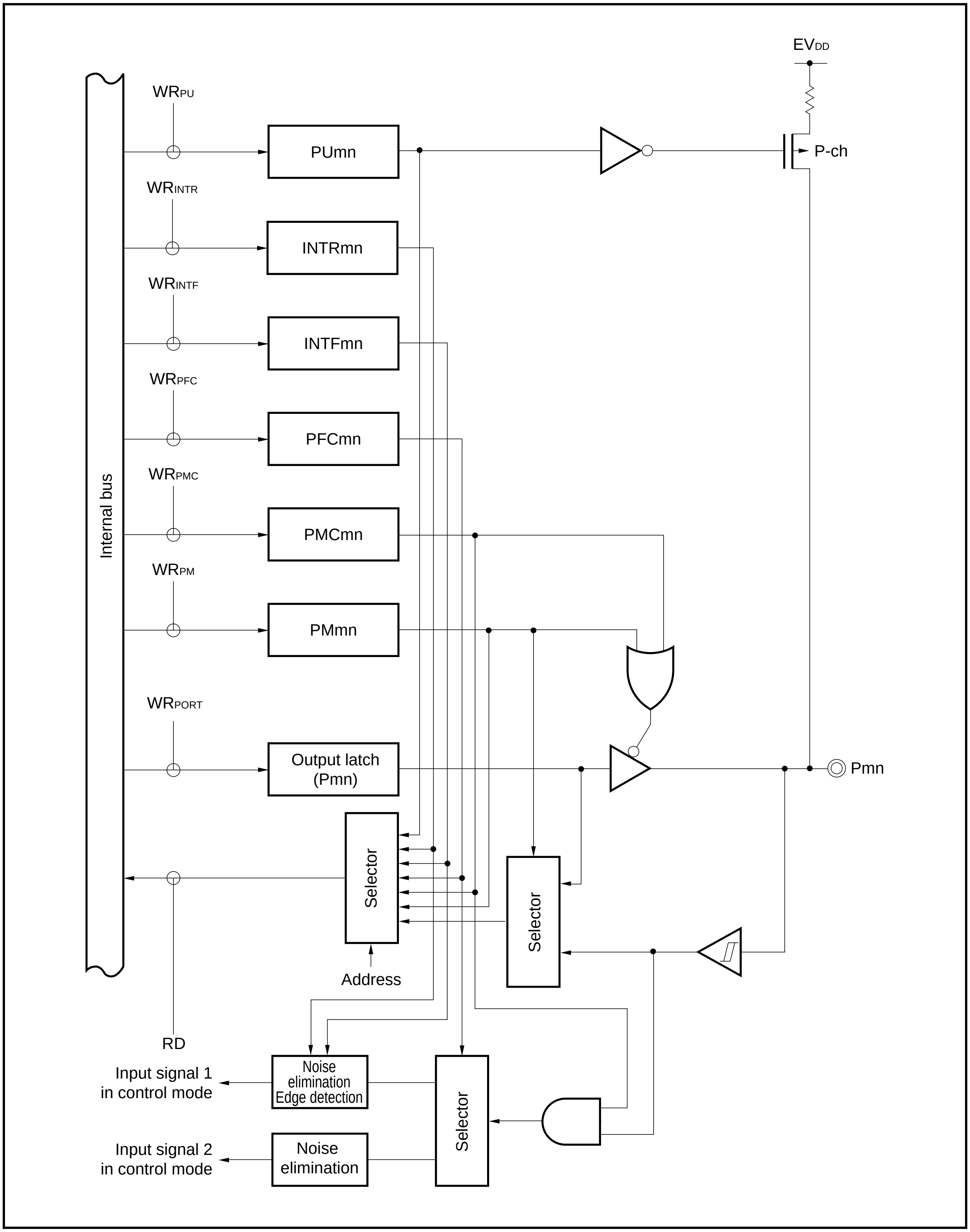


Figure 4-7. Block Diagram of Type A-6

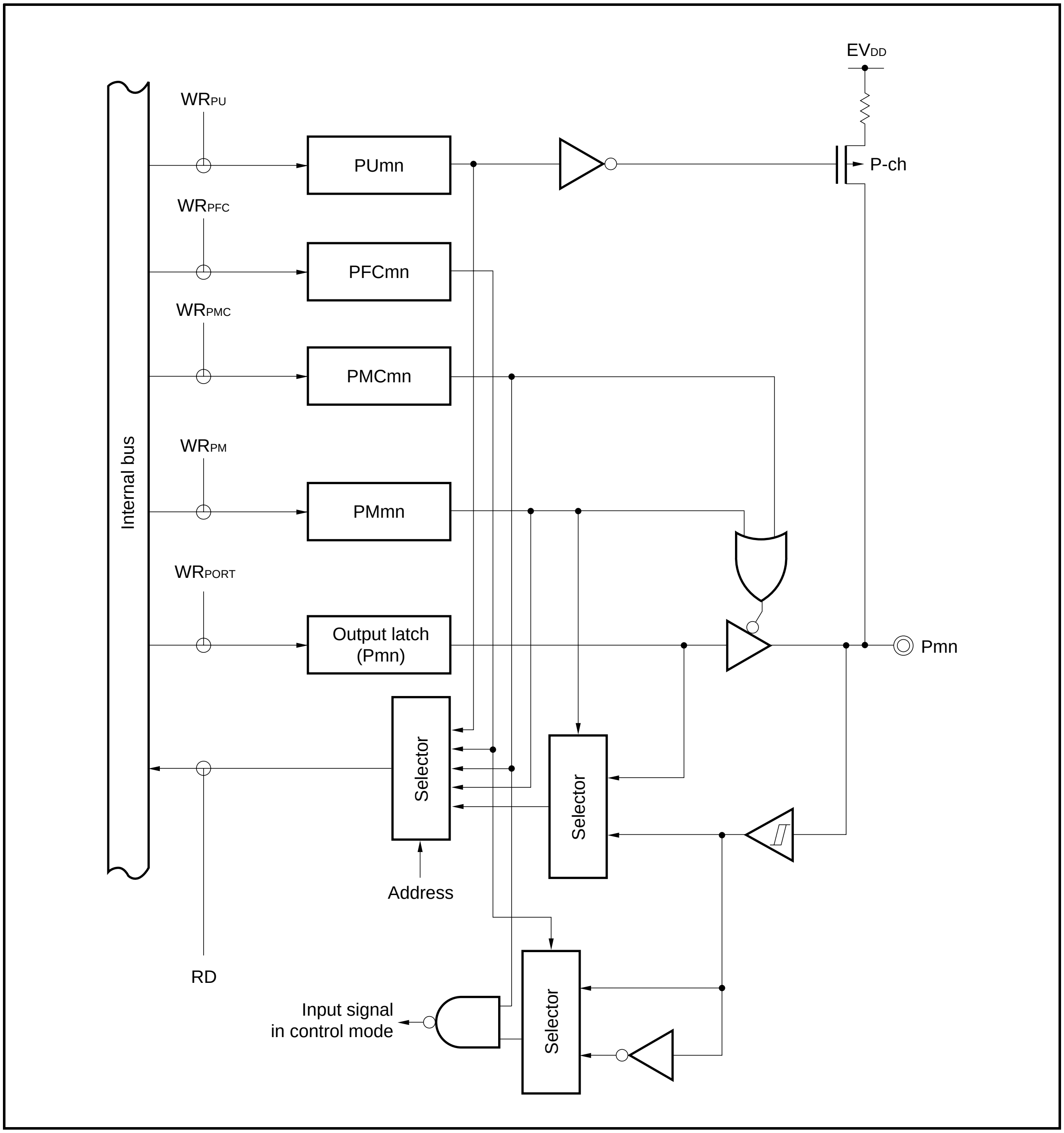


Figure 4-8. Block Diagram of Type B-1

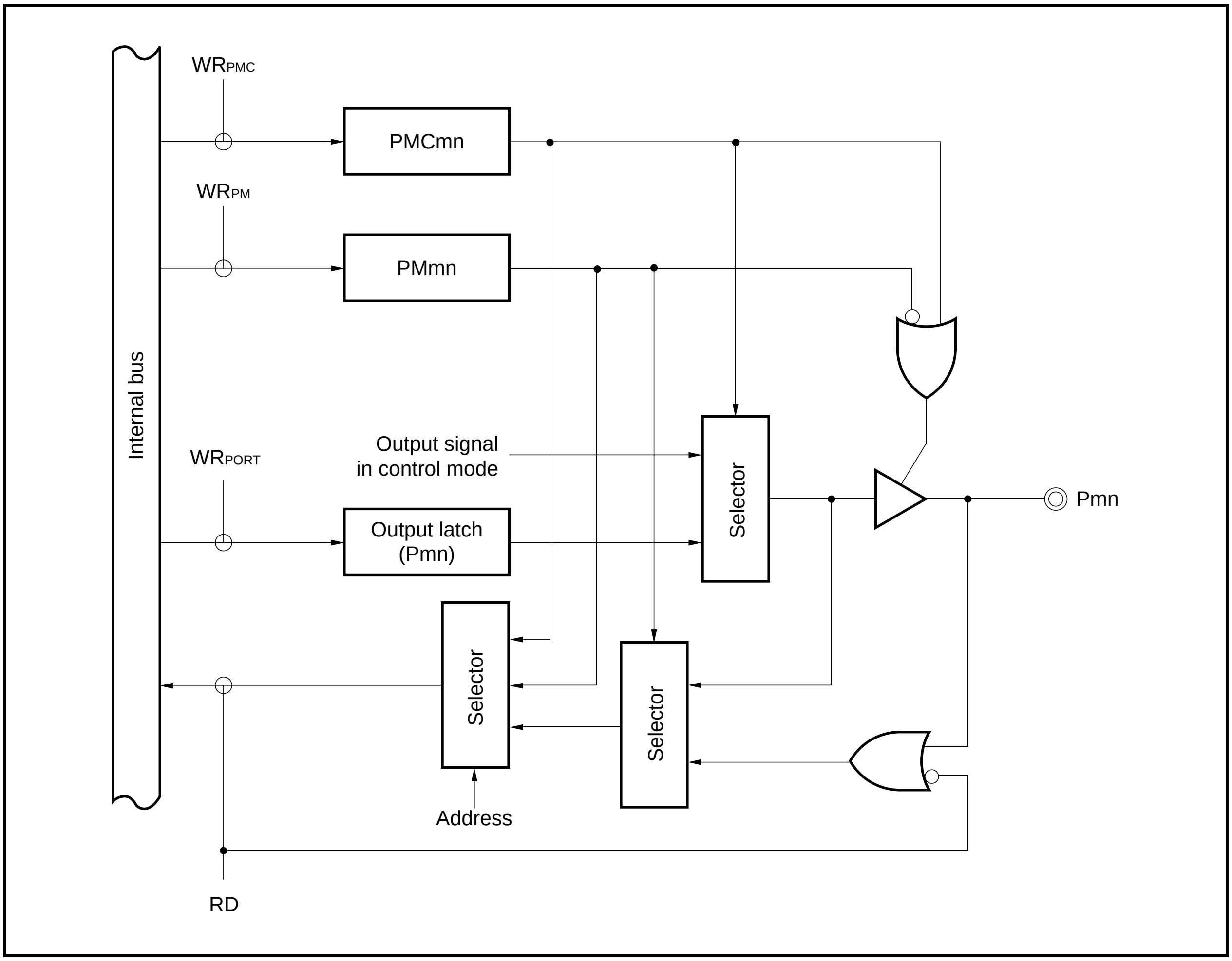


Figure 4-9. Block Diagram of Type B-2

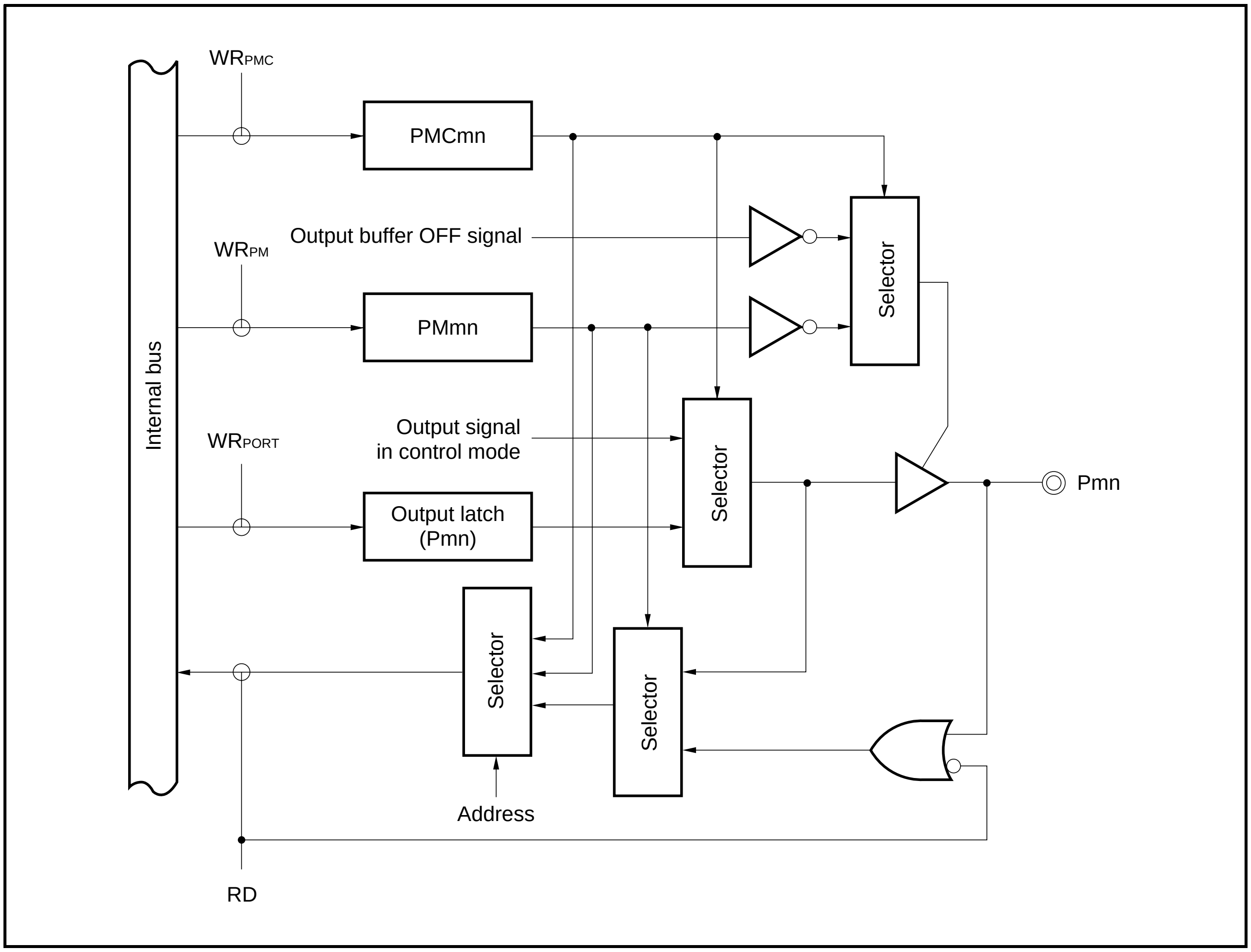


Figure 4-10. Block Diagram of Type B-3

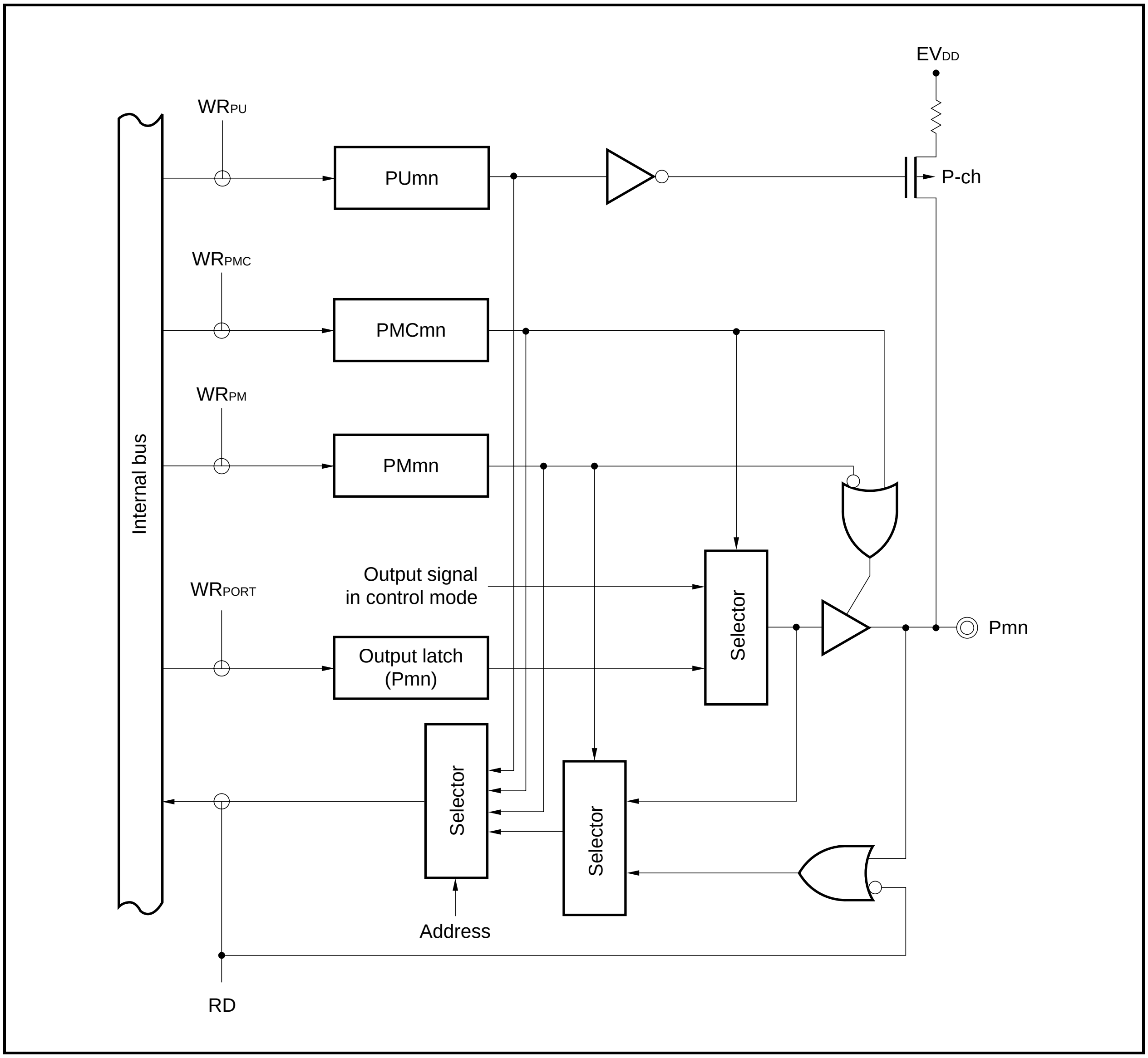


Figure 4-11. Block Diagram of Type B-4

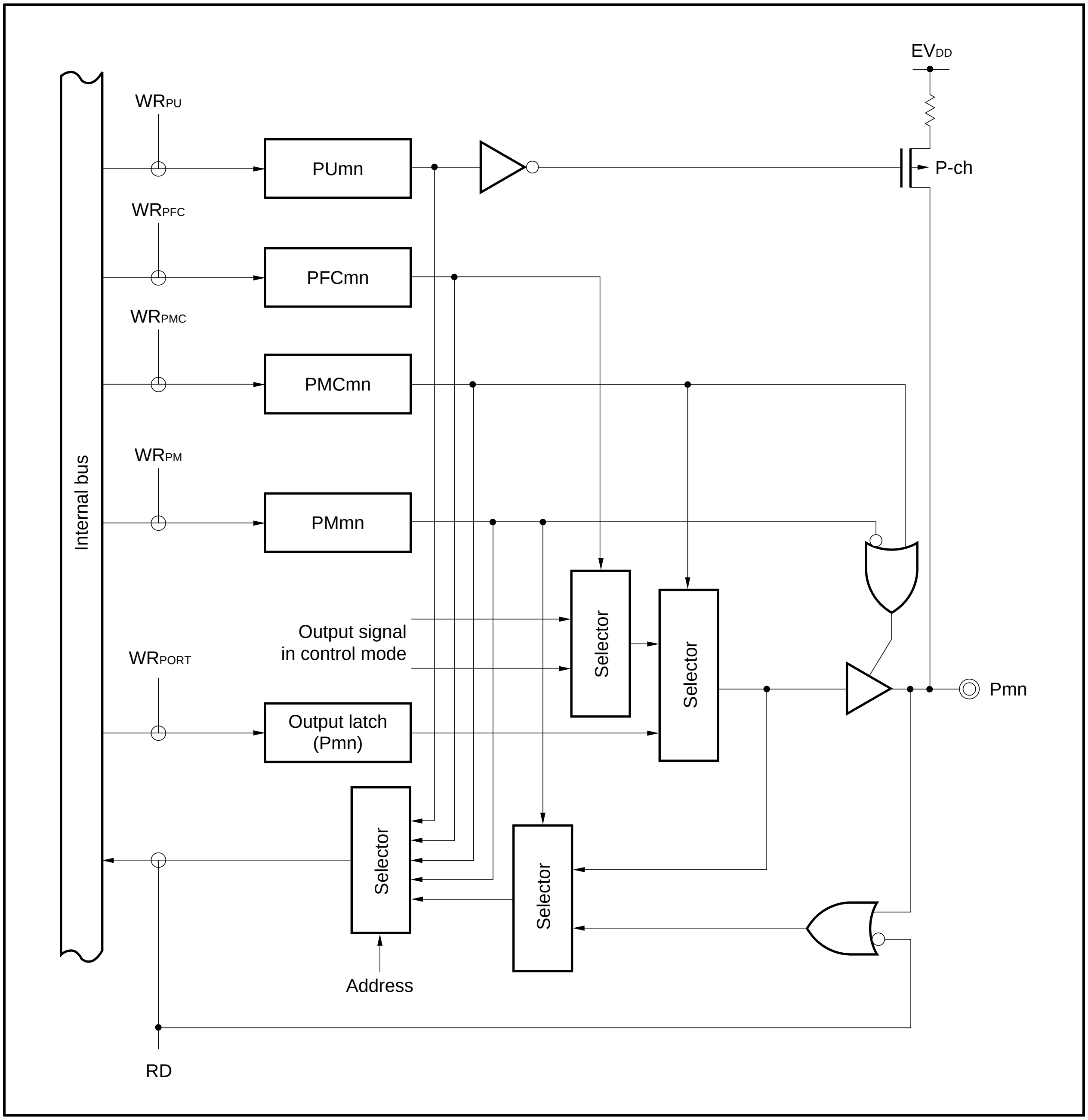


Figure 4-12. Block Diagram of Type C-1

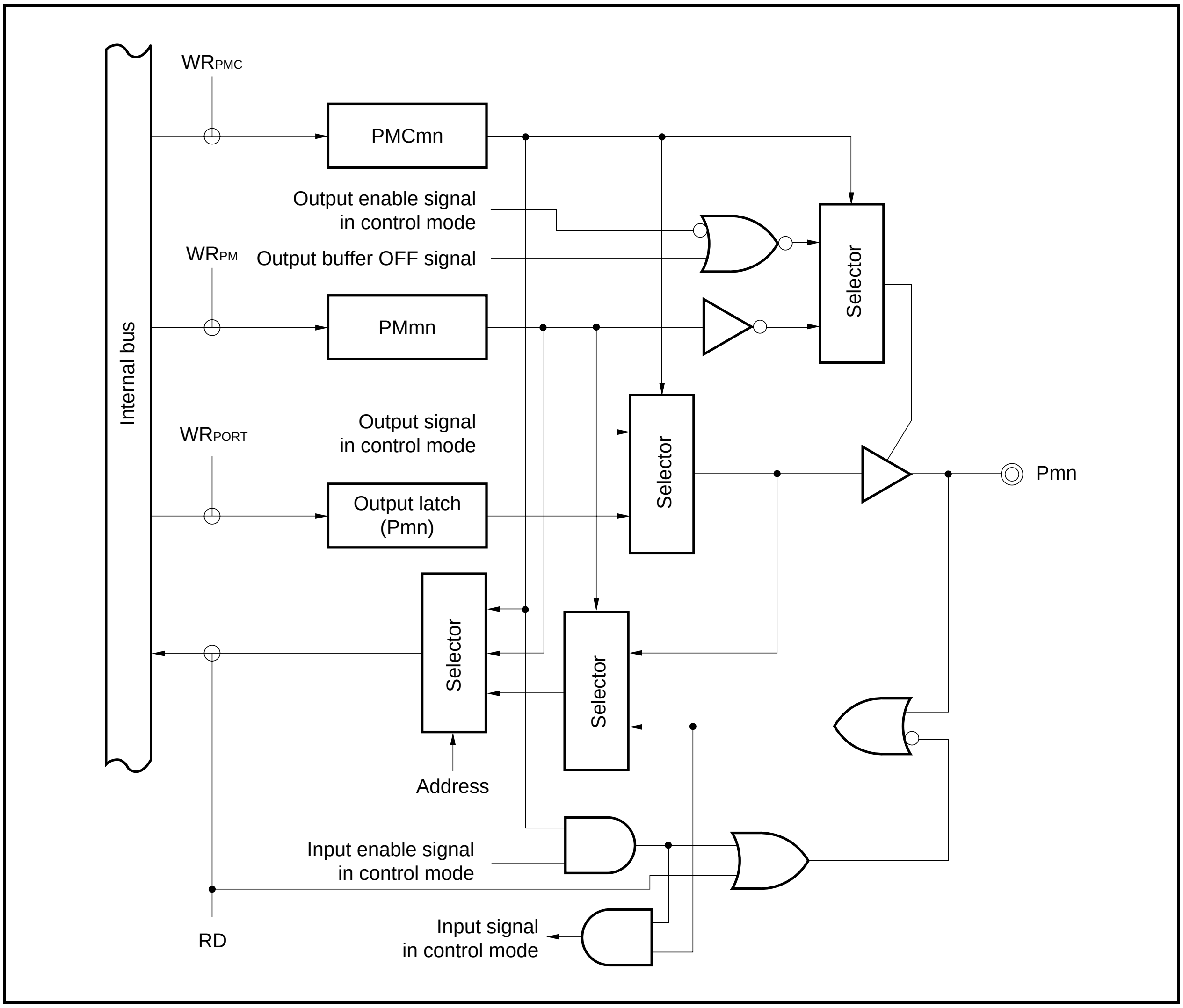


Figure 4-13. Block Diagram of Type C-2

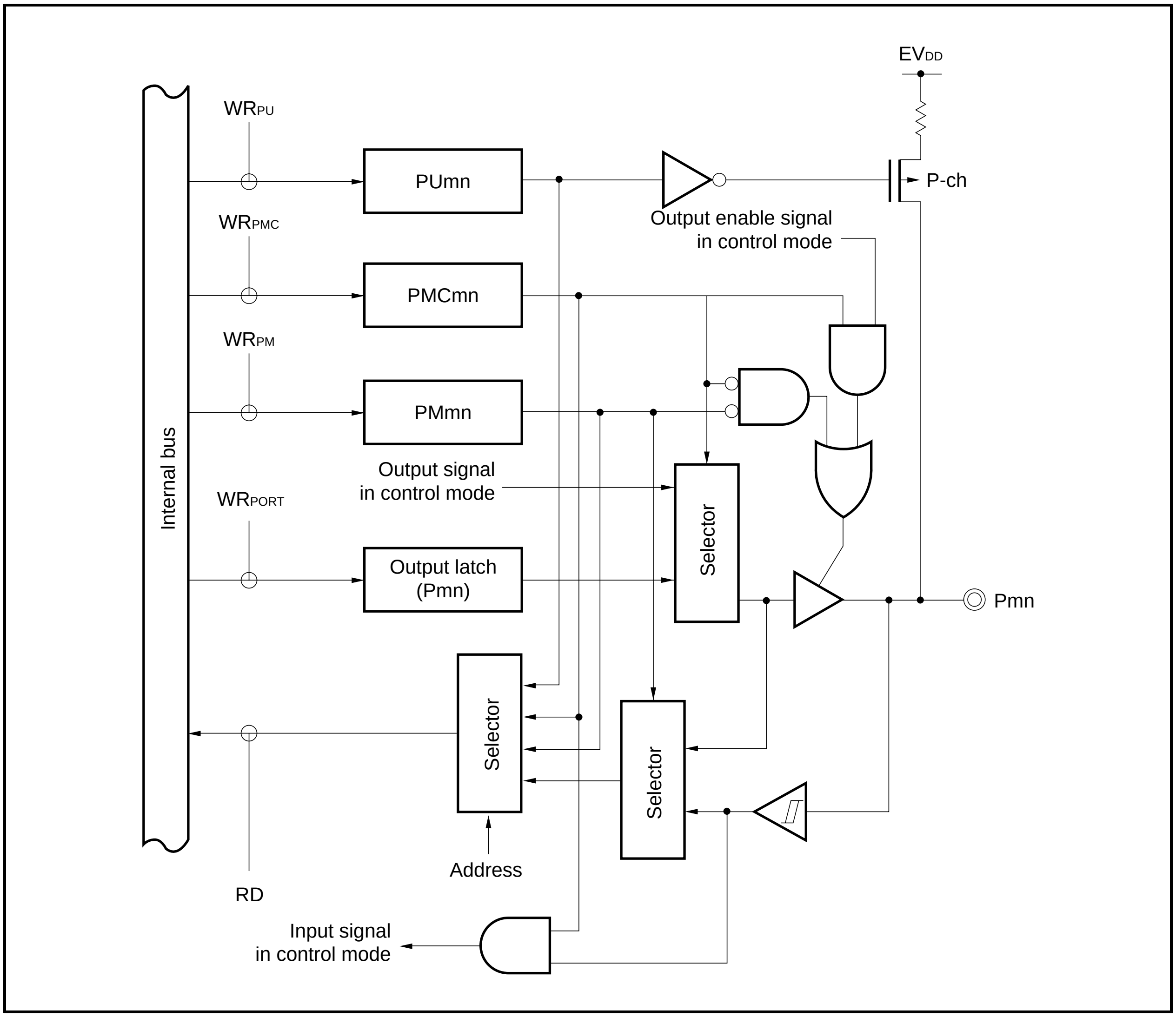


Figure 4-14. Block Diagram of Type D-1

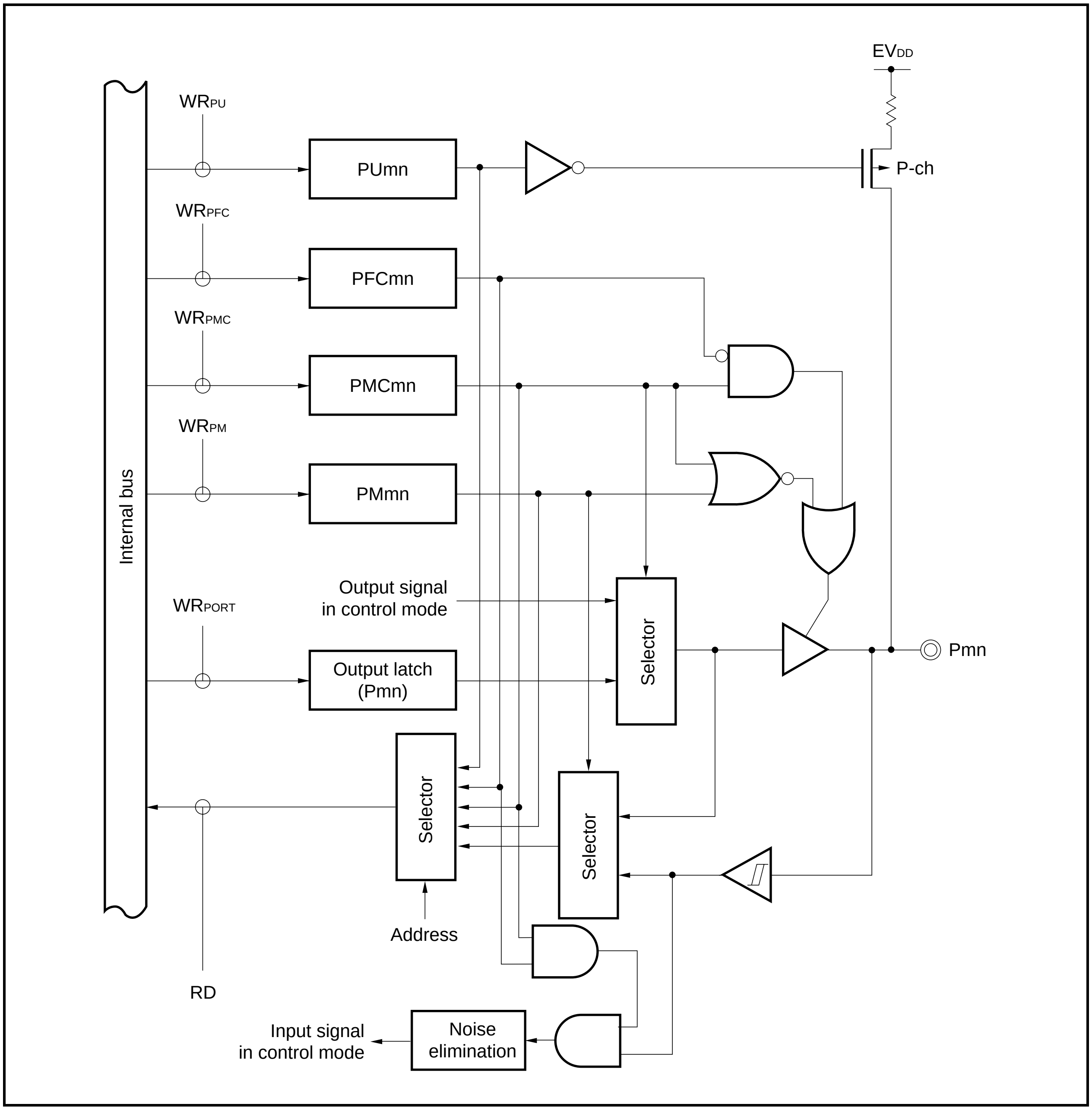
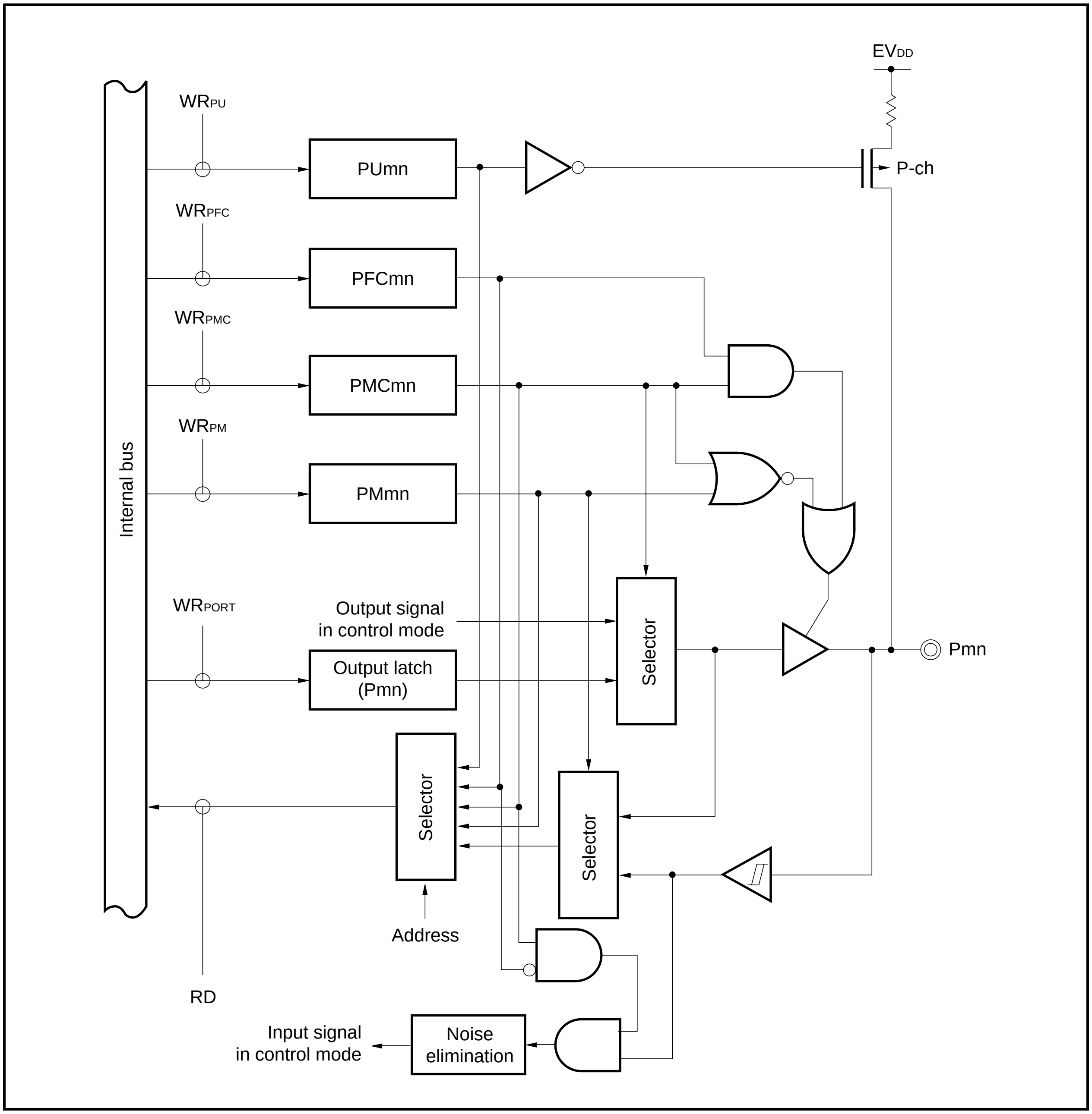


Figure 4-15. Block Diagram of Type D-2



4.5 Register Settings for Ports When Alternate Function Is Used

Table 4-15 shows the register settings for the ports when each port pin is used as its alternate function pin. When using port pins as their alternate function pins, refer to the descriptions of the respective functions.

Table 4-15. Settings When Port Pins Are Used for Alternate Functions (1/5)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)
	Name	I/O					
P00	NMI	Input	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	–	INTR00 (INTR0), INTF00 (INTF0)
P01	INTP0	Input	P01 = Setting not required	PM01 = Setting not required	PMC01 = 1	–	INTR01 (INTR0), INTF01 (INTF0)
P02	INTP1	Input	P02 = Setting not required	PM02 = Setting not required	PMC02 = 1	–	INTR02 (INTR0), INTF02 (INTF0)
P03	INTP2	Input	P03 = Setting not required	PM03 = Setting not required	PMC03 = 1	PFC03 = 0	INTR03 (INTR0), INTF03 (INTF0)
	TI20	Input	P03 = Setting not required	PM03 = Setting not required	PMC03 = 1	PFC03 = 1	TCL202 to TCL200 (TCL20)
P10	PWM0	Output	P10 = Setting not required	PM10 = Setting not required	PMC10 = 1	–	–
P11	TO00	Output	P11 = Setting not required	PM11 = Setting not required	PMC11 = 1	PFC11 = 0	–
	PWM1	Output	P11 = Setting not required	PM11 = Setting not required	PMC11 = 1	PFC11 = 1	–
P12	TO01	Output	P12 = Setting not required	PM12 = Setting not required	PMC12 = 1	PFC12 = 0	–
	PWM2	Output	P12 = Setting not required	PM12 = Setting not required	PMC12 = 1	PFC12 = 1	–
P13	TO20	Output	P13 = Setting not required	PM13 = Setting not required	PMC13 = 1	PFC13 = 0	–
	PWM3	Output	P13 = Setting not required	PM13 = Setting not required	PMC13 = 1	PFC13 = 1	–
P14	TO21	Output	P14 = Setting not required	PM14 = Setting not required	PMC14 = 1	PFC14 = 0	–
	TI21	Input	P14 = Setting not required	PM14 = Setting not required	PMC14 = 1	PFC14 = 1	TCL212 to TCL210 (TCL21)
P20	TO02	Output	P20 = Setting not required	PM20 = Setting not required	PMC20 = 1	–	–
P21	TO03	Output	P21 = Setting not required	PM21 = Setting not required	PMC21 = 1	–	–

Table 4-15. Settings When Port Pins Are Used for Alternate Functions (2/5)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)
	Name	I/O					
P30	RXD0	Input	P30 = Setting not required	PM30 = Setting not required	PMC30 = 1	PFC30 = 0	–
	$\overline{\text{RXD0}}$	Input	P30 = Setting not required	PM30 = Setting not required	PMC30 = 1	PFC30 = 1	–
P31	TXD0	Output	P31 = Setting not required	PM31 = Setting not required	PMC31 = 1	–	–
P32	SI1	Input	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	–	–
P33	SO1	Output	P33 = Setting not required	PM33 = Setting not required	PMC33 = 1	–	–
P34	$\overline{\text{SCK1}}$	I/O	P34 = Setting not required	PM34 = Setting not required	PMC34 = 1	–	–
P35	INTP100	Input	P35 = Setting not required	PM35 = Setting not required	PMC35 = 1	–	Set the valid edge by using the SES10 register
	TI10	Input	P35 = Setting not required	PM35 = Setting not required	PMC35 = 1	–	
	TCLR10	Input	P35 = Setting not required	PM35 = Setting not required	PMC35 = 1	–	
P36	INTP110	Input	P36 = Setting not required	PM36 = Setting not required	PMC36 = 1	–	Set the valid edge by using the SES11 register
	TI11	Input	P36 = Setting not required	PM36 = Setting not required	PMC36 = 1	–	
	TCLR11	Input	P36 = Setting not required	PM36 = Setting not required	PMC36 = 1	–	
P40	SI0	Input	P40 = Setting not required	PM40 = Setting not required	PMC40 = 1	–	–
P41	SO0	Output	P41 = Setting not required	PM41 = Setting not required	PMC41 = 1	–	–
P42	$\overline{\text{SCK0}}$	I/O	P42 = Setting not required	PM42 = Setting not required	PMC42 = 1	–	–
P43	RXD1	Input	P43 = Setting not required	PM43 = Setting not required	PMC43 = 1	PFC43 = 0	–
	$\overline{\text{RXD1}}$	Input	P43 = Setting not required	PM43 = Setting not required	PMC43 = 1	PFC43 = 1	–
P44	TXD1	Output	P44 = Setting not required	PM44 = Setting not required	PMC44 = 1	–	–
P45	INTP101	Input	P45 = Setting not required	PM45 = Setting not required	PMC45 = 1	PFC45 = 0	Set the valid edge by using the SES10 register
	TO10	Output	P45 = Setting not required	PM45 = Setting not required	PMC45 = 1	PFC45 = 1	–
P46	INTP111	Input	P46 = Setting not required	PM46 = Setting not required	PMC46 = 1	PFC46 = 0	Set the valid edge by using the SES11 register
	TO11	Output	P46 = Setting not required	PM46 = Setting not required	PMC46 = 1	PFC46 = 1	–

Table 4-15. Settings When Port Pins Are Used for Alternate Functions (3/5)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)
	Name	I/O					
P90	A0	Output	P90 = Setting not required	PM90 = Setting not required	PMC90 = 1	–	**Note**
P91	A1	Output	P91 = Setting not required	PM91 = Setting not required	PMC91 = 1	–	**Note**
P92	A2	Output	P92 = Setting not required	PM92 = Setting not required	PMC92 = 1	–	**Note**
P93	A3	Output	P93 = Setting not required	PM93 = Setting not required	PMC93 = 1	–	**Note**
P94	A4	Output	P94 = Setting not required	PM94 = Setting not required	PMC94 = 1	–	**Note**
P95	A5	Output	P95 = Setting not required	PM95 = Setting not required	PMC95 = 1	–	**Note**
P96	A6	Output	P96 = Setting not required	PM96 = Setting not required	PMC96 = 1	–	**Note**
P97	A7	Output	P97 = Setting not required	PM97 = Setting not required	PMC97 = 1	–	**Note**
P98	A8	Output	P98 = Setting not required	PM98 = Setting not required	PMC98 = 1	PFC98 = 0	**Note**
	TI030	Input	P98 = Setting not required	PM98 = Setting not required	PMC98 = 1	PFC98 = 1	–
P99	A9	Output	P99 = Setting not required	PM99 = Setting not required	PMC99 = 1	PFC99 = 0	**Note**
	TI031	Input	P99 = Setting not required	PM99 = Setting not required	PMC99 = 1	PFC99 = 1	–
P910	A10	Output	P910 = Setting not required	PM910 = Setting not required	PMC910 = 1	PFC910 = 0	**Note**
	TI020	Input	P910 = Setting not required	PM910 = Setting not required	PMC910 = 1	PFC910 = 1	–
P911	A11	Output	P911 = Setting not required	PM911 = Setting not required	PMC911 = 1	PFC911 = 0	**Note**
	TI021	Input	P911 = Setting not required	PM911 = Setting not required	PMC911 = 1	PFC911 = 1	–
P912	A12	Output	P912 = Setting not required	PM912 = Setting not required	PMC912 = 1	PFC912 = 0	**Note**
	TI010	Input	P912 = Setting not required	PM912 = Setting not required	PMC912 = 1	PFC912 = 1	–
P913	A13	Output	P913 = Setting not required	PM913 = Setting not required	PMC913 = 1	PFC913 = 0	**Note**
	TI011	Input	P913 = Setting not required	PM913 = Setting not required	PMC913 = 1	PFC913 = 1	–
P914	A14	Output	P914 = Setting not required	PM914 = Setting not required	PMC914 = 1	PFC914 = 0	**Note**
	TI000	Input	P914 = Setting not required	PM914 = Setting not required	PMC914 = 1	PFC914 = 1	–
P915	A15	Output	P915 = Setting not required	PM915 = Setting not required	PMC915 = 1	PFC915 = 0	**Note**
	TI001	Input	P915 = Setting not required	PM915 = Setting not required	PMC915 = 1	PFC915 = 1	–

Note To set the A0 to A15 pins, clear the PFC9 register to 0000H and set the PMC9 register to FFFFH in 16-bit units.

Table 4-15. Settings When Port Pins Are Used for Alternate Functions (4/5)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)
	Name	I/O					
PCM0	$\overline{\text{WAIT}}$	Input	PCM0 = Setting not required	PMCM0 = Setting not required	PMCCM0 = 1	–	–
PCM1	CLKOUT	Output	PCM1 = Setting not required	PMCM1 = Setting not required	PMCCM1 = 1	–	–
PCS0	$\overline{\text{CS0}}$	Output	PCS0 = Setting not required	PMCS0 = Setting not required	PMCCS0 = 1	–	–
PCS1	$\overline{\text{CS1}}$	Output	PCS1 = Setting not required	PMCS1 = Setting not required	PMCCS1 = 1	–	–
PCS2	$\overline{\text{CS2}}$	Output	PCS2 = Setting not required	PMCS2 = Setting not required	PMCCS2 = 1	–	–
PCT0	$\overline{\text{WR0}}$	Output	PCT0 = Setting not required	PMCT0 = Setting not required	PMCCT0 = 1	–	–
PCT1	$\overline{\text{WR1}}$	Output	PCT1 = Setting not required	PMCT1 = Setting not required	PMCCT1 = 1	–	–
PCT4	$\overline{\text{RD}}$	Output	PCT4 = Setting not required	PMCT4 = Setting not required	PMCCT4 = 1	–	–
PDH0	A16	Output	PDH0 = Setting not required	PMDH0 = Setting not required	PMCDH0 = 1	–	–
PDH1	A17	Output	PDH1 = Setting not required	PMDH1 = Setting not required	PMCDH1 = 1	–	–
PDH2	A18	Output	PDH2 = Setting not required	PMDH2 = Setting not required	PMCDH2 = 1	–	–

Table 4-15. Settings When Port Pins Are Used for Alternate Functions (5/5)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)
	Name	I/O					
PDL0	D0	I/O	PDL0 = Setting not required	PMDL0 = Setting not required	PMCDL0 = 1	–	–
PDL1	D1	I/O	PDL1 = Setting not required	PMDL1 = Setting not required	PMCDL1 = 1	–	–
PDL2	D2	I/O	PDL2 = Setting not required	PMDL2 = Setting not required	PMCDL2 = 1	–	–
PDL3	D3	I/O	PDL3 = Setting not required	PMDL3 = Setting not required	PMCDL3 = 1	–	–
PDL4	D4	I/O	PDL4 = Setting not required	PMDL4 = Setting not required	PMCDL4 = 1	–	–
PDL5	D5	I/O	PDL5 = Setting not required	PMDL5 = Setting not required	PMCDL5 = 1	–	–
PDL6	D6	I/O	PDL6 = Setting not required	PMDL6 = Setting not required	PMCDL6 = 1	–	–
PDL7	D7	I/O	PDL7 = Setting not required	PMDL7 = Setting not required	PMCDL7 = 1	–	–
PDL8	D8	I/O	PDL8 = Setting not required	PMDL8 = Setting not required	PMCDL8 = 1	–	–
PDL9	D9	I/O	PDL9 = Setting not required	PMDL9 = Setting not required	PMCDL9 = 1	–	–
PDL10	D10	I/O	PDL10 = Setting not required	PMDL10 = Setting not required	PMCDL10 = 1	–	–
PDL11	D11	I/O	PDL11 = Setting not required	PMDL11 = Setting not required	PMCDL11 = 1	–	–
PDL12	D12	I/O	PDL12 = Setting not required	PMDL12 = Setting not required	PMCDL12 = 1	–	–
PDL13	D13	I/O	PDL13 = Setting not required	PMDL13 = Setting not required	PMCDL13 = 1	–	–
PDL14	D14	I/O	PDL14 = Setting not required	PMDL14 = Setting not required	PMCDL14 = 1	–	–
PDL15	D15	I/O	PDL15 = Setting not required	PMDL15 = Setting not required	PMCDL15 = 1	–	–

★ 4.6 Cautions

4.6.1 Cautions on bit manipulation instruction for port n register (Pn)

When 1-bit manipulation instruction is executed to a port that includes both input and output pins, the value of the output latch of the input port not subject to the manipulation may be rewritten.

Therefore, it is recommended to rewrite the value of the output latch before switching a port from input mode to output mode.

<Example> When P90 is an output port, P91 to P97 are input ports (the status of all pins is high level), and the value of the port latch is 00H, if the output of output port 90 is changed from low level to high level by a bit manipulation instruction, the value of the port latch is FFH.

Explanation: The contents written to/read from the Pn register of a port whose PMnm bit is 1 are the output latch contents/pin status.

Bit manipulation instructions are executed in the following order in the V850ES/PM1.

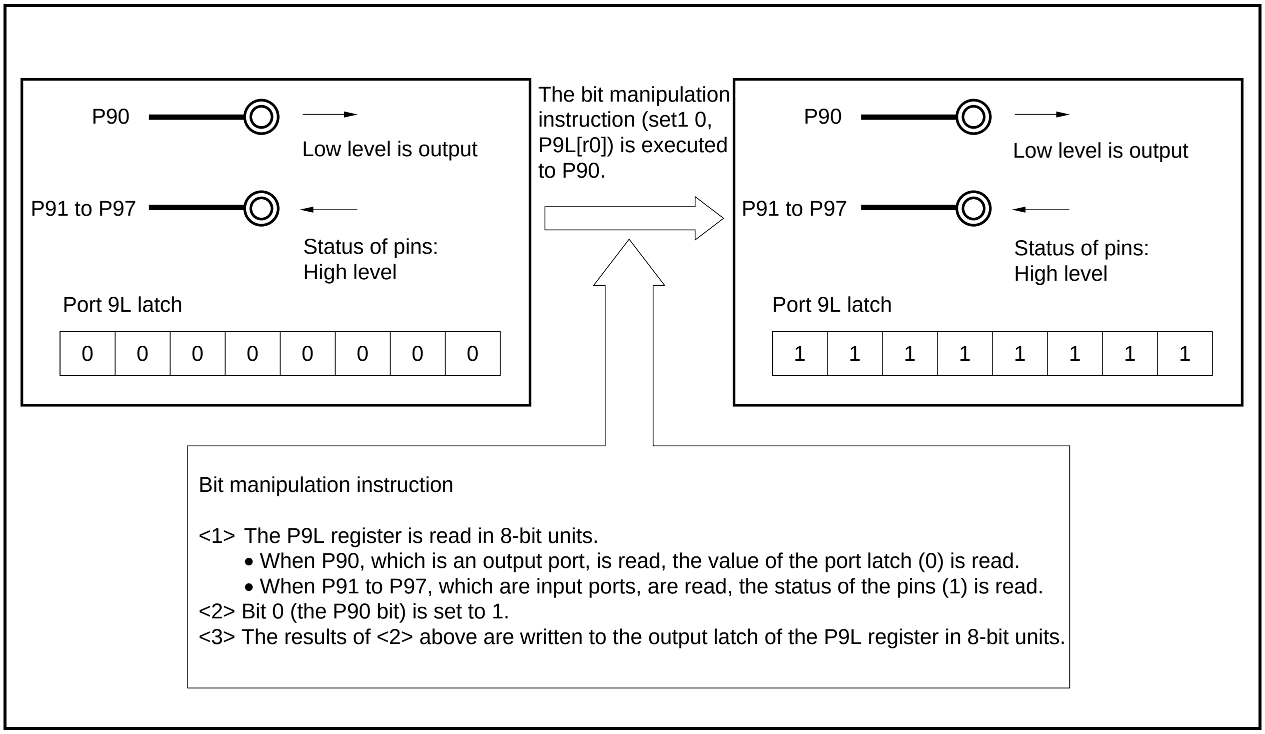
<1> The Pn register is read in 8-bit units.
<2> The targeted bit is manipulated.
<3> The Pn register is written in 8-bit units.

In <1> above, P90, an output port, reads the value of the output latch (0), while P91 to P97, input ports, read the statuses of pins. If the status of the pins P91 to P97 is high level at this time, the read value is FEH.

In <2> above, the value is FFH.

In <3> above, FFH is written to the output latch.

Figure 4-16. Bit Manipulation Instruction (P90)



Bit manipulation instruction

<1> The P9L register is read in 8-bit units.
- When P90, which is an output port, is read, the value of the port latch (0) is read.
- When P91 to P97, which are input ports, are read, the status of the pins (1) is read.

<2> Bit 0 (the P90 bit) is set to 1.
<3> The results of <2> above are written to the output latch of the P9L register in 8-bit units.

CHAPTER 5 BUS CONTROL FUNCTION

The V850ES/PM1 is provided with an external bus interface function by which external memories such as ROM and RAM, and I/O can be connected.

5.1 Features

○ Separate bus output with a minimum of 2 bus cycles
○ Three-space chip select function
○ 8-bit/16-bit data bus selectable (for each area selected by chip select function)
○ Wait function
- Programmable wait function of up to 7 states (selectable for each area selected by chip select function)
- External wait function using $\overline{\text{WAIT}}$ pin

○ Idle state function

5.2 Bus Control Pins

The pins used to connect an external device are listed in the table below.

Table 5-1. Bus Control Pins

Bus Control Pin	Alternate-Function Pin	I/O	Function
D0 to D15	PDL0 to PDL15	I/O	Data bus
A0 to A15	P90 to P915	Output	Address bus
A16 to A18	PDH0 to PDH2	Output	Address bus
$\overline{\text{WAIT}}$	PCM0	Input	External wait control
CLKOUT	PCM1	Output	Internal system clock
$\overline{\text{CS0}}$ to $\overline{\text{CS2}}$	PCS0 to PCS2	Output	Chip select
$\overline{\text{WR0}}$, $\overline{\text{WR1}}$	PCT0, PCT1	Output	Write strobe signal
$\overline{\text{RD}}$	PCT4	Output	Read strobe signal

★ 5.2.1 Pin status when internal ROM, internal RAM, or on-chip peripheral I/O is accessed

When the internal ROM, internal RAM, or on-chip peripheral I/O is accessed, the statuses of the pins are as follows.

Table 5-2. Pin Status When Internal ROM, Internal RAM, or On-Chip Peripheral I/O Is Accessed

Address bus (A18 to A0)	Undefined[Note]
Data bus (D15 to D0)	Hi-Z
Control signal	Inactive

Note The output data varies depending on the area to be accessed. When the internal ROM area or internal RAM area is accessed, the output value may different between the V850ES/PM1 and IE.

Caution When a write access to the internal ROM area is made, the address, data, and control signals are activated in the same way as the access to the external memory area.

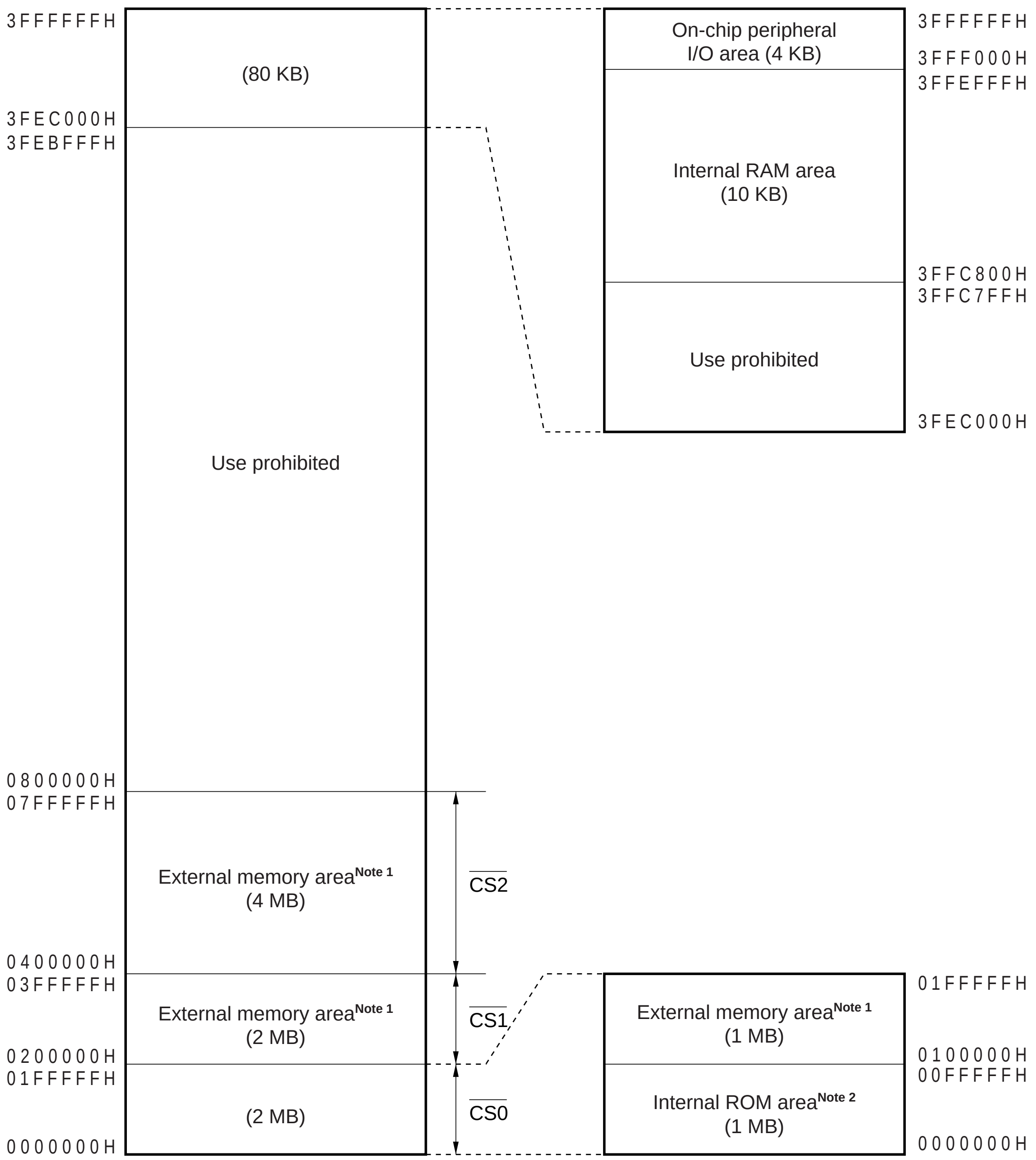
5.2.2 Pin status in each operation mode

For the pin status of the V850ES/PM1 in each operation mode, refer to **2.2 Pin Status**.

5.3 Memory Block Function

The 64 MB memory space is divided into memory blocks of (lower) 2 MB, 2 MB, and 4 MB. The programmable wait function and bus cycle operation mode for each of these blocks can be independently controlled in one-block units.

Figure 5-1. Data Memory Map: Physical Addresses



Notes 1. Each of these areas is a 512 KB space of 0100000H to 017FFFFH, 0200000H to 027FFFFH, or 0400000H to 047FFFFH (0180000H to 01FFFFFH, 0280000H to 03FFFFFH, and 0480000H to 07FFFFFH are an image).

2. This area is an external memory area in the case of a data write access.

5.3.1 Chip select control function

Of the 64 MB (linear) address space, the lower 8 MB (0000000H to 0FFFFFFH) include three chip select functions, $\overline{\text{CS0}}$ to $\overline{\text{CS2}}$. The areas that can be selected by $\overline{\text{CS0}}$ to $\overline{\text{CS2}}$ are fixed.

By using these chip select functions, the memory block can be divided to enable effective use of the memory space. The allocation of the memory blocks is shown in the table below.

	V850ES/PM1 (Single-Chip Mode)	V850ES/PM1 (ROMless Mode)
$\overline{\text{CS0}}$	0100000H to 017FFFFH (512 KB)	0000000H to 007FFFFH (512 KB)
$\overline{\text{CS1}}$	0200000H to 027FFFFH (512 KB)	0200000H to 027FFFFH (512 KB)
$\overline{\text{CS2}}$	0400000H to 047FFFFH (512 KB)	0400000H to 047FFFFH (512 KB)

5.4 Bus Access

5.4.1 Number of clocks for access

The following table shows the number of basic clocks required for accessing each resource.

Area (Bus Width) Bus Cycle Type	Internal ROM (32 Bits)	Internal RAM (32 Bits)	External Memory (16 Bits)
Instruction fetch (normal access)	1	1$^{\text{Note 1}}$	2 + n$^{\text{Note 2}}$
Instruction fetch (branch)	2	2$^{\text{Note 2}}$	2 + n$^{\text{Note 2}}$
Operand data access	3	1	2 + n$^{\text{Note 2}}$

Notes 1. When the access conflicts with a data access, the cycle is incremented by 1.

2. n: Number of wait states

Remark Unit: Clocks/access

5.4.2 Bus size setting function

The bus size of each external memory area selected by $\overline{\text{CSn}}$ can be set (to 8 bits or 16 bits) by using the BSC register.

The external memory area is selected by $\overline{\text{CS0}}$ to $\overline{\text{CS2}}$.

(1) Bus size configuration register (BSC)

The BSC register can be read or written in 16-bit units.

Caution Write to the BSC register after reset, and then do not change the set values. Also, do not access an external memory area until the initial settings of the BSC register are complete.

After reset: 5555H R/W Address: FFFFF066H

	15	14	13	12	11	10	9	8
BSC	0	1	0	1	0	1	0	1
	7	6	5	4	3	2	1	0
	0	1	0	BS20	0	BS10	0	BS00
$\overline{\text{CSn}}$ signal				$\overline{\text{CS2}}$		$\overline{\text{CS1}}$		$\overline{\text{CS0}}$

BSn0	Data bus width of CSn space (n = 0 to 2)
0	8 bits
1	16 bits

Caution Be sure to set bits 14, 12, 10, 8, and 6 to 1, and clear bits 15, 13, 11, 9, 7, 5, 3, and 1 to 0.

5.4.3 Access by bus size

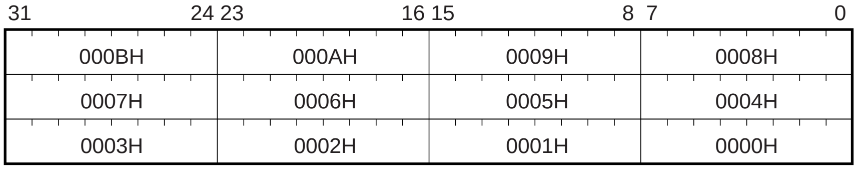
The V850ES/PM1 accesses the peripheral I/O and external memory in 8-bit, 16-bit, or 32-bit units. The bus size is as follows.

- The bus size of the peripheral I/O is fixed to 16 bits.
- The bus size of the external memory is selectable from 8 bits or 16 bits (by using the BSC register).

The operation when each of the above is accessed is described below. All data is accessed starting from the lower side.

The V850ES/PM1 supports only the little endian format.

Figure 5-2. Little Endian Address in Word



31 24	23 16	15 8	7 0
000BH	000AH	0009H	0008H
0007H	0006H	0005H	0004H
0003H	0002H	0001H	0000H

(1) Data space

The V850ES/PM1 has an address misalign function.

With this function, data can be placed at all addresses, regardless of the format of the data (word data or halfword data). However, if the word data or halfword data is not aligned at the boundary, a bus cycle is generated at least twice, causing the bus efficiency to drop.

(a) Halfword-length data access

A byte-length bus cycle is generated twice if the least significant bit of the address is 1.

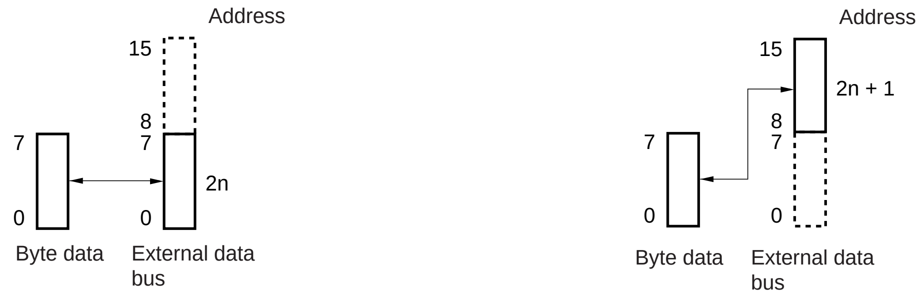
(b) Word-length data access

(i) A byte-length bus cycle, halfword-length bus cycle, and byte-length bus cycle are generated in that order if the least significant bit of the address is 1.

(ii) A halfword-length bus cycle is generated twice if the lower 2 bits of the address are 10.

(2) Byte access (8 bits)

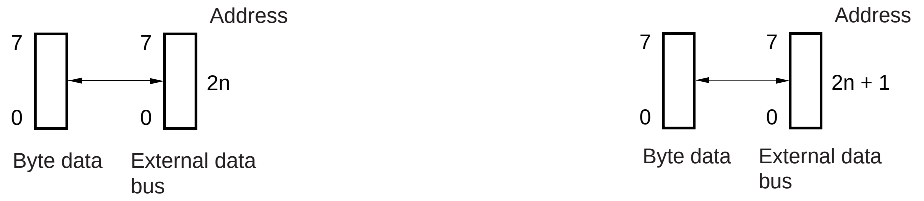
(a) 16-bit data bus width



<1> Access to even address (2n)

<2> Access to odd address (2n + 1)

(b) 8-bit data bus width

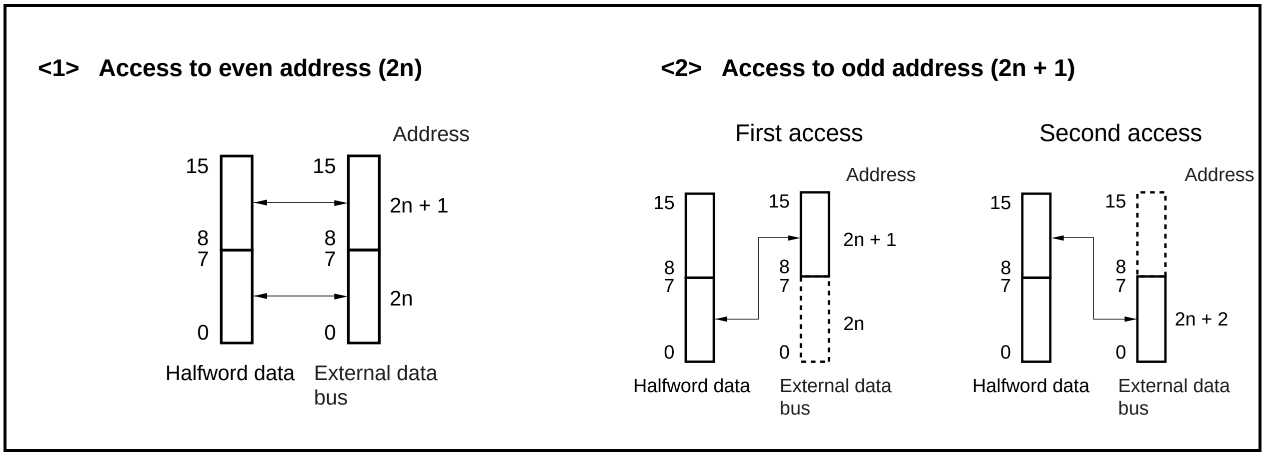


<1> Access to even address (2n)

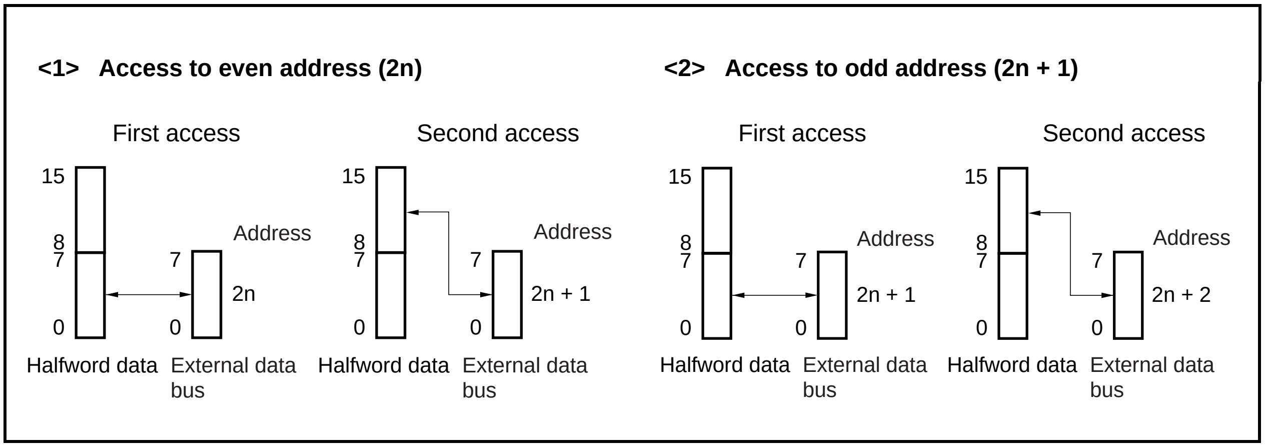
<2> Access to odd address (2n + 1)

(3) Halfword access (16 bits)

(a) 16-bit data bus width



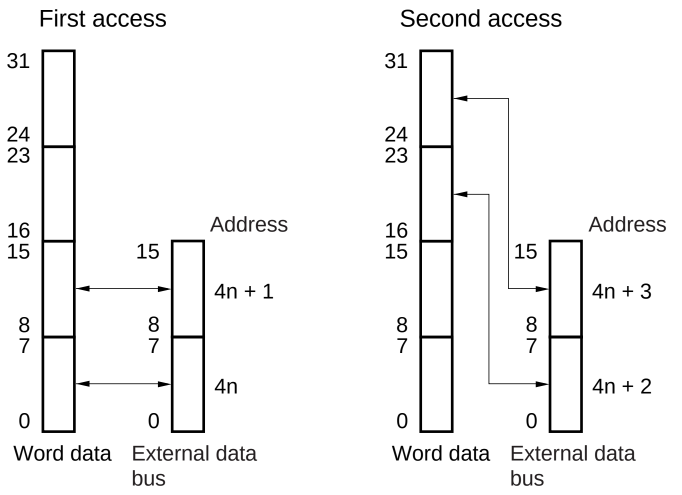
(b) 8-bit data bus width



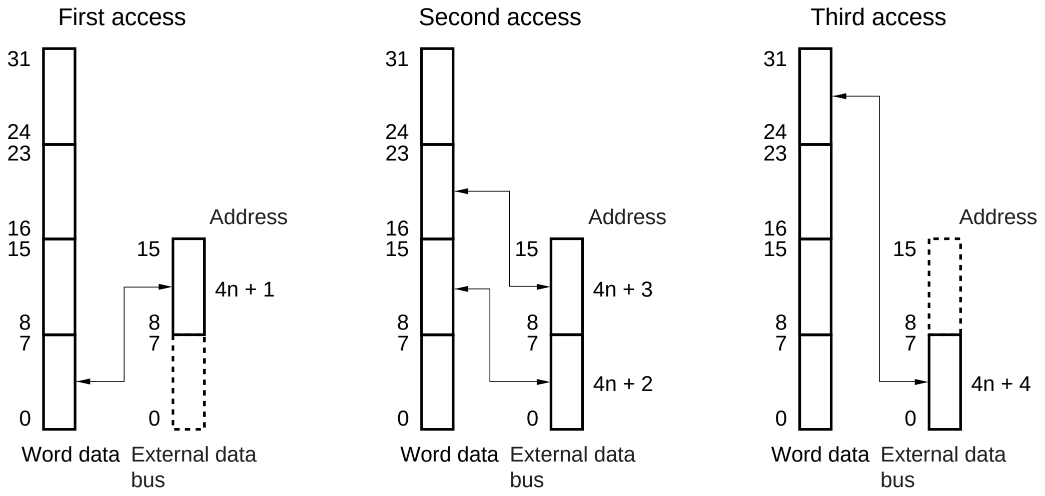
(4) Word access (32 bits)

(a) 16-bit data bus width (1/2)

<1> Access to address (4n)

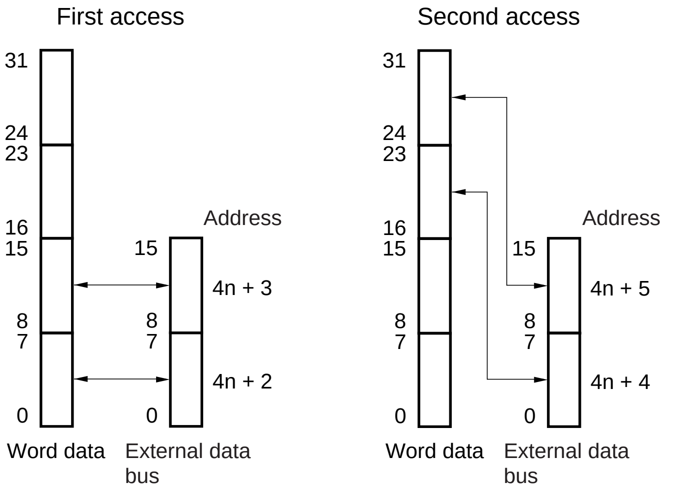


<2> Access to address (4n + 1)

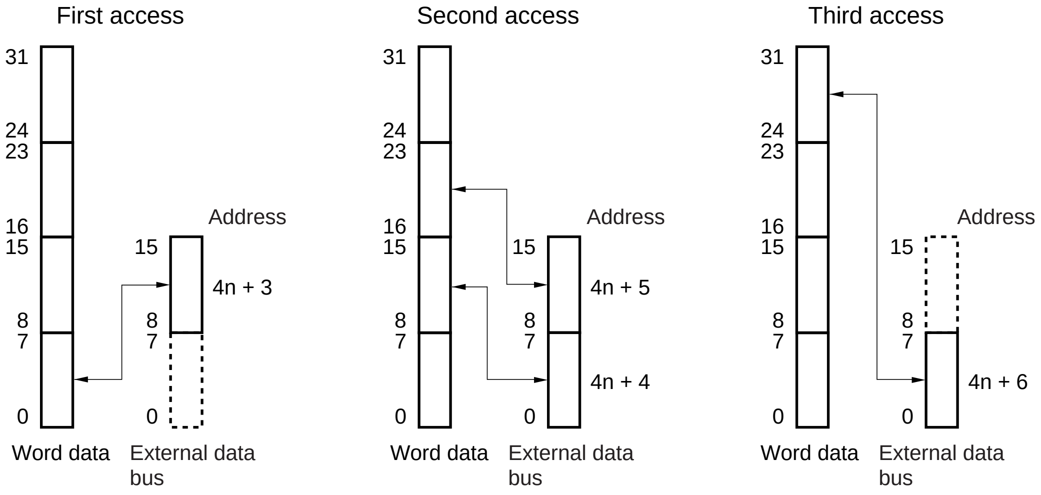


(a) 16-bit data bus width (2/2)

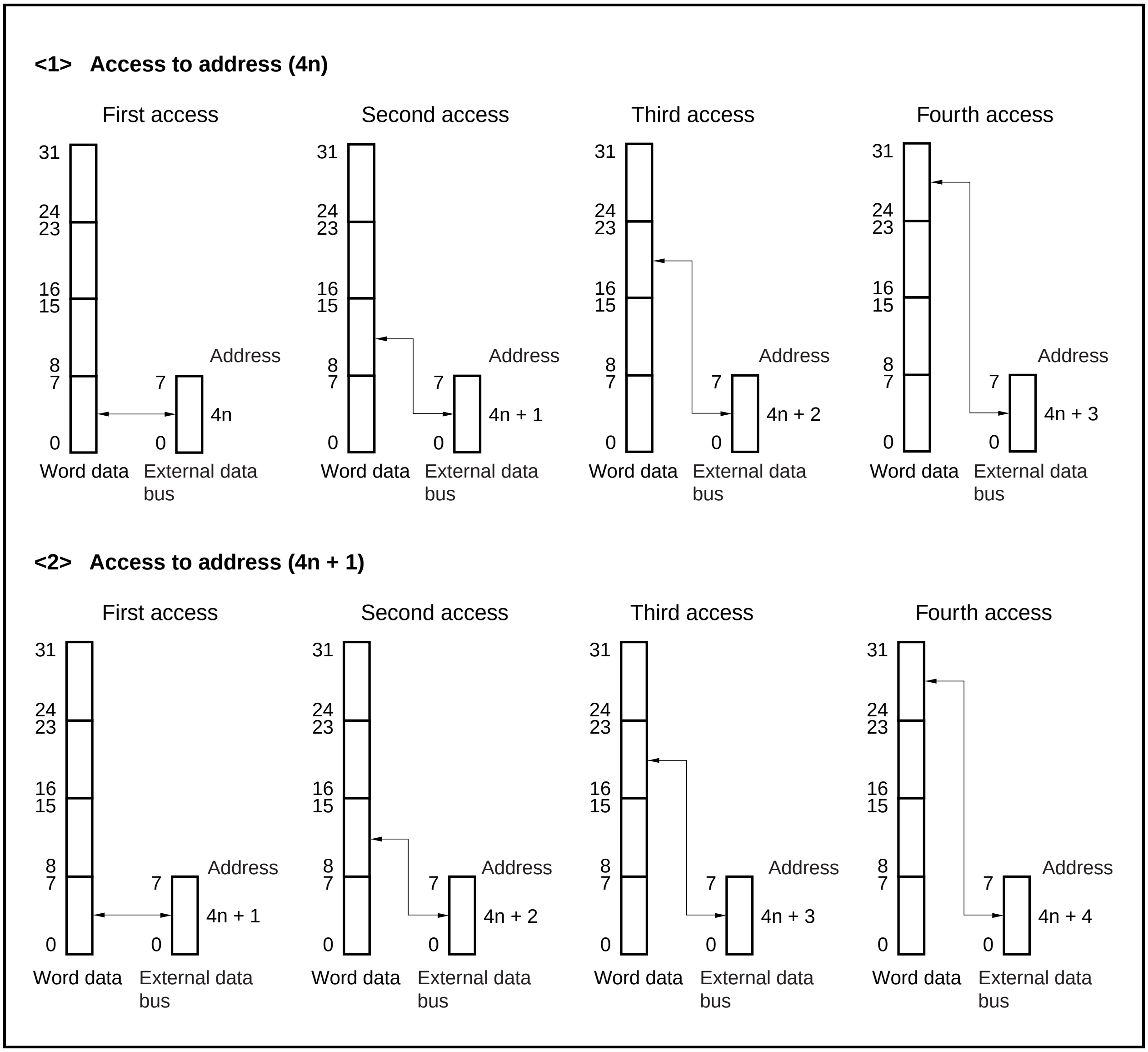
<3> Access to address (4n + 2)



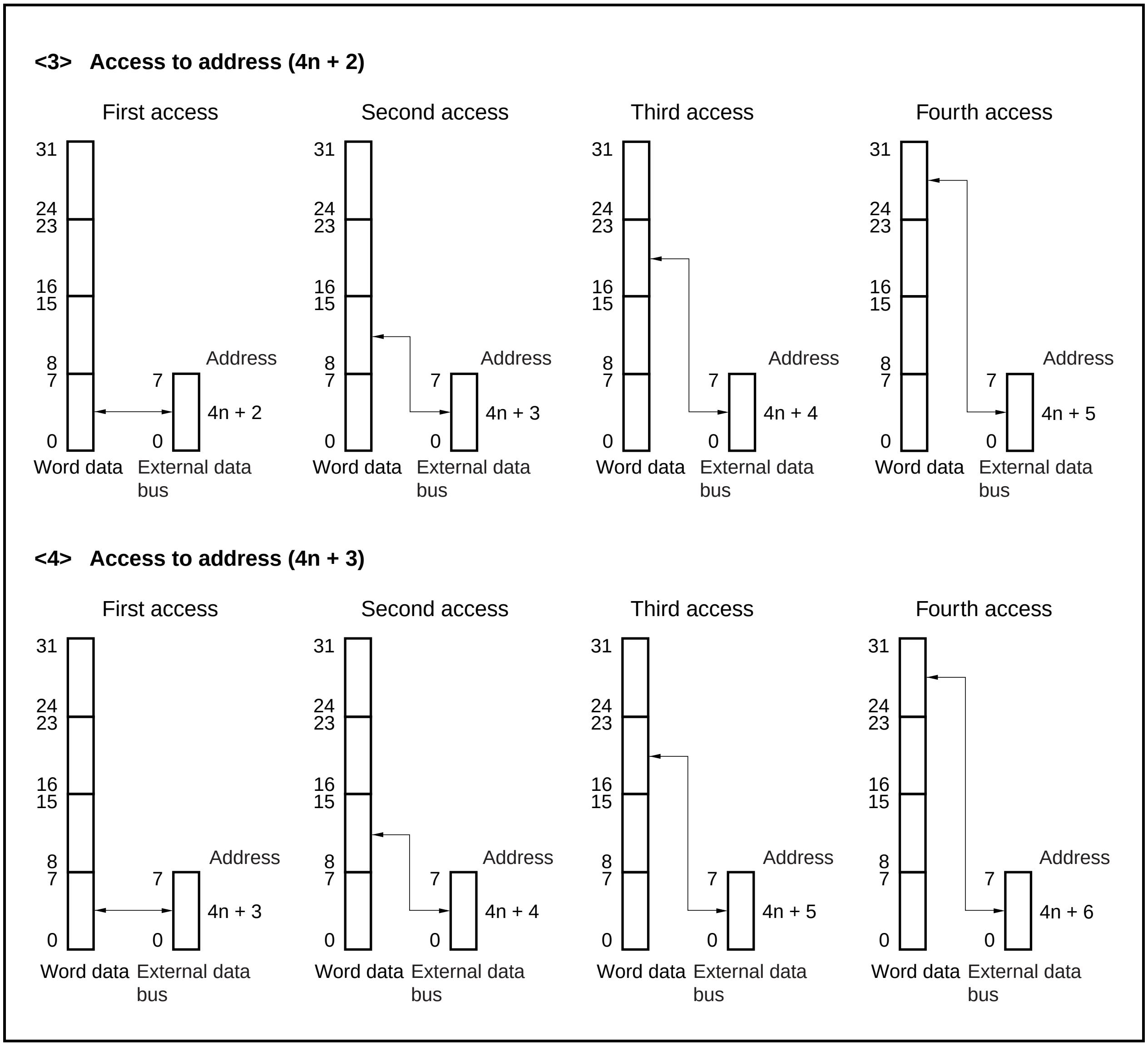
<4> Access to address (4n + 3)



(b) 8-bit data bus width (1/2)



(b) 8-bit data bus width (2/2)



<3> Access to address (4n + 2)

<4> Access to address (4n + 3)

5.5 Wait Function

5.5.1 Programmable wait function

(1) Data wait control register 0 (DWC0)

To realize interfacing with a low-speed memory or I/O, up to seven data wait states can be inserted in the bus cycle that is executed for each CS space.

The number of wait states can be programmed by using the DWC0 register. Immediately after system reset, 7 data wait states are inserted for all the blocks.

The DWC0 register can be read or written in 16-bit units.

This register is set to 7777H after reset.

Cautions 1. The internal ROM and internal RAM areas are not subject to programmable wait, and are always accessed without a wait state. The on-chip peripheral I/O area is also not subject to programmable wait, and only wait control from each peripheral function is performed.

2. Write to the DWC0 register after reset, and then do not change the set values. Also, do not access an external memory area until the initial settings of the DWC0 register are complete.

After reset: 7777H R/W Address: FFFFF484H

	15	14	13	12	11	10	9	8
DWC0	0	1	1	1	0	DW22	DW21	DW20
$\overline{\text{CSn}}$ signal						$\overline{\text{CS2}}$		

	7	6	5	4	3	2	1	0
	0	DW12	DW11	DW10	0	DW02	DW01	DW00
$\overline{\text{CSn}}$ signal		$\overline{\text{CS1}}$				$\overline{\text{CS0}}$		

DWn2	DWn1	DWn0	Number of wait states inserted in CSn space (n = 0 to 2)
0	0	0	None
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Caution Be sure to set bits 14, 13, and 12 to 1, and clear bits 15, 11, 7, and 3 to 0.

5.5.2 External wait function

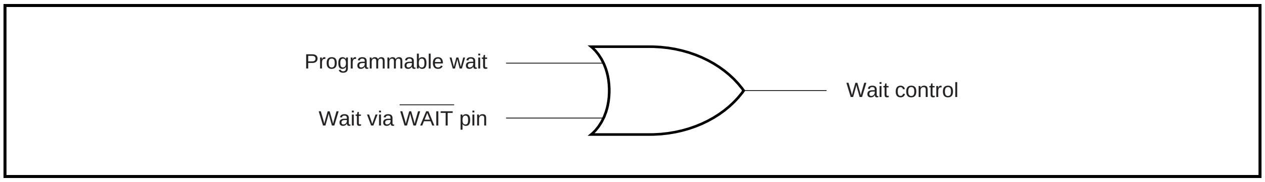
To synchronize an extremely slow external memory, I/O, or asynchronous system, any number of wait states can be inserted in the bus cycle by using the external wait pin ($\overline{\text{WAIT}}$).

Access to each area of the internal ROM, internal RAM, and on-chip peripheral I/O is not subject to control by the external wait function, in the same manner as the programmable wait function.

The $\overline{\text{WAIT}}$ signal can be input asynchronously to CLKOUT, and is sampled at the rising edge of the clock immediately after the T1 and TW states of the bus cycle. If the setup/hold time of the sampling timing is not satisfied, a wait state is inserted in the next state, or not inserted at all.

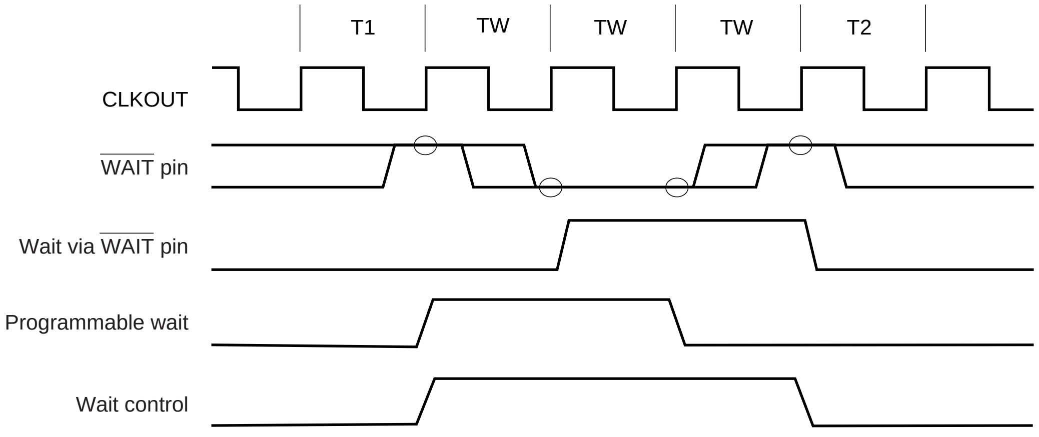
5.5.3 Relationship between programmable wait and external wait

Wait cycles are inserted as the result of an OR operation between the wait cycles specified by the set value of the programmable wait and the wait cycles controlled by the $\overline{\text{WAIT}}$ pin.



For example, if the timing of the programmable wait and the $\overline{\text{WAIT}}$ pin signal is as illustrated below, three wait states will be inserted in the bus cycle.

Figure 5-3. Example of Inserting Wait States



Remark The circles indicate the sampling timing.

5.5.4 Programmable address wait function

Address-setup or address-hold waits to be inserted in each bus cycle can be set by using the AWC register. Address wait insertion is set for each chip select area ($\overline{\text{CS0}}$ to $\overline{\text{CS2}}$).

If an address setup wait is inserted, it seems that the high-clock period of T1 state is extended by 1 clock. If an address hold wait is inserted, it seems that the low-clock period of T1 state is extended by 1 clock.

(1) Address wait control register (AWC)

The AWC register can be read or written in 16-bit units.

This register is set to FFFFH after reset.

★ **Cautions 1. The internal ROM, internal RAM, and on-chip peripheral I/O areas are not subject to idle state insertion.**

2. Write to the AWC register after reset, and then do not charge the set values. Also, do not access an external memory area until the initial settings of the AWC register are complete.

After reset: FFFFH R/W Address: FFFFF488H

	15	14	13	12	11	10	9	8
AWC	1	1	1	1	1	1	1	1

	7	6	5	4	3	2	1	0
	1	1	AHW2	ASW2	AHW1	ASW1	AHW0	ASW0
$\overline{\text{CSn}}$ signal			$\overline{\text{CS2}}$		$\overline{\text{CS1}}$		$\overline{\text{CS0}}$	

AHWn	Specifies insertion of address hold wait (n = 0 to 2)
0	Not inserted
1	Inserted

ASWn	Specifies insertion of address setup wait (n = 0 to 2)
0	Not inserted
1	Inserted

Caution Be sure to set bits 15 to 6 to 1.

5.6 Idle State Insertion Function

To facilitate interfacing with low-speed memories, one idle state (TI) can be inserted after the T2 state in the bus cycle that is executed for each space selected by the memory block function. By inserting an idle state, the data output float delay time of the memory can be secured during read access (an idle state cannot be inserted during write access).

Whether the idle state is to be inserted can be programmed by using the BCC register.

An idle state is inserted for all the areas immediately after system reset.

(1) Bus cycle control register (BCC)

The BCC register can be read or written in 16-bit units.

This register is set to AAAAH after reset.

Cautions 1. The internal ROM, internal RAM, and on-chip peripheral I/O areas are not subject to idle state insertion.

2. Write to the BCC register after reset, and then do not change the set values. Also, do not access an external memory area until the initial settings of the BCC register are complete.

After reset: AAAAH R/W Address: FFFFF48AH

	15	14	13	12	11	10	9	8
BCC	1	0	1	0	1	0	1	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1	0	BC21	0	BC11	0	BC01	0
$\overline{\text{CSn}}$ signal			$\overline{\text{CS2}}$		$\overline{\text{CS1}}$		$\overline{\text{CS0}}$	

BCn1	Specifies insertion of idle state (n = 0 to 2)
0	Not inserted
1	Inserted

Caution Be sure to set bits 15, 13, 11, 9, and 7 to 1, and clear bits 14, 12, 10, 8, 6, 4, 2, and 0 to 0.

5.7 Bus Priority

Instruction fetch (branch), instruction fetch (successive), and operand data accesses are executed in the external bus cycle.

Operand data access has the highest priority, followed by instruction fetch (branch) and instruction fetch (successive).

An instruction fetch may be inserted between the read access and write access in a read-modify-write access.

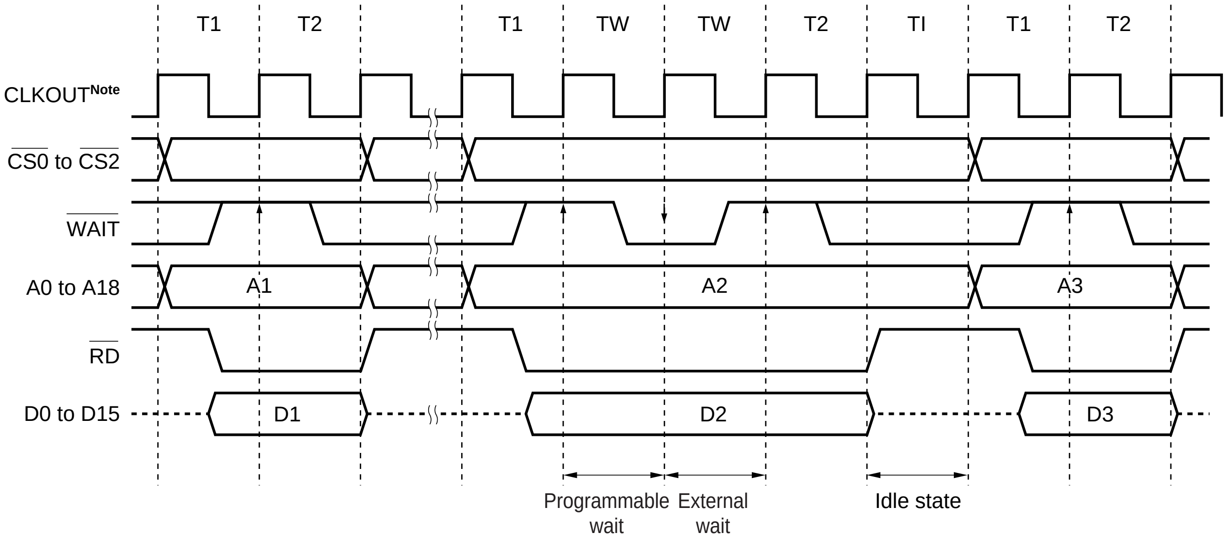
If an instruction is executed for two or more accesses, an instruction fetch is not inserted between accesses due to bus size limitations.

Table 5-3. Bus Priority

Priority	External Bus Cycle	Bus Master
High ↕ Low	Operand data access	CPU
	Instruction fetch (branch)	CPU
	Instruction fetch (successive)	CPU

5.8 Bus Timing

Figure 5-4. Bus Read Timing (Bus Size: 16 Bits, 16-Bit Access)



8-bit access	Odd address	Even address
D15 to D8	Active	Hi-Z
D7 to D0	Hi-Z	Active

Note CLKOUT is not output in the ROMless mode immediately after reset has been released. It is output in the control mode (PMCCM).

Figure 5-5. Bus Read Timing (Bus Size: 8 Bits)

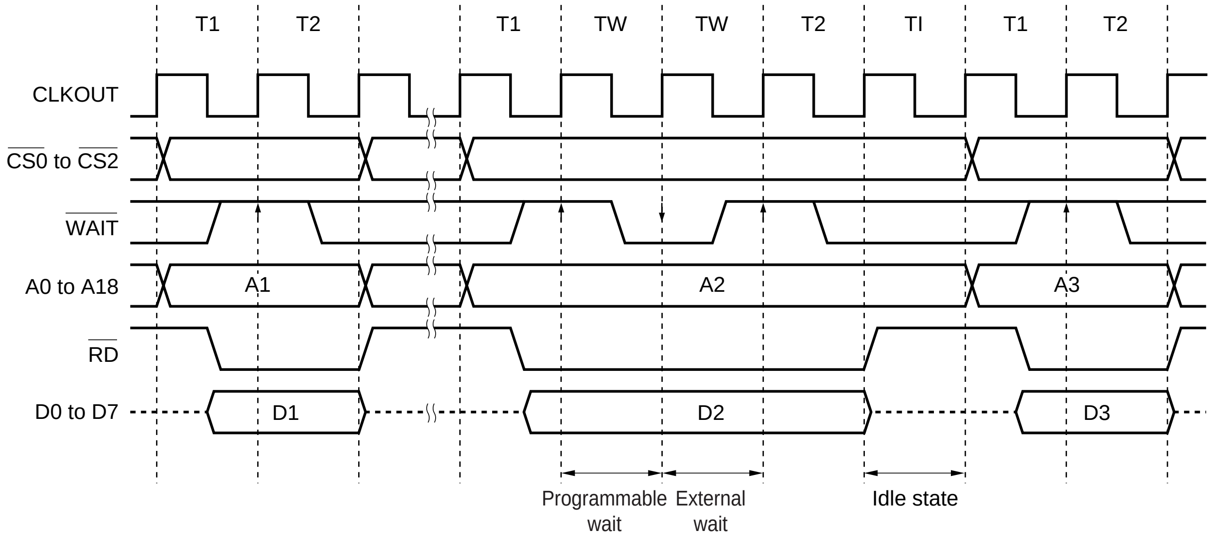
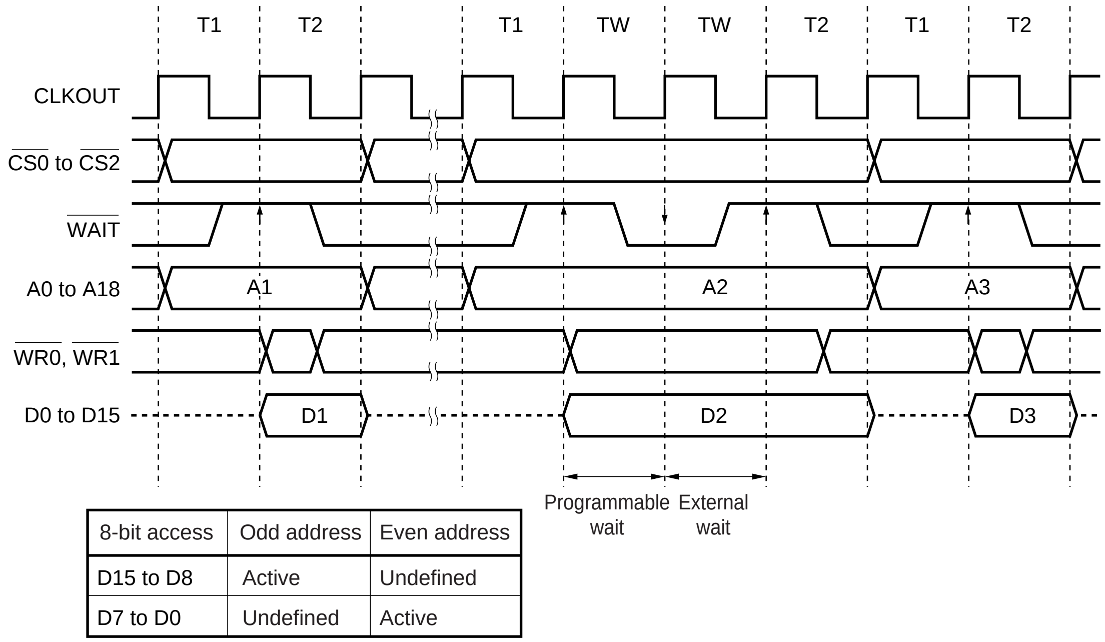


Figure 5-6. Bus Write Timing (Bus Size: 16 Bits, 16-Bit Access)



8-bit access	Odd address	Even address
D15 to D8	Active	Undefined
D7 to D0	Undefined	Active

Figure 5-7. Bus Write Timing (Bus Size: 8 Bits)

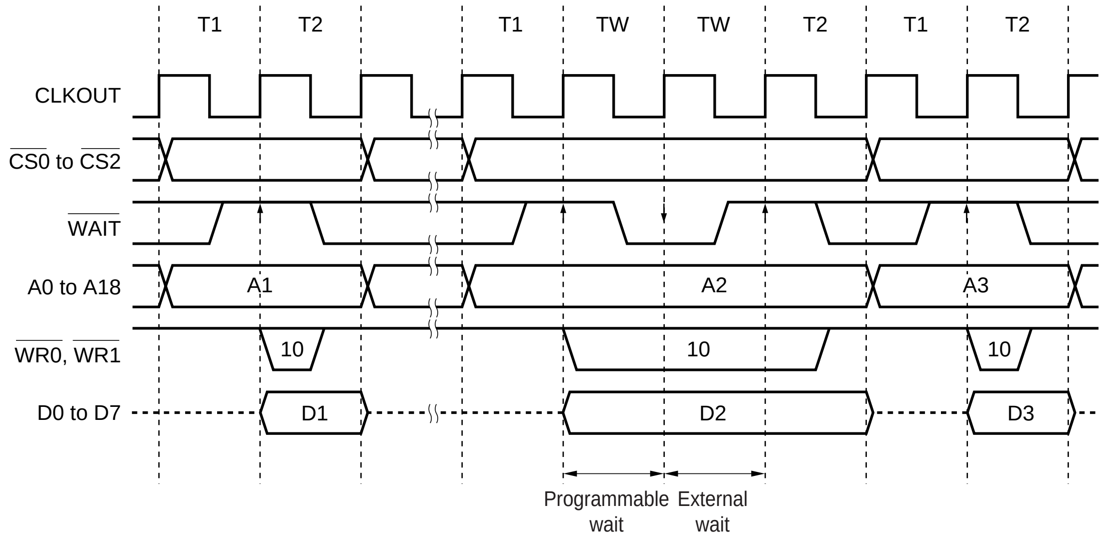
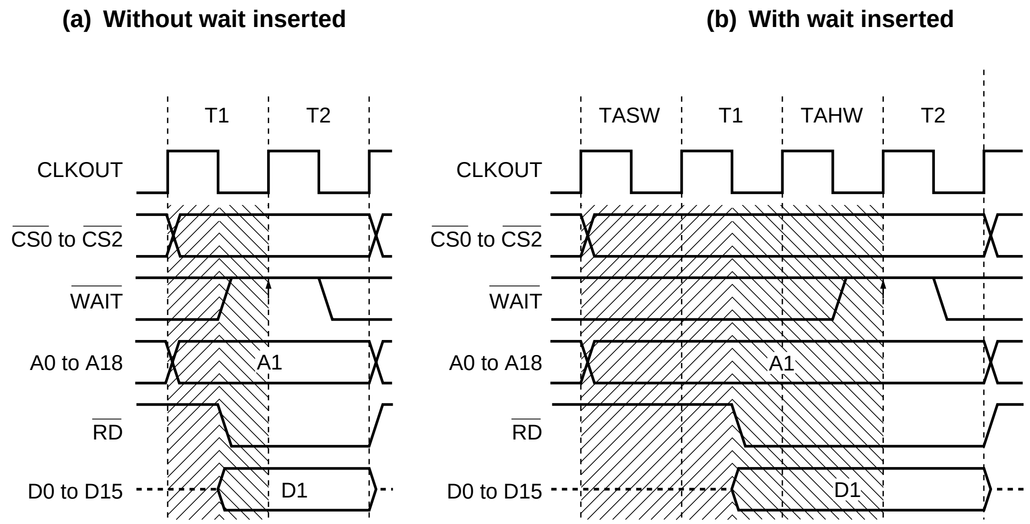


Figure 5-8. Address Wait Timing (Bus Read, Bus Size: 16 Bits, 16-Bit Access)



Remarks 1. TASW (address setup wait): Image of extended high-level width of T1 state

2. TAHW (address hold wait): Image of extended low-level width of T1 state

CHAPTER 6 CLOCK GENERATION FUNCTION

6.1 Overview

The features of the clock generation function are as follows.

○ Main clock oscillator
- f_X = 2 to 20 MHz (at 2.7 to 3.6 V operation)

○ Subclock oscillator
- f_{XT} = 32.768 kHz (at 2.2 to 3.6 V operation)

○ Generation of internal system clock and CPU clock
- Five steps (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, f_{XT})

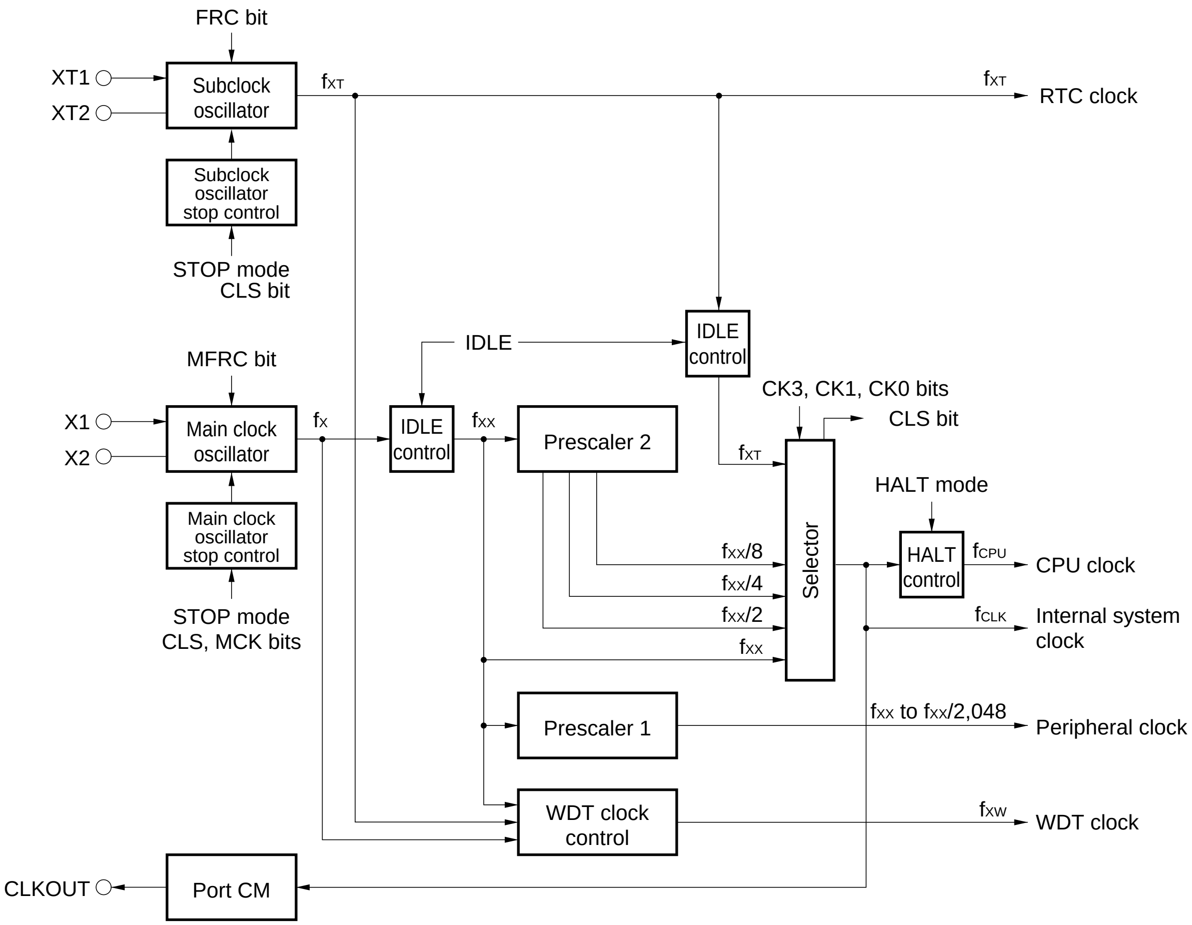
○ Generation of peripheral clock

○ Clock output function

Remark f_X: Main clock oscillation frequency
f_{XX}: Main clock frequency

6.2 Configuration

Figure 6-1. Clock Generator



Remark f_X: Main clock oscillation frequency
f_{XX}: Main clock frequency
f_{CLK}: Internal system clock frequency
f_{XT}: Subclock frequency
f_{CPU}: CPU clock frequency
f_{XW}: Watchdog timer clock frequency

(1) Main clock oscillator

This circuit oscillates the following frequency (f_X):

- 2 to 20 MHz (at 2.7 to 3.6 V operation)

(2) Subclock oscillator

This circuit oscillates a frequency of 32.768 kHz (f_{XT}).

(3) Main clock resonator stop control

This circuit generates a control signal that stops oscillation of the main clock resonator.

If the software STOP mode is set when the CLS bit of the processor clock control register (PCC) is 0, or if the MCK bit of the PCC register is set to 1 when the CLS bit is 1, oscillation of the main clock resonator is stopped.

(4) Subclock resonator stop control

This circuit generates a control signal that stops oscillation of the subclock resonator.

If the software STOP mode is set when the CLS bit is 1, oscillation of the subclock resonator is stopped.

(5) Prescaler 1

This circuit generates the clock (f_{XX} to f_{XX}/2,048) to be supplied to the on-chip peripheral functions.

The clock is supplied to the following blocks:

TM00 to TM03, TM10, TM11, TM20, TM21, CSI0, CSI1, UART0, UART1, PWM0 to PWM3, ADC

(6) Prescaler 2

This circuit divides the main clock (f_{XX}).

The clock generated by prescaler 2 (f_{XX} to f_{XX}/8) is supplied to the selector that generates the internal system clock (f_{CLK}).

f_{CLK} is the clock that is supplied to the CPU, INTC, and ROM correction blocks, and can be output from the CLKOUT pin.

(7) Watchdog timer clock control

This circuit divides the main clock oscillation frequency (f_X) by 16 to generate the clock to be supplied to the watchdog timer (f_{XW}). The watchdog timer is stopped when the subclock is used.

6.3 Registers

(1) Processor clock control register (PCC)

The processor clock control register (PCC) is a special register. Data can be written to it only in combination of specific sequences (refer to **3.4.7 Special registers**).

This register can be read or written in 8-bit or 1-bit units. The CLS bit is a read-only bit.

After reset: 03H   R/W   Address: FFFFF828H

	<7>	<6>	<5>	<4>	<3>	2	1	0
PCC	FRC	MCK	MFRC	CLS[Note]	CK3	0	CK1	CK0

FRC	Selects internal feedback resistor of subclock
0	Used
1	Not used

MCK	Controls main clock oscillator
0	Oscillation enabled
1	Oscillation stopped

- Even if the MCK bit is set to 1 while the system is operating with the main clock as the CPU clock, the operation of the main system clock does not stop. It stops after the CPU clock has been changed to the subclock.
- When the main clock is stopped and the device is operating on the subclock, clear the MCK bit to 0 and wait until the oscillation stabilization time has been secured by the program before switching back to the main clock.

MFRC	Selects internal feedback resistor of main clock
0	Used
1	Not used

CLS[Note]	Status of CPU clock (f_{CPU})
0	Main clock operation
1	Subclock operation

CK3	CK1	CK0	Selects clock (f_{CPU})
0	0	0	f_X
0	0	1	$f_{XX}/2$
0	1	0	$f_{XX}/4$
0	1	1	$f_{XX}/8$
1	X	X	f_{XT} (subclock: 32.768 kHz)

Note The CLS bit is a read-only bit.

Cautions 1. Do not change the CPU clock (by using the CK3, CK1, and CK0 bits of the PCC register) while CLKOUT is being output.

2. Be sure to clear bit 2 to 0.

★ **3. When the CPU is operating on the subclock and when a clock is not input to X1 or the main oscillator is stopped, do not access the registers that cause a wait. If a wait is generated, only a reset can release the wait.**

Remark X: Don’t care

★ (a) Example of setting main clock operation → subclock operation

<1> CK3 bit ← 1: A bit manipulation instruction is recommended. Do not change the CK1 and CK0 bits.

<2> Subclock operation: Read the CLS bit to confirm whether the operation has switched to the subclock. The following is the time required for switching to subclock operation after the CK3 bit is set.
Maximum: 1/f_{XT} (1/subclock frequency)

<3> MCK bit ← 1: Set the MCK bit to 1 only when stopping the main clock.

[Description example]

```
<1> _SET_SUB_RUN :
    st.b      r0, prcmd[r0]
    set1      3, PCC[r0]                -- CK3 bit ← 1
<2> _CHECK_CLS :
    tst1       4, PCC[r0]               -- CK3 bit ← Wait until the mode is changed
    bz         _CHECK_CLS                              to subclock operation
<3> _STOP_MAIN_CLOCK :
    st.b      r0, PRCMD[r0]
    set1      6, PCC[r0]                -- MCK bit ← 1, the main clock stops
```

★ **(b) Example of setting subclock operation → main clock operation**

<1> MCK bit ← 0: Main clock oscillation starts

<2> Insert waits by program and wait until the oscillation stabilization time of the main clock has elapsed.

<3> CK3 bit ← 0: A bit manipulation instruction is recommended. Do not change the CK1 and CK0 bits.

<4> Main clock operation: After the CK3 bit has been set, it takes the following time until the CPU starts operating on the main clock.

Maximum: 1/f_{XT} (1/subclock frequency)

Therefore, insert one NOP instruction immediately after the CK3 bit is set to 0 or read the CLS bit to confirm that the CPU has started to operate on the main clock.

[Description example]

```
<1> _START_MAIN_OSC :
    st.b       r0, PRCMD[r0]                  -- Release protection of special register
    clr1       6, PCC[r0]                     -- Main clock oscillation starts
<2> movea      0x55, r0, r11                  -- Wait for oscillation stabilization time
    _WAIT_OST :
     nop
     nop
     nop
     addi      -1, r11, r11
     mp        r0, r11
     bne       _PROGRAM_WAIT
<3> st.b      r0, PRCMD[r0]
     clr1      r0, PRCMD[r0]                  -- CK3 bit ← 1
<4> _CHECK_CLS :
     tst1      4, PCC[r0]                     -- Wait until CPU starts to operate on main clock
     bns       _CHECK_CLS
```

6.4 Operation

6.4.1 Operation of each clock

The following table shows the operation status of each clock.

Table 6-1. Operation Status of Each Clock

PLL Register Setting and Operation Status / Clock	CLS Bit = 0 MCK Bit = 0					CLS Bit = 1 MCK Bit = 0		CLS Bit = 1 MCK Bit = 1		Sub-Software STOP Mode
	During Reset	During Oscillation Stabilization Time Count	HALT Mode	IDLE Mode	Software STOP Mode	Subclock Mode	Sub-IDLE Mode	Subclock Mode	Sub-IDLE Mode	
Main resonator (f_X)	×	√	√	√	×	√	√	×	×	×
Sub-resonator (f_{XT})	√	√	√	√	√	√	√	√	√	×
CPU clock (f_{CPU})	×	×	×	×	×	√	×	√	×	×
Internal system clock (f_{CLK})	×	×	√	×	×	√	×	√	×	×
Peripheral clock (f_X to $f_X/512$)	×	×	√	×	×	√	×	×	×	×
WDT clock (f_{XW})	×	√	√	×	×	**Note**	×	×	×	×
RTC clock (f_{XT})	×	√	√	√	√	√	√	√	√	×

Note The watchdog timer clock (f_{XW}) is operable but it stops operating in the watchdog timer if the CLS bit is set to 1.

Remark √: Operable
×: Stops

6.4.2 Clock output function

The clock output function allows the CLKOUT pin to output the internal system clock (f_{CLK}).

The internal system clock (f_{CLK}) is selected by using the PCC.CK3, PCC.CK1, and PCC.CK0 bits.

The CLKOUT pin functions alternately as the PCM1 pin and operates as a clock output pin when the PMCCM register is set (refer to **4.3.7 Port CM**).

The status of the CLKOUT pin is the same as the internal system clock in Table 6-1, and can output the clock when it is √ (operable). When it is × (stops), it outputs a low level. However, the port mode (PCM1: input mode) is selected until the CLKOUT pin is set after reset. Consequently, the pin goes into a high-impedance state.

★ 6.4.3 External clock input function

An external clock can be directly input to the oscillator. Input the clock to the X1 pin and its inverse signal to the X2 pin. Set the MFRC bit of the PCC register to 1 (on-chip feedback resistor not used). Note, however, that oscillation stabilization time is inserted even in the external clock mode. Connect V_{DD} directly to the REGC pin.

CHAPTER 7 16-BIT TIMER/EVENT COUNTERS 00 TO 03

7.1 Functions

16-bit timer/event counters 00 to 03 have the following functions.

(1) Interval timer
Generates an interrupt at predetermined time intervals.

(2) PPG output
Can output a rectangular wave with any frequency and any output pulse width.

(3) Pulse width measurement
Can measure the pulse width of a signal input from an external source.

(4) External event counter
Can measure the number of pulses of a signal input from an external source.

(5) Square-wave output
Can output a square wave of any frequency.

(6) One-shot pulse output
Can output a one-shot pulse with any output pulse width.

7.2 Configuration

16-bit timer/event counters 00 to 03 consist of the following hardware.

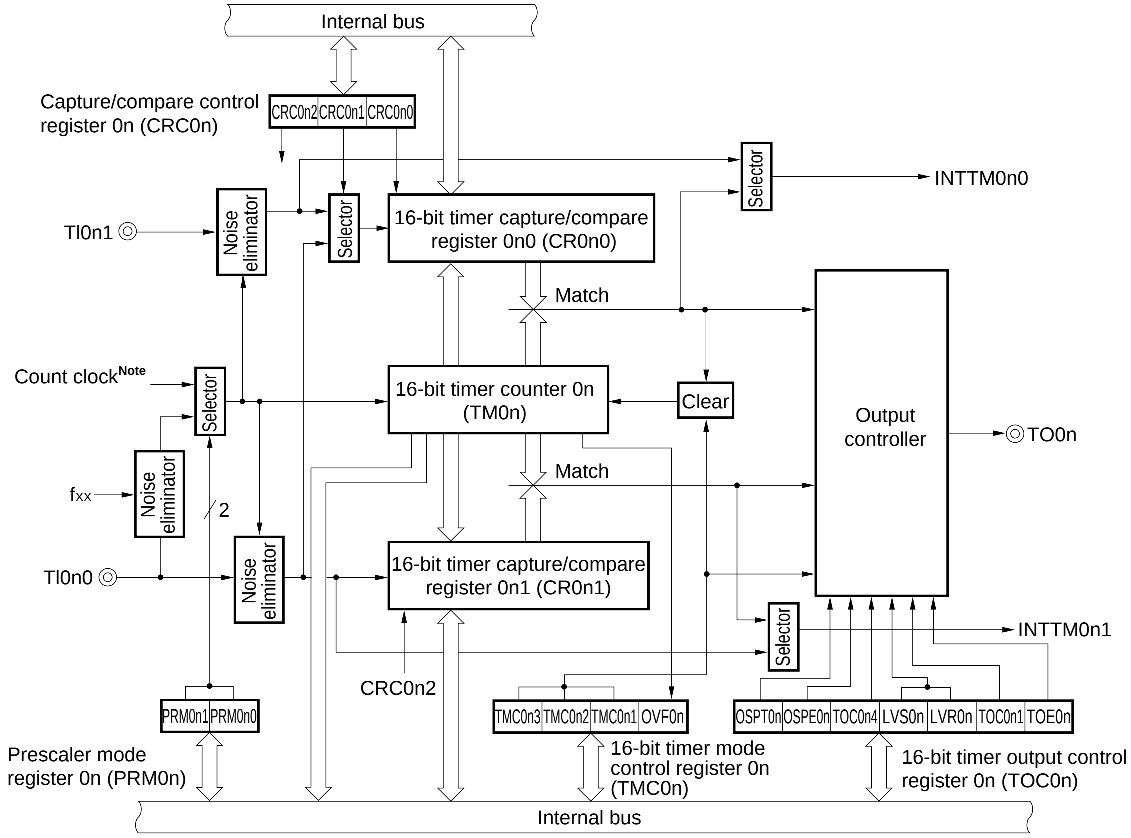
Table 7-1. Configuration of 16-Bit Timer/Event Counters 00 to 03

Item	Configuration
Timer/counters	16 bits × 1 (TM0n)
Registers	16-bit timer capture/compare register: 16 bits × 2 (CR0n0, CR0n1)
Timer inputs	2 (TI0n0, TI0n1)
Timer outputs	1 (TO0n), output controller
Control registers	16-bit timer mode control register n (TMC0n) Capture/compare control register n (CRC0n) 16-bit timer output control register n (TOC0n) Prescaler mode register 0n (PRM0n)

Remark n = 0 to 3

Figure 7-1 shows the block diagram.

Figure 7-1. Block Diagram of 16-Bit Timer/Event Counter 0n


Internal bus
Capture/compare control register 0n (CRC0n)
CRC0n2
CRC0n1
CRC0n0
Selector
INTTM0n0
TI0n1
Noise eliminator
Selector
16-bit timer capture/compare register 0n0 (CR0n0)
Match
Output controller
Count clockNote
Selector
16-bit timer counter 0n (TM0n)
Clear
TO0n
fXX
Noise eliminator
2
Match
TI0n0
Noise eliminator
16-bit timer capture/compare register 0n1 (CR0n1)
Selector
INTTM0n1
CRC0n2
PRM0n1
PRM0n0
TMC0n3
TMC0n2
TMC0n1
OVF0n
OSPT0n
OSPE0n
TOC0n4
LVS0n
LVR0n
TOC0n1
TOE0n
Prescaler mode register 0n (PRM0n)
16-bit timer mode control register 0n (TMC0n)
16-bit timer output control register 0n (TOC0n)
Internal bus

Note Set by the PRM0n register.

Remarks 1. n = 0 to 3

2. fXX: Main clock frequency

(1) 16-bit timer counter 0n (TM0n)

The TM0n register is a 16-bit read-only register that counts count pulses. The counter is incremented in synchronization with the rising edge of the input clock.

After reset: 0000H R Address: TM00 FFFFF5C0H, TM01 FFFFF5D0H,
TM02 FFFFF5E0H, TM03 FFFFF5F0H,

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TM0n (n = 0 to 3)																

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

<1> Reset
<2> If the TMC0n.TMC0n3 and TMC0n.TMC0n2 bits are cleared.
<3> If the valid edge of TI0n0 is input in the mode in which clear & start occurs when inputting the valid edge of TI0n0
<4> If the TM0n register and the CR0n0 register match each other in the mode in which clear & start occurs on CR0n0 register match
<5> If the TOC0n.OSPT0n bit is set (1) or if the valid edge of TI0n0 is input in the one-shot pulse output mode

Remark n = 0 to 3

(2) 16-bit timer capture/compare register 0n0 (CR0n0)

The CR0n0 register is a 16-bit register that combines capture register and compare register functions.

The CRC0n.CRC0n0 bit is used to set whether to use the CR0n0 register as a capture register or as a compare register.

The CR0n0 register is set by a 16-bit memory manipulation instruction.

This register is set to 0000H after reset.

After reset: 0000H R/W Address: CR000 FFFFF5C2H, CR010 FFFFF5D2H,
CR020 FFFFF5E2H, CR030 FFFFF5F2H,

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR0n0 (n = 0 to 3)																

(a) When using the CR0n0 register as a compare register

The value set to the CR0n0 register and the count value set to the TM0n register are always compared and when these values match, an interrupt request (INTTM0n0) is generated. The values are retained until rewritten.

(b) When using the CR0n0 register as a capture register

The TM0n register count value is captured to the CP0n0 register by inputting a capture trigger. The valid edge of the TI0n0 pin or TI0n1 pin can be selected as the capture trigger. The valid edge of the TI0n0 pin or TI0n1 pin is set using the PRM0n.ES0n1 and PRM0n.ES0n0 bits or PRM0n.ES1n1 and PRM0n.ES1n1 bits.

Table 7-2 shows the settings when the valid edge of the TI0n0 pin is specified as the capture trigger, and Table 7-3 shows the settings when the valid edge of the TI0n1 is specified as the capture trigger.

Table 7-2. Capture Trigger of CR0n0 Register and Valid Edge of TI0n0 Pin

Capture Trigger of CR0n0 Register	Valid Edge of TI0n0 Pin		
		ES0n1	ES0n0
Falling edge	Rising edge	0	1
Rising edge	Falling edge	0	0
No capture operation	Both rising and falling edges	1	1

Remarks 1. n = 0 to 3

2. Setting the ES0n1 and ES0n0 bits to 10 is prohibited.

Table 7-3. Capture Trigger of CR0n0 Register and Valid Edge of TI0n1 Pin

Capture Trigger of CR0n0 Register	Valid Edge of TI0n1 Pin		
		ES1n1	ES1n0
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. n = 0 to 3

2. Setting the ES1n1 and ES1n0 bits to 10 is prohibited.

Cautions 1. Set a value other than 0000H to the CR0n0 register in the mode in which clear & start occurs upon a match of the values of the TM0n register and CR0n0 register. However, if 0000H is set to the CR0n0 register in the free-running mode or the TI0n0 valid edge clear mode, an interrupt request (INTTM0n0) is generated when the value changes from 0000H to 0001H after an overflow (FFFFH).

2. If, when the CR0n0 register is used as a capture register, the register read interval and capture trigger input conflict, the read data becomes undefined (but the capture data itself is normal). Moreover, when the count stop input and capture trigger input conflict, the capture data becomes undefined.

3. Do not rewrite the value of the CR0n0 register during TM0n register operation.

(3) 16-bit timer capture/compare register 0n1 (CR0n1)

The CR0n1 register is a 16-bit register that combines capture register and compare register functions. The CRC0n.CRC0n2 bit is used to set whether to use the CR0n1 register as a capture register or as a compare register.

The CR0n1 register is set by a 16-bit memory manipulation instruction.

This register is set to 0000H after reset.

After reset: 0000H R/W Address: CR001 FFFFF5C4H, CR011 FFFFF5D4H,
CR021 FFFFF5E4H, CR031 FFFFF5F4H,

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR0n1 (n = 0 to 3)																

(a) When using the CR0n1 register as a compare register

The value set to the CR0n1 register and the count value of the TM0n register are always compared and when these values match, an interrupt request (INTTM0n1) is generated.

(b) When using the CR0n1 register as a capture register

The TM0n register count value is captured to the CR0n1 register by inputting a capture trigger.

The valid edge of the TI0n0 pin can be selected as the capture trigger. The valid edge of the TI0n0 is set with the PRM0n register.

Table 7-4 shows the settings when the valid edge of the TI0n0 pin is specified as the capture trigger.

Table 7-4. Capture Trigger of CR0n1 Register and Valid Edge of TI0n0 Pin

Capture Trigger of CR0n1 Register	Valid Edge of TI0n0 Pin		
		ES0n1	ES0n0
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. n = 0 to 3

2. Setting the ES0n1 and ES0n0 bits to 10 is prohibited.

Cautions 1. When the P11 and P12 pins are used as the PWM1 and PWM2 output pins, they cannot be used as timer output pins (TO00, TO01).

★ **2. If 0000H is set to the CR0n1 register, an interrupt request (INTTM0n1) is generated after an overflow of the TM0n register, after clear & start on a match between the TM0n register and CR0n0 register, after clear by the TI0n0 valid edge, or after clear by a one-shot trigger.**

3. If, when the CR0n1 register is used as a capture register, the register read interval and capture trigger input conflict, the read data becomes undefined (but the capture data itself is normal). Moreover, when the count stop input and capture trigger input conflict, the capture data becomes undefined.

4. The value of the CR0n1 register can be changed while the TM0n register is operating. For details, refer to 7.4.2 PPG output operation.

5. The TI0n0 and TI0n1 pins function alternately as the P98/A8 to P915/A15 pins. To use the TI0n0 and TI0n1 pins, select the timer input function by using the PMC9m and PFC9m bits, before enabling the timer operation with the TMC0n register. If the PMC9m and PFC9m bits are manipulated after the timer starts operating, the edge cannot be detected correctly.

Remark n = 0 to 3, m = 8 to 15

7.3 Registers

The registers that control 16-bit timer/event counters 00 to 03 are as follows.

- 16-bit timer mode control register 0n (TMC0n)
- Capture/compare control register 0n (CRC0n)
- 16-bit timer output control register 0n (TOC0n)
- Prescaler mode register 0n (PRM0n)

(1) 16-bit timer mode control register 0n (TMC0n)

TMC0n is used to set the 16-bit timer operation mode, TM0n clear mode, and the output timing, and to detect an overflow.

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

This register is set to 00H after reset.

Caution The TM0n register starts operating when a value other than 00 (operation stop mode) is set to the TMC0n.TMC0n3 and TMC0n.TMC0n2 bits. To stop the operation, set 00 to the TMC0n3 and TMC0n2 bits.

Remark n = 0 to 3

After reset: 00H R/W Address: TMC00 FFFFF5C6H, TMC01 FFFFF5D6H,
TMC02 FFFFF5E6H, TMC03 FFFFF5F6H

	7	6	5	4	3	2	1	<0>
TMC0n	0	0	0	0	TMC0n3	TMC0n2	TMC0n1	OVF0n

(n = 0 to 3)

TMC0n3	TMC0n2	TMC0n1	Selection of operation mode and clear mode	Selection of TO0n output inversion timing	Generation of interrupt
0	0	0	Operation stop (TM0n cleared to 0)	Unchanged	Not generated
0	0	1			
0	1	0	Free-running mode	Match of TM0n and CR0n0 or match of TM0n and CR0n1	Generated upon match of TM0n and CR0n0 and match of TM0n and CR0n1, or generated when valid edges of TI0n0 and TI0n1 are generated
0	1	1		Match of TM0n and CR0n0, match of TM0n and CR0n1, or valid edge of TI0n0	
1	0	0	Clear & start with valid edge of TI0n0	Match of TM0n and CR0n0 or match of TM0n and CR0n1	
1	0	1		Match of TM0n and CR0n0, match of TM0n and CR0n1, or valid edge of TI0n0	
1	1	0	Clear & start upon match of TM0n and CR0n0	Match of TM0n and CR0n0 or match of TM0n and CR0n1	
1	1	1		Match of TM0n and CR0n0, match of TM0n and CR0n1, or valid edge of TI0n0	

OVF0n	Detection of overflow of TM0n register
0	No overflow
1	Overflow

Cautions 1. Write to bits other than the OVF0n flag after stopping the timer operation.

2. The valid edge of the TI0n0 pin is set by the PRM0n register.

3. When the mode in which the timer is cleared and started upon match of TM0n and CR0n0 is selected, the setting value of CR0n0 is FFFFH, and when the value of TM0n changes from FFFFH to 0000H, the OVF0n flag is set to 1.

Remark		
	TO0n:	Output pin of 16-bit timer/event counter 0n
	TI0n0:	Input pin of 16-bit timer/event counter 0n
	TM0n:	16-bit timer counter 0n
	CR0n0:	16-bit timer capture/compare register 0n0
	CR0n1:	16-bit timer capture/compare register 0n1

(2) Capture/compare control register 0n (CRC0n)

CRC0n controls the operation of the CR0n0 and CR0n1 registers.

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

After reset, this register is set to 00H.

After reset: 00H R/W Address: CRC00 FFFFF5C8H, CRC01 FFFFF5D8H,
CRC02 FFFFF5E8H, CRC03 FFFFF5F8H

	7	6	5	4	3	2	1	0
CRC0n	0	0	0	0	0	CRC0n2	CRC0n1	CRC0n0

(n = 0 to 3)

CRC0n2	Selection of operation mode of CR0n1 register
0	Operation as compare register
1	Operation as capture register

CRC0n1	Selection of capture trigger of CR0n0 register
0	Capture at valid edge of TI0n1
1	Capture at inverse phase of valid edge of TI0n0

CRC0n0	Selection of operation mode of CR0n0 register
0	Operation as compare register
1	Operation as capture register

Cautions 1. Before setting the CRC0n register, be sure to stop the timer operation.

2. When the mode in which the timer is cleared and started upon match of the TM0n register and CR0n0 register is selected by the TMC0n register, do not specify the CR0n0 register as the capture register.

★ **3. When both the rising and falling edges are specified for the TI0n0 valid edge and the capture trigger of the CR0n0 register is specified as the inverse phase of the valid edge of TI0n0, a capture operation is not performed.**

4. To ensure a reliable capture operation, a pulse longer than two cycles of the count clock selected by the PRM0n register is required.

(3) 16-bit timer output control register 0n (TOC0n)

TOC0n controls the operation of the 16-bit timer/event counter 0n output controller by setting or resetting the timer output F/F, enabling or disabling inverse output, enabling or disabling the timer of 16-bit timer/event counter 0n, enabling or disabling the one-shot pulse output operation, and selecting the output trigger for the software-based one-shot pulse.

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

This register is cleared to 00H after reset.

After reset: 00H R/W Address: TOC00 FFFFF5C9H, TOC01 FFFFF5D9H,
TOC02 FFFFF5E9H, TOC03 FFFFF5F9H

	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC0n	0	OSPT0n	OSPE0n	TOC0n4	LVS0n	LVR0n	TOC0n1	TOE0n

(n = 0 to 3)

OSPT0n	Output trigger for software-based one-shot pulse
0	–
1	One-shot pulse output

OSPE0n	Control of one-shot pulse output operation
0	Successive pulse output
1	One-shot pulse output[Note]

TOC0n4	Control of timer output F/F upon match of CR0n1 register and TM0n register
0	Inversion operation disabled
1	Inversion operation enabled

LVS0n	LVR0n	Setting of timer output F/F status
0	0	Unchanged
0	1	Reset timer output F/F (0)
1	0	Set timer output F/F (1)
1	1	Setting prohibited

TOC0n1	Control of timer output F/F upon match of CR0n0 register and TM0n register (control of timer output F/F by valid edge of TI0n0)
0	Inversion operation disabled
1	Inversion operation enabled

TOE0n	Control of timer output
0	Output disabled (output is fixed to 0 level)
1	Output enabled

Note The one-shot pulse output operates normally in the free-running mode and the mode in which clear & start occurs on the valid edge of TI0n0. In the mode in which clear & start occurs on match between the TM0n register and the CR0n0 register, one-shot pulse output is not performed because no overflow occurs.

Cautions 1. Be sure to set the bits of this register other than the OSPT0n bit after stopping the timer operation.

2. The LVS0n and LVR0n bits are 0 when read after data has been set to them.

3. The OSPT0n bit is 0 when read because it is automatically cleared after data has been set.

4. Do not set (to 1) the OSPT0n bit other than for one-shot pulse output.

5. When performing successive writes to the OSPT0n bit, place an interval between writes of two or more cycles of the count clock selected by the PRM0n register.

★ **6. Do not the LVS0n bit to 1 before setting the TOE0n bit.**
Do not set the LVS0n and TOE0n to 1 at the same time.

(4) Prescaler mode register 0n (PRM0n)

PRM0n sets the count clock of the TM0n register and the valid edge of the TI0n0 and TI0n1 inputs. This register is set by an 8-bit or 1-bit memory manipulation instruction.
This register is set to 00H after reset.

Cautions 1. When setting the count clock to the TI0n0 valid edge, do not set the mode in which clear & start occurs on the TI0n0 valid edge and do not set the TI0n0 valid edge as the capture trigger.

2. Before setting the PRM0n register, be sure to stop the timer operation.

3. If the TM0n register operation is enabled by specifying the rising edge or both edges for the valid edge of the TI0n0 pin or TI0n1 pin while the TI0n0 pin or TI0n1 pin is high level immediately after system reset, the rising edge is detected immediately after the rising edge or both edges is specified. Be careful when pulling up the TI0n0 pin or TI0n1 pin. However, the rising edge is not detected when operation is enabled after it has been stopped.

4. When the P11 and P12 pins are used as the PWM1 and PWM2 output pins, they cannot be used as timer output pins (TO00, TO01).

(a) Prescaler mode registers 00, 01 (PRM00, PRM01)

After reset: 00H R/W Address: PRM00 FFFFF5C7H, PRM01 FFFFF5D7H

	7	6	5	4	3	2	1	0
PRM0n (n = 0, 1)	ES1n1	ES1n0	ES0n1	ES0n0	0	0	PRM0n1	PRM0n0

ES1n1	ES1n0	Selection of valid edge of TI0n1
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES0n1	ES0n0	Selection of valid edge of TI0n0
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

PRM0n1	PRM0n0	Selection of count clock	
		Count clock	f_{XX}
			20 MHz
0	0	$f_{XX}/4$	200 ns
0	1	$f_{XX}/16$	800 ns
1	0	$f_{XX}/32$	1.6 μs
1	1	Valid edge of TI0n0[Note]	–

Note A pulse longer than two cycles of the internal clock (f_{XX}) is required for the external clock.

Remark f_{XX}: Main clock frequency

(b) Prescaler mode registers 02, 03 (PRM02, PRM03)

After reset: 00H R/W Address: PRM02 FFFFF5E7H, PRM03 FFFFF5F7H

	7	6	5	4	3	2	1	0
PRM0n (n = 2, 3)	ES1n1	ES1n0	ES0n1	ES0n0	0	0	PRM0n1	PRM0n0

ES1n1	ES1n0	Selection of valid edge of TI0n1
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES0n1	ES0n0	Selection of valid edge of TI0n0
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

PRM0n1	PRM0n0	Selection of count clock	
		Count clock	f_{XX}
			20 MHz
0	0	$f_{XX}/4$	200 ns
0	1	$f_{XX}/2^{10}$	51.2 μs
1	0	$f_{XX}/2^{11}$	102.4 μs
1	1	Valid edge of TI0n0[Note]	–

Note A pulse longer than two cycles of the internal clock (f_{XX}) is required for the external clock.

Remark f_{XX}: Main clock frequency

7.4 Operation of 16-Bit Timer/Event Counter 0n

7.4.1 Operation as interval timer (16 bits)

16-bit timer/event counter 0n can be made to operate as an interval timer by setting the TMC0n register and CRC0n register as shown in **Figure 7-2** (n = 0 to 3).

★ Setting method

The basic operation setting method is as follows.

<1> Set the count clock using the PRM0n register.
<2> Set the CRC0n register (see **Figure 7-2** for the setting value).
<3> Set any value to the CR0n0 register.
<4> Set the TMC0n register: Start operation (see **Figure 7-2** for the setting value).

Caution The CR0n0 register cannot be rewritten while TM0n is operating.

Remarks 1. For the alternate-function pin settings, refer to **Table 4-15 Settings When Port Pins Are Used for Alternate Functions**.

2. For INTTM0n0 interrupt enable, refer to **CHAPTER 16 INTERRUPT/EXCEPTION PROCESSING FUNCTION**.

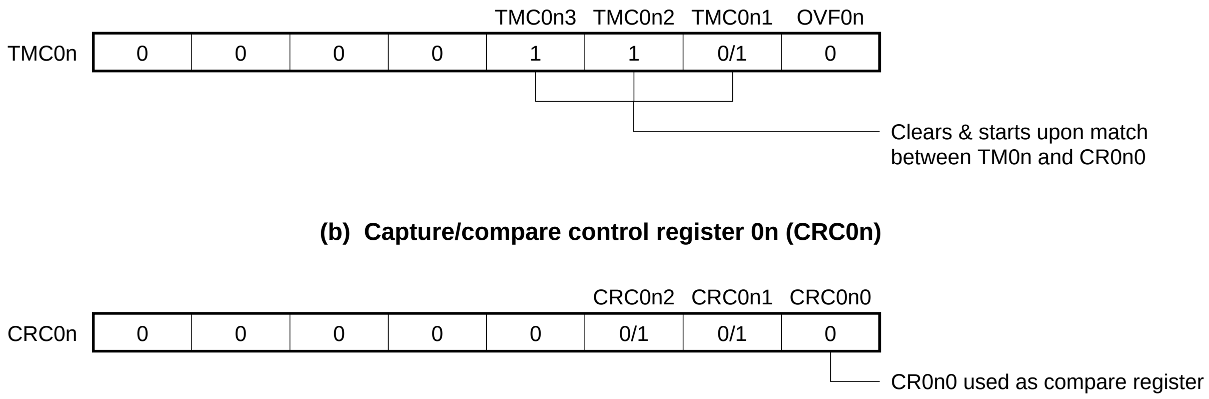
The interval timer repeatedly generates interrupts at the interval of the preset count value in 16-bit timer capture/compare register 0n0 (CR0n0).

If the count value in the TM0n register matches the value set in the CR0n0 register, an interrupt request signal (INTTM0n0) is generated at the same time that the value of the TM0n register is cleared to 0000H and counting is continued.

The count clock of 16-bit timer/event counter 0n can be selected with the PRM0n.PRM0n0 and PRM0n.PRM0n1 bits of prescaler mode register 0n (PRM0n).

Remark n = 0 to 3

Figure 7-2. Control Register Settings for Interval Timer Operation



(a) 16-bit timer mode control register 0n (TMC0n)

					TMC0n3	TMC0n2	TMC0n1	OVF0n
TMC0n	0	0	0	0	1	1	0/1	0

Clears & starts upon match between TM0n and CR0n0

(b) Capture/compare control register 0n (CRC0n)

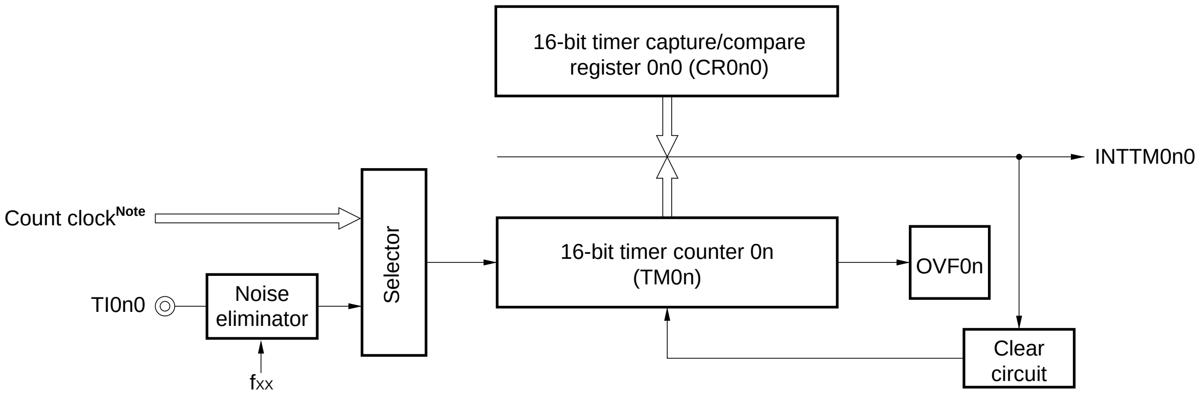
						CRC0n2	CRC0n1	CRC0n0
CRC0n	0	0	0	0	0	0/1	0/1	0

CR0n0 used as compare register

Remarks 1. 0/1: When these bits are reset to 0 or set to 1, other functions can be used together with the interval timer function. For details, refer to **7.3 (1) 16-bit timer mode control register 0n (TMC0n)** and **7.3 (2) Capture/compare control register 0n (CRC0n)**.

2. n = 0 to 3

Figure 7-3. Configuration of Interval Timer

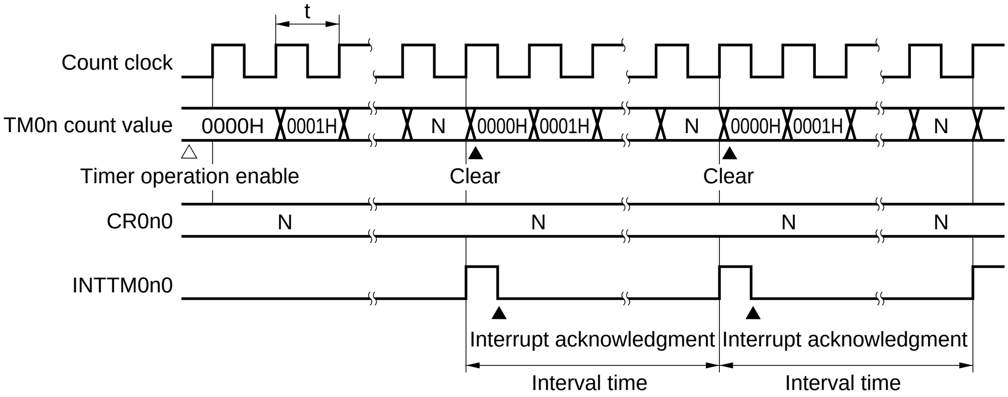

16-bit timer capture/compare register 0n0 (CR0n0)
INTTM0n0
Count clockNote
Selector
16-bit timer counter 0n (TM0n)
OVF0n
TI0n0
Noise eliminator
Clear circuit
fxx

Note Set with PRM0n register.

Remarks 1. fxx: Main clock frequency

2. n = 0 to 3

★ Figure 7-4. Timing of Interval Timer Operation



Remarks 1. Interval time = (N + 1) × t: N = 0001H to FFFFH

2. n = 0 to 3

7.4.2 PPG output operation

16-bit timer/event counter 0n can be used for PPG (Programmable Pulse Generator) output by setting the TMC0n register and CRC0n register as shown in **Figure 7-5**.

★ Setting method

The basic operation setting method is as follows.

<1> Set the CRC0n register (see **Figure 7-5** for the setting value).
<2> Set any value to the CR0n0 register.
<3> Set any value as a duty to the CR0n1 register.
<4> Set the TOC0n register (see **Figure 7-5** for the setting value).
<5> Set the count clock using the PRM0n register.
<6> Set the TMC0n register: Start operation (see **Figure 7-5** for the setting value).

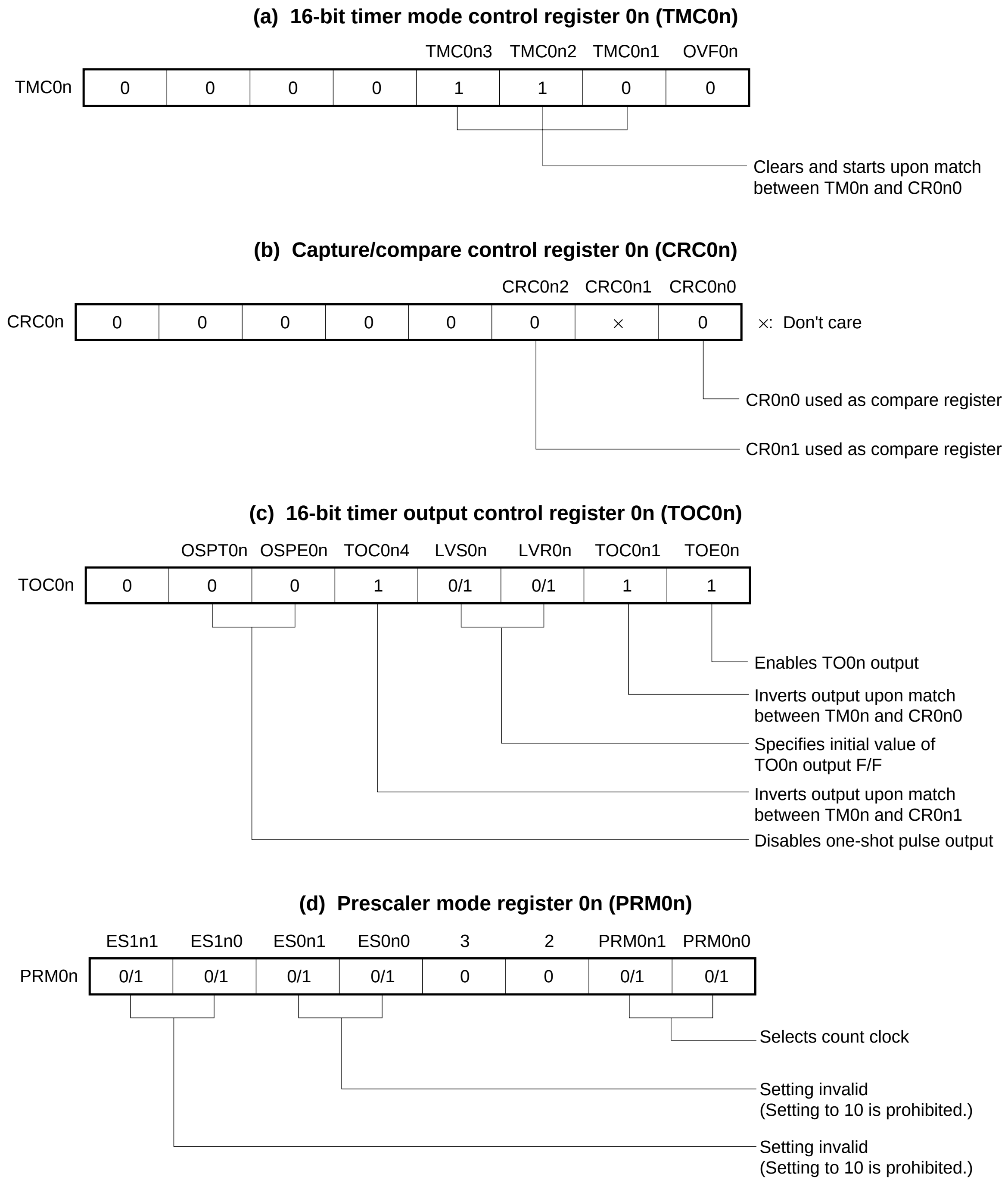
Caution To change the duty value (CR0n1 register) during operation, refer to Remark 2 in Figure 7-7 PPG Output Operation Timing.

Remarks 1. For the alternate-function pin (TO0n) settings, refer to **Table 4-15 Settings When Port Pins Are Used for Alternate Function**.

2. For INTTM0n0 interrupt enable, refer to **CHAPTER 16 INTERRUPT/EXCEPTION PROCESSING FUNCTION**.

The PPG output function outputs a rectangular wave from the TO0n pin with the cycle specified by the count value set in advance to the CR0n0 register and the pulse width specified by the count value set in advance to the CR0n1 register.

Figure 7-5. Control Register Settings in PPG Output Operation



(a) 16-bit timer mode control register 0n (TMC0n)

					TMC0n3	TMC0n2	TMC0n1	OVF0n
TMC0n	0	0	0	0	1	1	0	0

Clears and starts upon match between TM0n and CR0n0

(b) Capture/compare control register 0n (CRC0n)

						CRC0n2	CRC0n1	CRC0n0
CRC0n	0	0	0	0	0	0	×	0

×: Don't care
CR0n0 used as compare register
CR0n1 used as compare register

(c) 16-bit timer output control register 0n (TOC0n)

		OSPT0n	OSPE0n	TOC0n4	LVS0n	LVR0n	TOC0n1	TOE0n
TOC0n	0	0	0	1	0/1	0/1	1	1

Enables TO0n output
Inverts output upon match between TM0n and CR0n0
Specifies initial value of TO0n output F/F
Inverts output upon match between TM0n and CR0n1
Disables one-shot pulse output

(d) Prescaler mode register 0n (PRM0n)

	ES1n1	ES1n0	ES0n1	ES0n0	3	2	PRM0n1	PRM0n0
PRM0n	0/1	0/1	0/1	0/1	0	0	0/1	0/1

Selects count clock
Setting invalid (Setting to 10 is prohibited.)
Setting invalid (Setting to 10 is prohibited.)

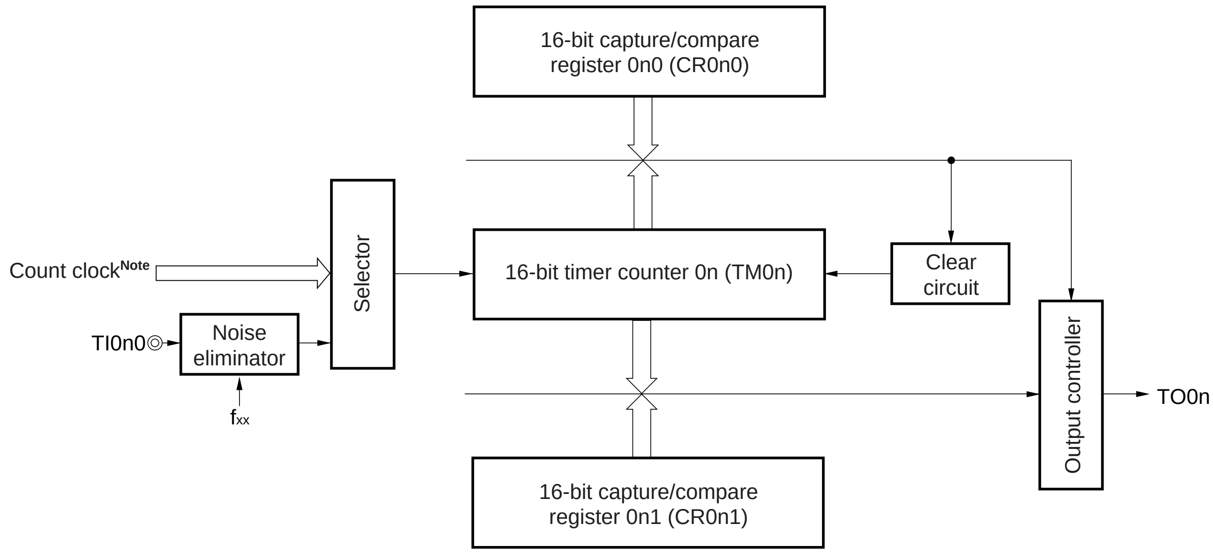
Cautions 1. Make sure that 0000H < CR0n1 < CR0n0 ≤ FFFFH is set to the CR0n0 register and CR0n1 register.

2. The cycle of the pulse generated by PPG output is (CR0n0 setting value + 1).
The duty factor is (CR0n1 setting value + 1) / (CR0n0 setting value + 1).

Remark n = 0 to 3

★

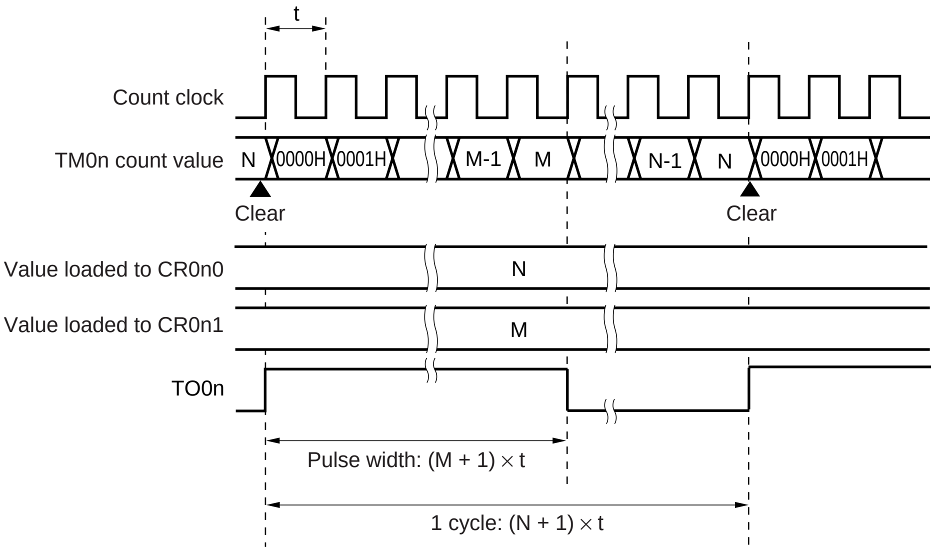
Figure 7-6. Configuration of PPG Output


16-bit capture/compare register 0n0 (CR0n0)
Count clockNote
TI0n0
Noise eliminator
fxx
Selector
16-bit timer counter 0n (TM0n)
Clear circuit
Output controller
TO0n
16-bit capture/compare register 0n1 (CR0n1)

Note The count clock is set by the PRM0n register.

Remark n = 0 to 3

★ **Figure 7-7. PPG Output Operation Timing**



Caution CR0n0 cannot be rewritten during TM0n operation.

Remarks 1. 0000H < M < N ≤ FFFFH

2. Change the pulse width during TM0n operation (rewrite CR0n1) as follows in a PPG output operation.

<1> Disable the timer output inversion operation based on a match of TM0n and CR0n1 (TOC0n4 = 0).
<2> Disable the INTTM0n1 interrupt (TMMK0n1 = 1).
<3> Rewrite CR0n1.
<4> Wait for a cycle of the TM0n count clock.
<5> Enable the timer output inversion operation based on a match of TM0n and CR0n1 (TOC0n4 = 1).
<6> Clear the interrupt request flag of INTTM0n1 (TMIF0n1 = 0).
<7> Enable the INTTM0n1 interrupt (TMMK0n1 = 0).

3. n = 0 to 3

7.4.3 Pulse width measurement

The TM0n register can be used to measure the pulse widths of the signals input to the TI0n0 and TI0n1 pins.

Measurement can be carried out with the TM0n register used as a free-running counter or by restarting the timer in synchronization with the edge of the signal input to the TI0n0 pin.

When an interrupt is generated, read the valid capture register value. After confirming the OVF flag, clear it by software and measure the pulse width.

★ Setting method

The basic operation setting method is as follows.

<1> Set the CRC0n register (see **Figures 7-9**, **7-12**, **7-14**, and **7-16** for the setting value).
<2> Set the count clock using the PRM0n register.
<3> Set the TMC0n register: Start operation (see **Figures 7-9**, **7-12**, **7-14**, and **7-16** for the setting value).

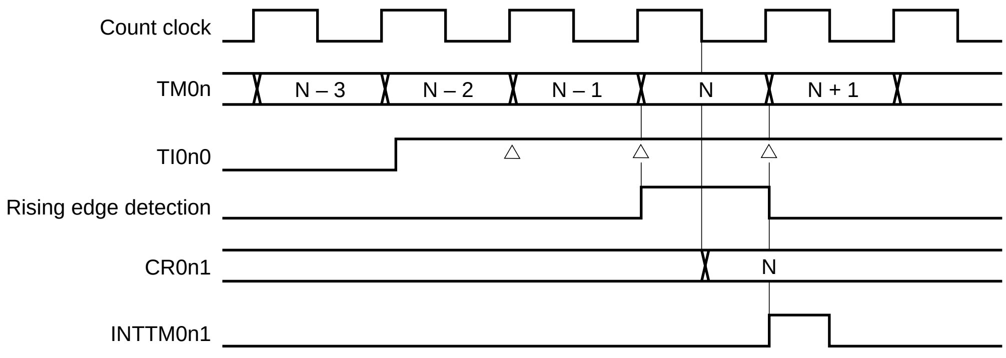
Note When using two capture registers, set the TI0n0 and TI0n1 pins.

Remarks 1. For the alternate-function pin (TI0n0, TI0n1) settings, refer **Table 4-15 Settings When Port Pins Are Used for Alternate Functions**.

2. For INTTM0n0 interrupt enable, refer to **CHAPTER 16 INTERRUPT/EXCEPTION PROCESSING FUNCTION**.

The valid edge is detected through sampling at a count clock cycle selected with the PRM0n register, and the capture operation is not performed until the valid edge is detected twice. As a result, noise with a short pulse width can be eliminated.

Figure 7-8. CR0n1 Capture Operation with Rising Edge Specified



Remark n = 0 to 3

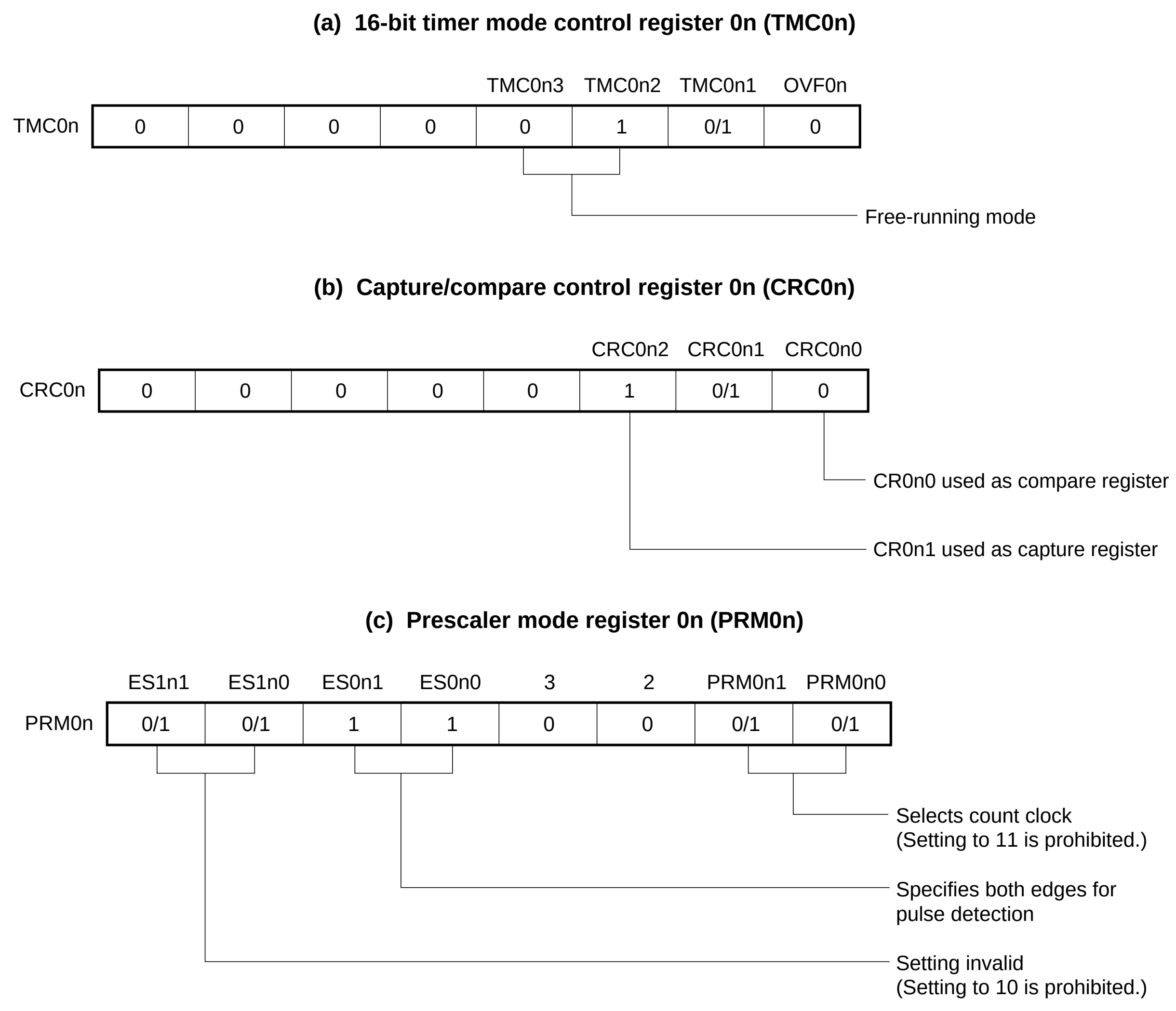
(1) Pulse width measurement with free-running counter and one capture register

If the edge specified by the PRM0n register is input to the TI0n0 pin when the TM0n register is operated as a free-running counter (refer to **Figure 7-9**), the value of the TM0n register is loaded to the CR0n1 register and an external interrupt request signal (INTTM0n1) is set.

The edge is specified by using the PRM0n.ES0n0 and PRM0n.ES0n1 bits. The rising edge, falling edge, or both the rising and falling edges can be selected.

Remark n = 0 to 3

Figure 7-9. Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register (When TI0n0 Pin and CR0n1 Register Are Used)



(a) 16-bit timer mode control register 0n (TMC0n)

					TMC0n3	TMC0n2	TMC0n1	OVF0n
TMC0n	0	0	0	0	0	1	0/1	0

(b) Capture/compare control register 0n (CRC0n)

						CRC0n2	CRC0n1	CRC0n0
CRC0n	0	0	0	0	0	1	0/1	0

CR0n0 used as compare register
CR0n1 used as capture register

(c) Prescaler mode register 0n (PRM0n)

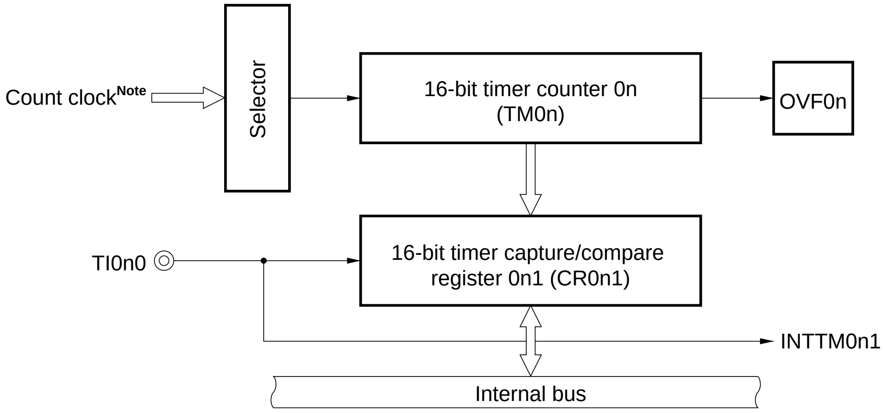
	ES1n1	ES1n0	ES0n1	ES0n0	3	2	PRM0n1	PRM0n0
PRM0n	0/1	0/1	1	1	0	0	0/1	0/1

Selects count clock
(Setting to 11 is prohibited.)
Specifies both edges for
pulse detection
Setting invalid
(Setting to 10 is prohibited.)

Remarks 1. 0/1: When these bits are reset to 0 or set to 1, other functions can be used together with the pulse width measurement function. For details, refer to **7.3 (1) 16-bit timer mode control register 0n (TMC0n)** and **7.3 (2) Capture/compare control register 0n (CRC0n)**.

2. n = 0 to 3

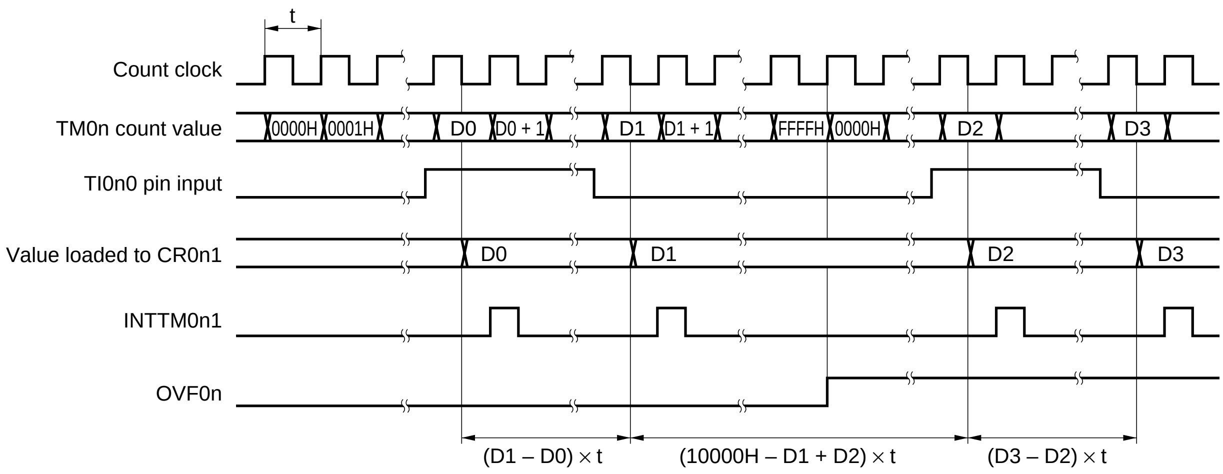
Figure 7-10. Configuration for Pulse Width Measurement with Free-Running Counter



Note The count clock is set with the PRM0n register.

Remark n = 0 to 3

Figure 7-11. Timing of Pulse Width Measurement with Free-Running Counter and One Capture Register (with Both Edges Specified)



Remark n = 0 to 3

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

The pulse widths of two signals respectively input to the TI0n0 pin and the TI0n1 pin can be simultaneously measured when the TM0n register is used as a free-running counter (refer to **Figure 7-12**).

When the edge specified by the PRM0n.ES0n0 and PRM0n.ES0n1 bits is input to the TI0n0 pin, the value of the TM0n register is loaded to 16-bit timer capture/compare register 0n1 (CR0n1) and an external interrupt request signal (INTTM0n1) is set.

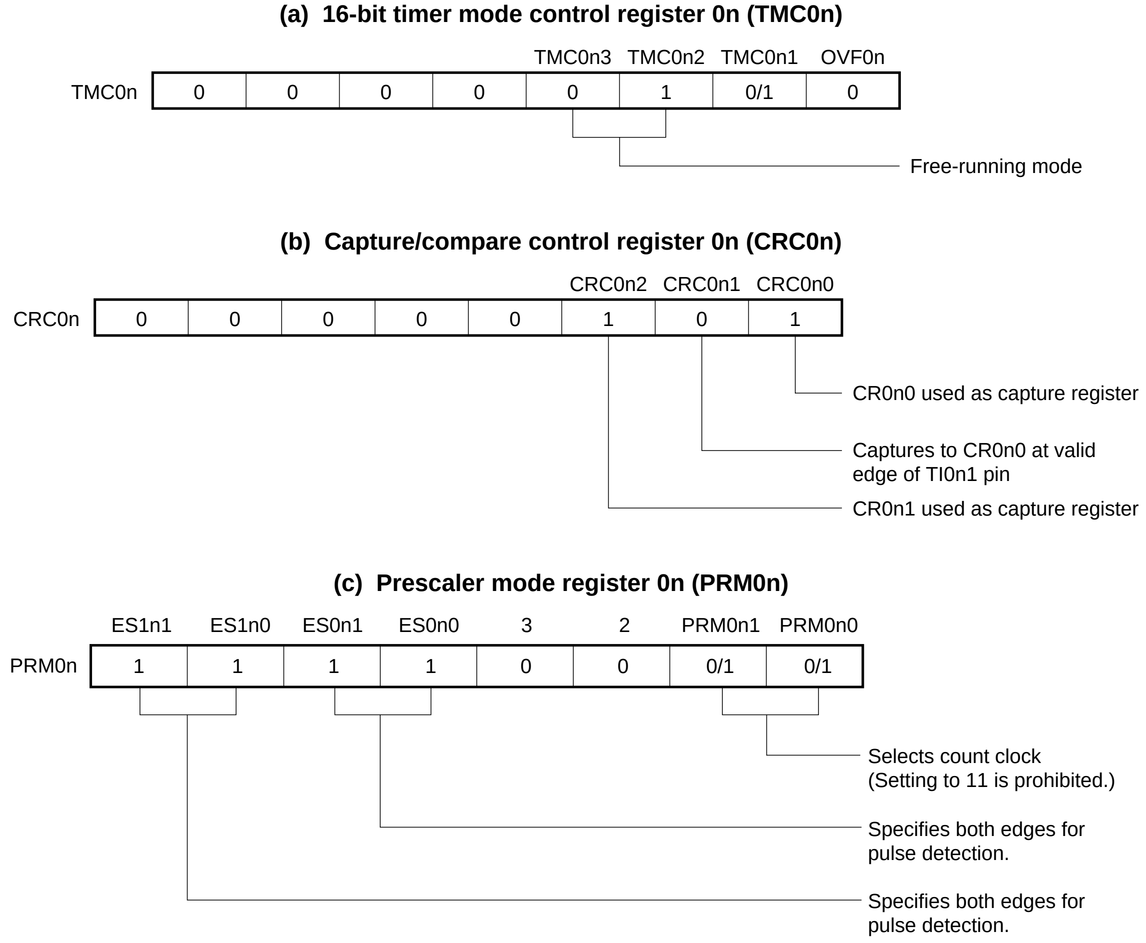
When the edge specified by the PRM0n.ES1n0 and PRM0n.ES1n1 bits is input to the TI0n1 pin, the value of the TM0n register is loaded to the CR0n0 register and an external interrupt request signal (INTTM0n0) is set.

The edges of the TI0n0 and TI0n1 pins are specified by the PRM0n.ES0n0 and PRM0n.ES0n1 bits and the PRM0n.ES1n0 and PRM0n.ES1n1 bits, respectively. The rising, falling, or both rising and falling edges can be specified.

The valid edge of the TI0n0 pin is detected through sampling at the count clock cycle selected with the PRM0n register, and the capture operation is not performed until the valid level is detected twice. As a result, noise with a short pulse width can be eliminated.

Remark n = 0 to 3

Figure 7-12. Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter



(a) 16-bit timer mode control register 0n (TMC0n)

					TMC0n3	TMC0n2	TMC0n1	OVF0n
TMC0n	0	0	0	0	0	1	0/1	0

(b) Capture/compare control register 0n (CRC0n)

						CRC0n2	CRC0n1	CRC0n0
CRC0n	0	0	0	0	0	1	0	1

CR0n0 used as capture register
Captures to CR0n0 at valid edge of TI0n1 pin
CR0n1 used as capture register

(c) Prescaler mode register 0n (PRM0n)

	ES1n1	ES1n0	ES0n1	ES0n0	3	2	PRM0n1	PRM0n0
PRM0n	1	1	1	1	0	0	0/1	0/1

Selects count clock (Setting to 11 is prohibited.)
Specifies both edges for pulse detection.
Specifies both edges for pulse detection.

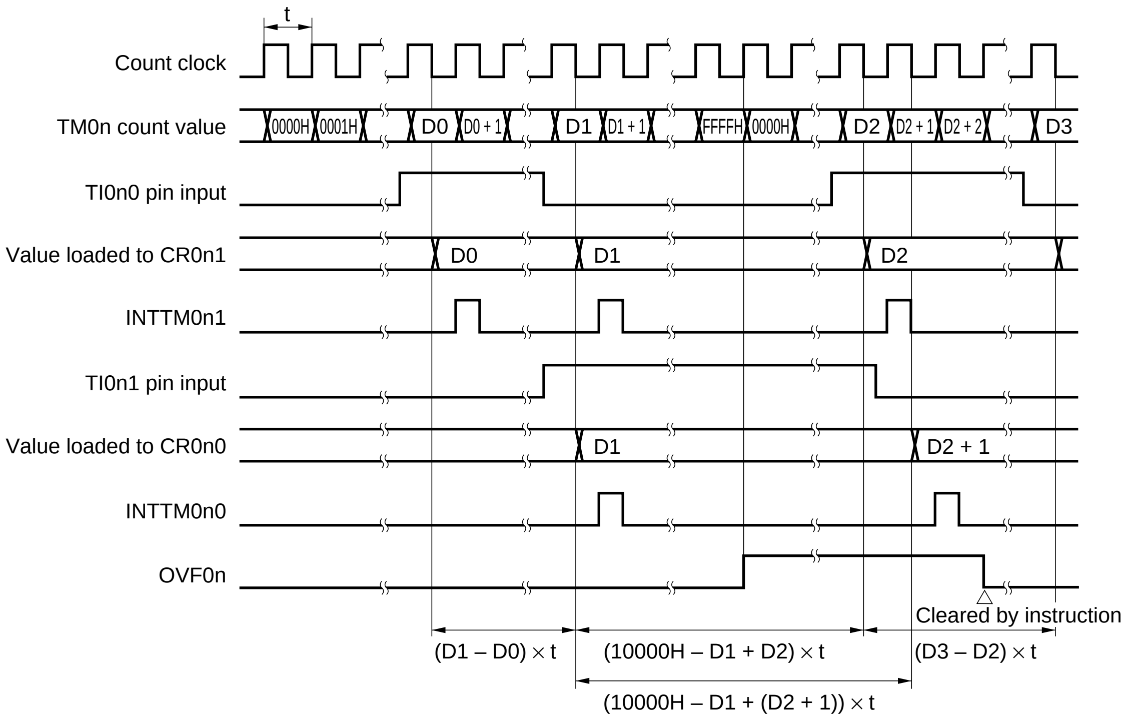
Remarks 1. 0/1: When these bits are reset to 0 or set to 1, other functions can be used together with the pulse width measurement function. For details, refer to **7.3 (1) 16-bit timer mode control register 0n (TMC0n)**.

2. n = 0 to 3

- Capture operation (free-running mode)

  The following figure illustrates the operation of the capture register when the capture trigger is input.

Figure 7-13. Timing of Pulse Width Measurement with Free-Running Counter (with Both Edges Specified)



Remark n = 0 to 3

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

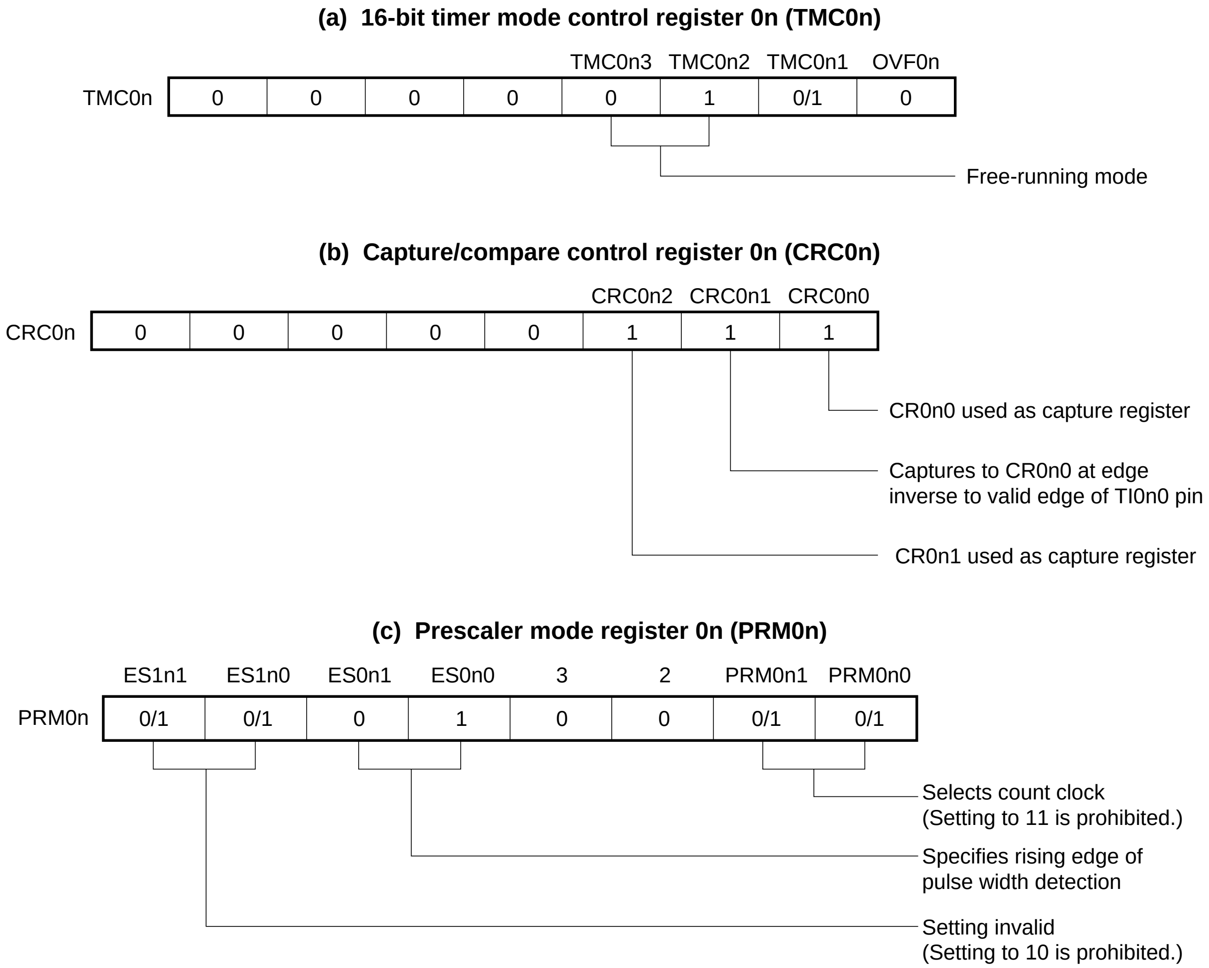
When the TM0n register is used as a free-running counter (refer to **Figure 7-14**), the pulse width of the signal input to the TI0n0 pin can be measured.

When the edge specified by the PRM0n.ES0n0 and PRM0n.ES0n1 bits is input to the TI0n0 pin, the value of the TM0n register is loaded to the CR0n1 register and an external interrupt request signal (INTTM0n1) is set.

The value of the TM0n register is also loaded to the CR0n0 register when an edge inverse to the one that triggers capturing to the CR0n1 register is input.

The valid edge of the TI0n0 pin is detected through sampling at a count clock cycle selected with the PRM0n register, and the capture operation is not performed until the valid edge is detected twice. As a result, noise with a short pulse width can be eliminated.

Figure 7-14. 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 0n (TMC0n)

					TMC0n3	TMC0n2	TMC0n1	OVF0n
TMC0n	0	0	0	0	0	1	0/1	0

(b) Capture/compare control register 0n (CRC0n)

						CRC0n2	CRC0n1	CRC0n0
CRC0n	0	0	0	0	0	1	1	1

CR0n0 used as capture register
Captures to CR0n0 at edge inverse to valid edge of TI0n0 pin
CR0n1 used as capture register

(c) Prescaler mode register 0n (PRM0n)

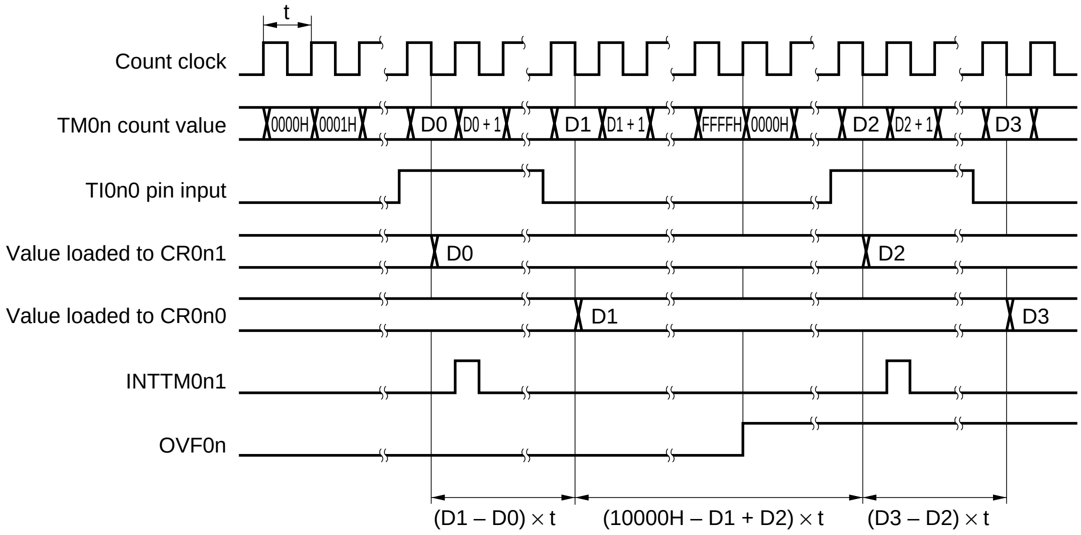
	ES1n1	ES1n0	ES0n1	ES0n0	3	2	PRM0n1	PRM0n0
PRM0n	0/1	0/1	0	1	0	0	0/1	0/1

Selects count clock (Setting to 11 is prohibited.)
Specifies rising edge of pulse width detection
Setting invalid (Setting to 10 is prohibited.)

Remarks 1. 0/1: When these bits are reset to 0 or set to 1, other functions can be used together with the pulse width measurement function. For details, refer to **7.3 (1) 16-bit timer mode control register 0n (TMC0n)**.

2. n = 0 to 3

Figure 7-15. Timing of Pulse Width Measurement with Free-Running Counter and Two Capture Registers (with Rising Edge Specified)



Remark n = 0 to 3

(4) Pulse width measurement by restarting

When the valid edge of the TI0n0 pin is detected, the pulse width of the signal input to the TI0n0 pin can be measured by clearing the TM0n register and then resuming counting after loading the count value of the TM0n register to the CR0n1 register (refer to **Figure 7-17**).

The edge is specified by the PRM0n.ES0n0 and PRM0n.ES0n1 bits. The rising or falling edge can be specified.

The valid edge is detected through sampling at a count clock cycle selected with the PRM0n register and the capture operation is not performed until the valid level is detected twice. As a result, noise with a short pulse width can be eliminated.

Figure 7-16. Control Register Settings for Pulse Width Measurement by Restarting (1/2)

(a) 16-bit timer mode control register 0n (TMC0n)

					TMC0n3	TMC0n2	TMC0n1	OVF0n
TMC0n	0	0	0	0	1	0	0/1	0

TMC0n3, TMC0n2 = 1, 0: Clears and starts at valid edge of TI0n0 pin

Figure 7-16. Control Register Settings for Pulse Width Measurement by Restarting (2/2)

(b) Capture/compare control register 0n (CRC0n)

						CRC0n2	CRC0n1	CRC0n0
CRC0n	0	0	0	0	0	1	1	1

CRC0n0: CR0n0 used as capture register

CRC0n1: Captures to CR0n0 at edge inverse to valid edge of TI0n0 pin

CRC0n2: CR0n1 used as capture register

(c) Prescaler mode register 0n (PRM0n)

	ES1n1	ES1n0	ES0n1	ES0n0	3	2	PRM0n1	PRM0n0
PRM0n	0/1	0/1	0	1	0	0	0/1	0/1

PRM0n1, PRM0n0: Selects count clock (Setting to 11 is prohibited.)

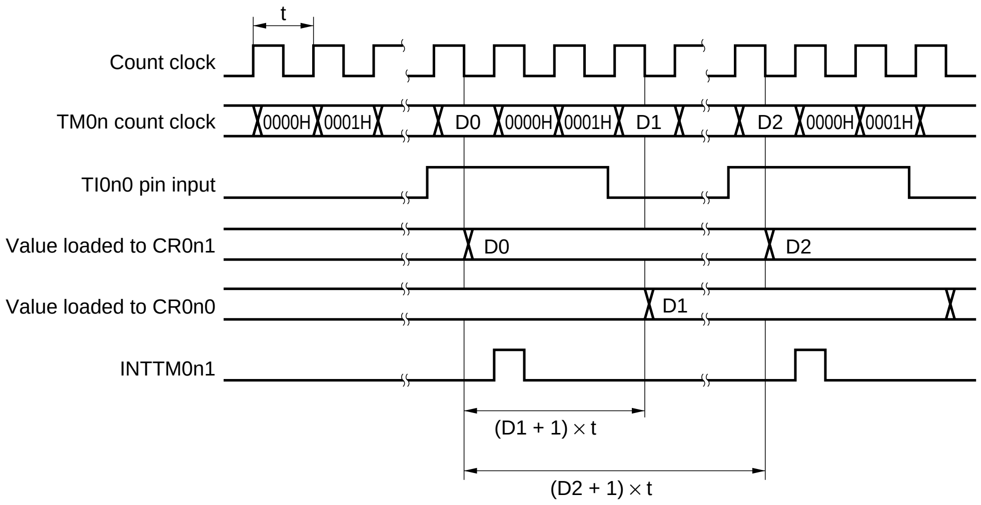
ES0n1, ES0n0: Specifies rising edge of pulse width detection

ES1n1, ES1n0: Setting invalid (Setting to 10 is prohibited.)

Remarks 1. 0/1: When these bits are reset to 0 or set to 1, other functions can be used together with the pulse width measurement function. For details, refer to **7.3 (1) 16-bit timer mode control register 0n (TMC0n)**.

2. n = 0 to 3

Figure 7-17. Timing of Pulse Width Measurement by Restarting (with Rising Edge Specified)



Remark n = 0 to 3

7.4.4 Operation as external event counter

★ Setting method

The basic operation setting method is as follows.

<1> Set the CRC0n register (see **Figure 7-18** for the setting value).
<2> Set the count clock using the PRM0n register.
<3> Set any value (except for 0000H) to the CRC0n0 register.
<4> Set the TMC0n register: Start operation (see **Figure 7-18** for the setting value).

Remarks 1. For the alternate-function pin (TI0n0) settings, refer to **Table 4-15 Settings When Port Pins Are Used for Alternate Functions**.

2. For INTTM0n0 interrupt enable, refer to **CHAPTER 16 INTERRUPT/EXCEPTION PROCESSING FUNCTION**.

The external event counter counts the number of clock pulses input to the TI0n0 pin from an external source by using the TM0n register.

Each time the valid edge specified by prescaler mode register 0n (PRM0n) has been input, the TM0n register is incremented.

When the count value of the TM0n register matches the value of the CR0n0 register, the TM0n register is cleared to 0000H and an interrupt request signal (INTTM0n0) is generated.

Set the CR0n0 register to a value other than 0000H (one-pulse count operation is not possible).

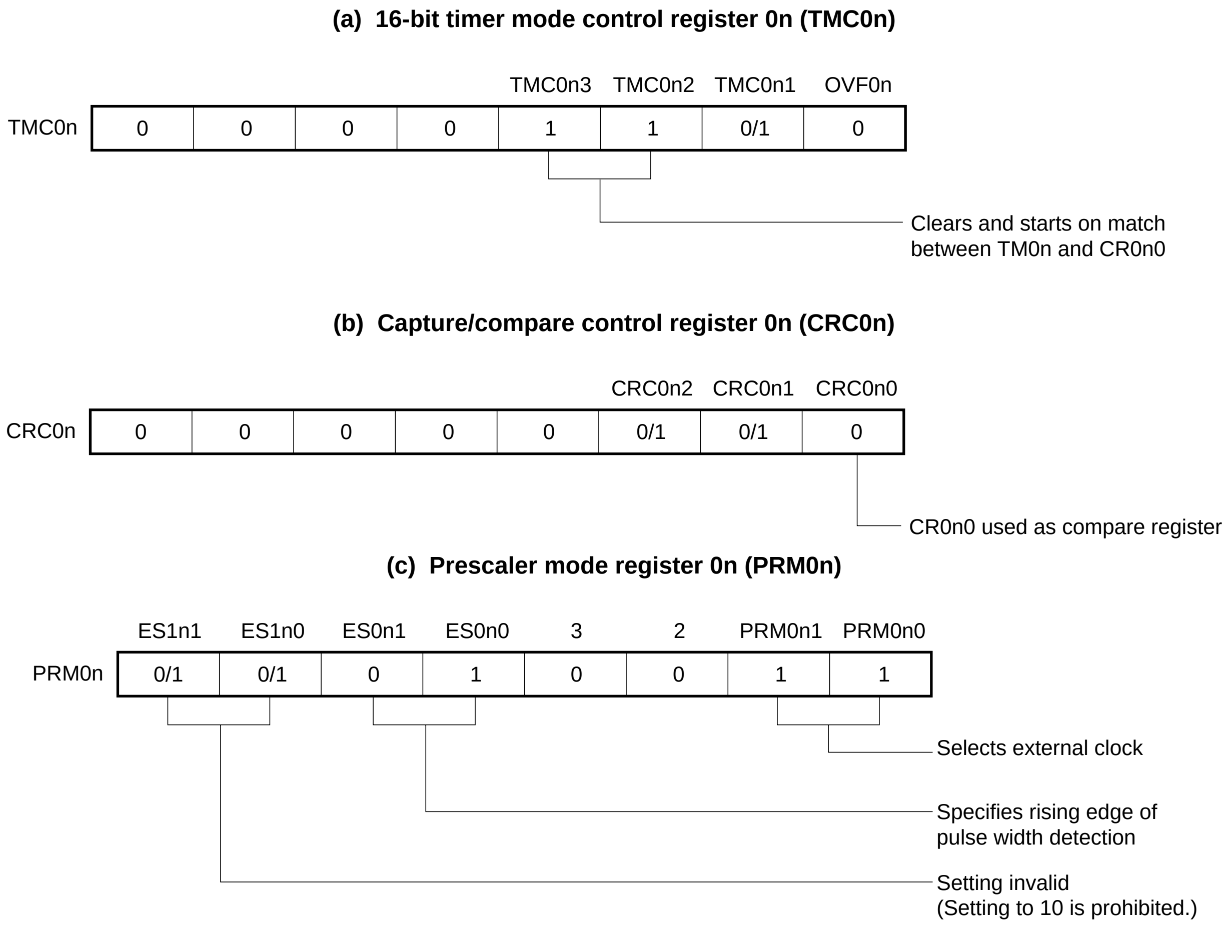
The edge is specified by the PRM0.ES0n0 and PRM0n.ES0n1 bits. The rising, falling, or both the rising and falling edges can be specified.

The valid edge is detected through sampling at a count clock cycle of f_{XX}, and the capture operation is not performed until the valid level is detected twice. As a result, noise with a short pulse width can be eliminated.

Caution The values of the CR0n0 and CR0n1 registers cannot be changed during timer count operation.

Remark n = 0 to 3

Figure 7-18. Control Register Settings in External Event Counter Mode



(a) 16-bit timer mode control register 0n (TMC0n)

					TMC0n3	TMC0n2	TMC0n1	OVF0n
TMC0n	0	0	0	0	1	1	0/1	0

Clears and starts on match between TM0n and CR0n0

(b) Capture/compare control register 0n (CRC0n)

						CRC0n2	CRC0n1	CRC0n0
CRC0n	0	0	0	0	0	0/1	0/1	0

CR0n0 used as compare register

(c) Prescaler mode register 0n (PRM0n)

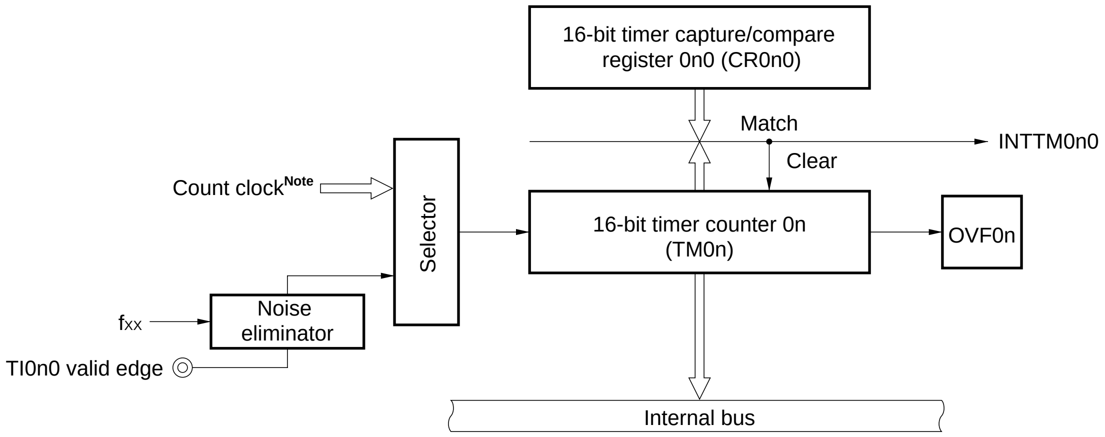
	ES1n1	ES1n0	ES0n1	ES0n0	3	2	PRM0n1	PRM0n0
PRM0n	0/1	0/1	0	1	0	0	1	1

Selects external clock
Specifies rising edge of pulse width detection
Setting invalid (Setting to 10 is prohibited.)

Remarks 1. 0/1: When these bits are reset to 0 or set to 1, other functions can be used together with the external event counter function. For details, refer to **7.3 (1) 16-bit timer mode control register 0n (TMC0n)** and **7.3 (2) Capture/compare control register 0n (CRC0n)**.

2. n = 0 to 3

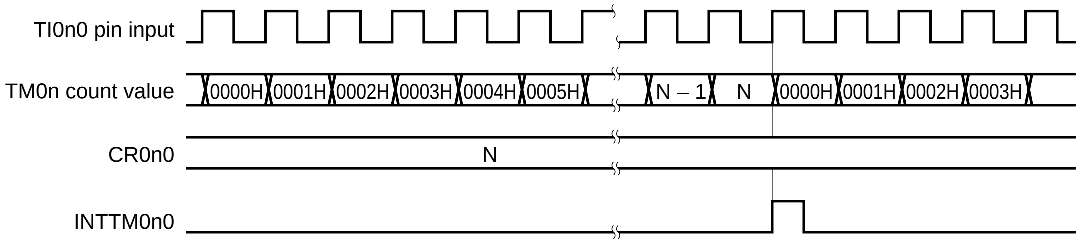
Figure 7-19. Configuration of External Event Counter



Note Set with the PRM0n register.

Remark n = 0 to 3

Figure 7-20. Timing of External Event Counter Operation (with Rising Edge Specified)



Caution Read the TM0n register when reading the count value of the external event counter.

Remark n = 0 to 3

7.4.5 Square-wave output operation

★ Setting method

The basic operation setting method is as follows.

<1> Set the count clock using the PRM0n register.
<2> Set the CRC0n register (see **Figure 7-21** for the setting value).
<3> Set the TOC0n register (see **Figure 7-21** for the setting value).
<4> Set any value (except for 0000H) to the CRC0n0 register.
<5> Set the TMC0n register: Start operation (see **Figure 7-21** for the setting value).

Remarks 1. For the alternate-function pin (TO0n) settings, refer to **Table 4-15 Settings When Port Pins Are Used for Alternate Function**.

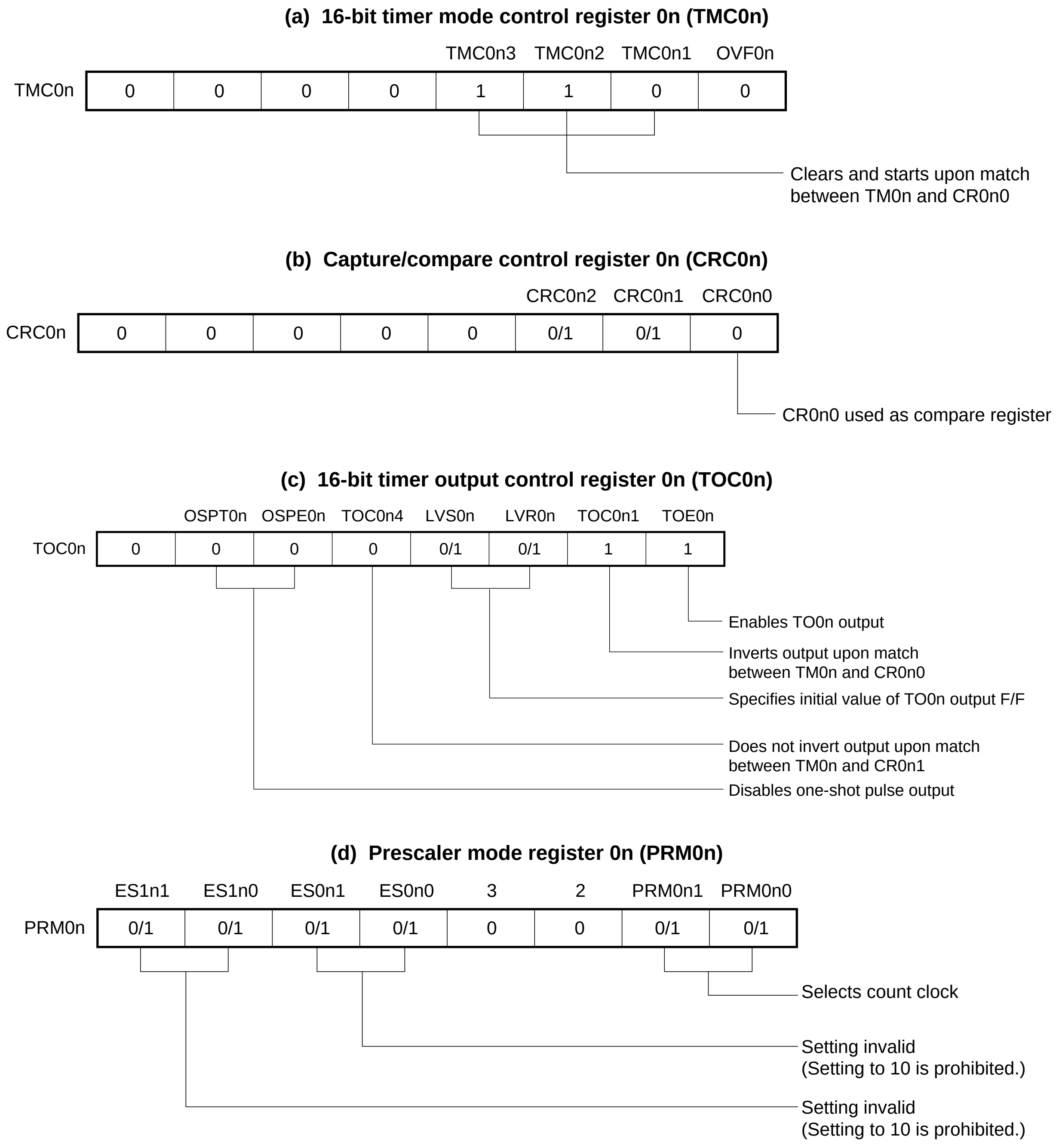
2. For INTTM0n0 interrupt enable, refer to **CHAPTER 16 INTERRUPT/EXCEPTION PROCESSING FUNCTION**.

16-bit timer/event counter 0n can be used to output a square wave with any frequency at an interval specified by the count value set in advance to the CR0n0 register.

By setting the TOC0n.TOE0n and TOC0n.TOC0n1 bits to 11, the output status of the TO0n pin is inverted at an interval set in advance to the CR0n0 register. In this way, a square wave of any frequency can be output.

Caution The values of the CR0n0 and CR0n1 registers cannot be changed during timer operation.

Figure 7-21. Control Register Settings in Square-Wave Output Mode



(a) 16-bit timer mode control register 0n (TMC0n)

					TMC0n3	TMC0n2	TMC0n1	OVF0n
TMC0n	0	0	0	0	1	1	0	0

TMC0n3, TMC0n2, TMC0n1: Clears and starts upon match between TM0n and CR0n0

(b) Capture/compare control register 0n (CRC0n)

						CRC0n2	CRC0n1	CRC0n0
CRC0n	0	0	0	0	0	0/1	0/1	0

CRC0n0: CR0n0 used as compare register

(c) 16-bit timer output control register 0n (TOC0n)

		OSPT0n	OSPE0n	TOC0n4	LVS0n	LVR0n	TOC0n1	TOE0n
TOC0n	0	0	0	0	0/1	0/1	1	1

TOE0n: Enables TO0n output

TOC0n1: Inverts output upon match between TM0n and CR0n0

LVS0n, LVR0n: Specifies initial value of TO0n output F/F

TOC0n4: Does not invert output upon match between TM0n and CR0n1

OSPT0n, OSPE0n: Disables one-shot pulse output

(d) Prescaler mode register 0n (PRM0n)

	ES1n1	ES1n0	ES0n1	ES0n0	3	2	PRM0n1	PRM0n0
PRM0n	0/1	0/1	0/1	0/1	0	0	0/1	0/1

PRM0n1, PRM0n0: Selects count clock

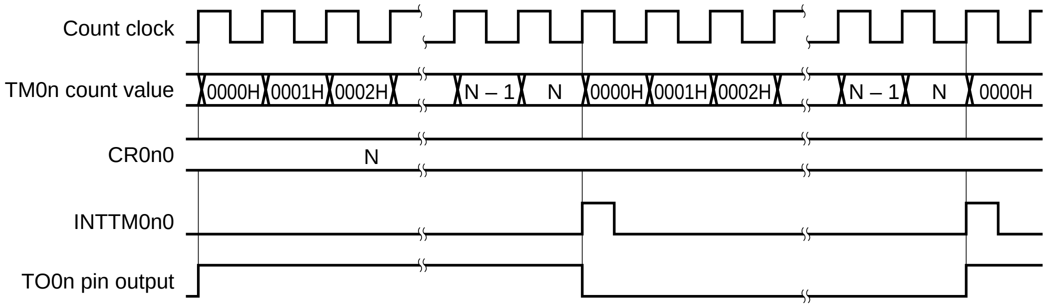
ES0n1, ES0n0: Setting invalid (Setting to 10 is prohibited.)

ES1n1, ES1n0: Setting invalid (Setting to 10 is prohibited.)

Remarks 1. For details, refer to **7.3 (2) Capture/compare control register 0n (CRC0n)** and **7.3 (3) 16-bit timer output control register 0n (TOC0n)**.

2. n = 0 to 3

Figure 7-22. Timing of Square-Wave Output Operation



Remark n = 0 to 3

7.4.6 One-shot pulse output operation

16-bit timer/event counter 0n can output a one-shot pulse in synchronization with a software trigger or an external trigger (TI0n0 pin input).

★ Setting method

The basic operation setting method is as follows.

<1> Set the count clock using the PRM0m register.
<2> Set the CRC0m register (see **Figures 7-23** and **7-24** for the setting value).
<3> Set the TOC0m register (see **Figures 7-23** and **7-24** for the setting value).
<4> Set any value to the CR0m0 and CR0m1 registers.
<5> Set the TMC0m register: Start operation (see **Figures 7-23** and **7-25** for the setting value).

Remarks 1. For the alternate-function pin (TO0m) settings, refer to **Table 4-15 Settings When Port Pins Are Used for Alternate Functions**.

2. For INTTM0m0 interrupt enable, refer to **CHAPTER 16 INTERRUPT/EXCEPTION PROCESSING FUNCTION**.

(1) One-shot pulse output with software trigger

A one-shot pulse can be output from the TO0n pin by setting the TMC0n, CRC0n, and TOC0n registers as shown in Figure 7-23, and by setting the TOC0n.OSPT0n bit (1) by software.

By setting the OSPT0n bit (1), 16-bit timer/event counter 0n is cleared and started, and its output becomes active at the count value (N) set in advance to the CR0n1 register. After that, the output becomes inactive at the count value (M) set in advance to the CR0n0 register[Note].

Even after the one-shot pulse has been output, the TM0n register continues its operation. To stop the TM0n register, the TMC0n.TMC0n3 and TMC0n.TMC0n2 bits must be set to 00.

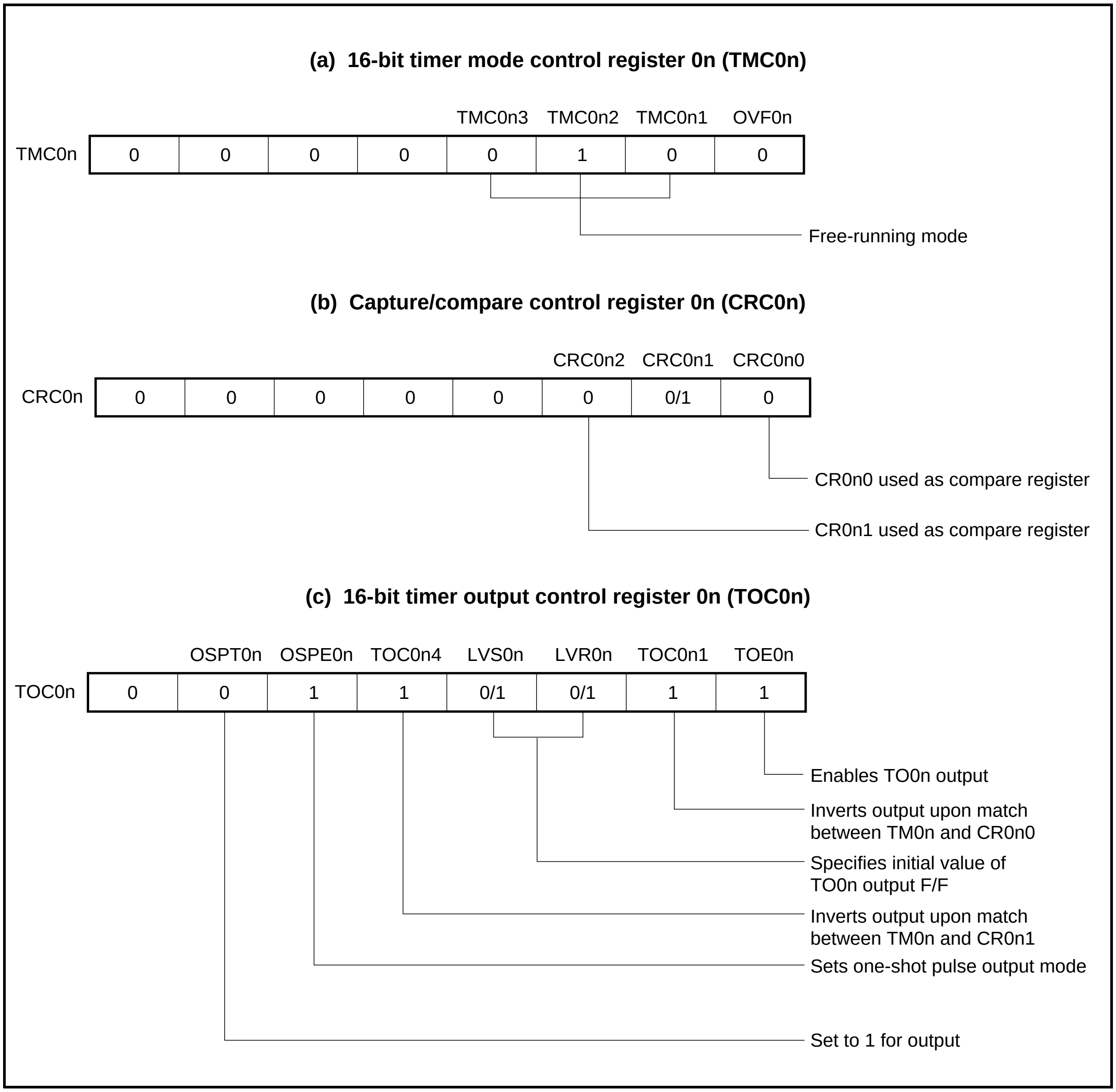
Note The case where N < M is described here. When N > M, the output becomes active with the CR0n0 register and inactive with the CR0n1 register.

Cautions 1. Do not set the OSPT0n 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. The values of the CR0n0 and CR0n1 registers cannot be changed during timer count operation.

Remark n = 0 to 3

Figure 7-23. Control Register Settings for One-Shot Pulse Output with Software Trigger (1/2)



(a) 16-bit timer mode control register 0n (TMC0n)

					TMC0n3	TMC0n2	TMC0n1	OVF0n
TMC0n	0	0	0	0	0	1	0	0

(b) Capture/compare control register 0n (CRC0n)

						CRC0n2	CRC0n1	CRC0n0
CRC0n	0	0	0	0	0	0	0/1	0

CR0n0 used as compare register
CR0n1 used as compare register

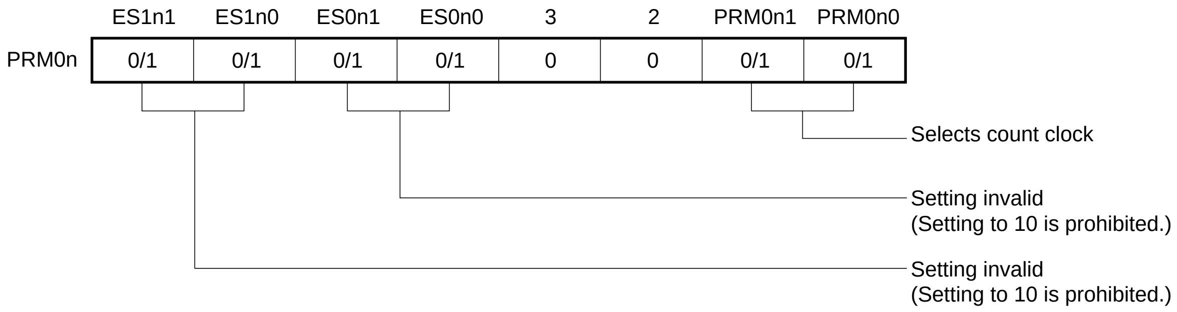
(c) 16-bit timer output control register 0n (TOC0n)

		OSPT0n	OSPE0n	TOC0n4	LVS0n	LVR0n	TOC0n1	TOE0n
TOC0n	0	0	1	1	0/1	0/1	1	1

Enables TO0n output
Inverts output upon match between TM0n and CR0n0
Specifies initial value of TO0n output F/F
Inverts output upon match between TM0n and CR0n1
Sets one-shot pulse output mode
Set to 1 for output

Figure 7-23. Control Register Settings for One-Shot Pulse Output with Software Trigger (2/2)

(d) Prescaler mode register 0n (PRM0n)



Caution Do not set the CR0n0 register to 0000H.

Remarks 1. For details, refer to **7.3 (2) Capture/compare control register 0n (CRC0n)** and **7.3 (3) 16-bit timer output control register 0n (TOC0n)**.

2. n = 0 to 3

(2) One-shot pulse output with external trigger

A one-shot pulse can be output from the TO0n pin by setting the TMC0n, CRC0n, and TOC0n registers as shown in Figure 7-24, and by using the valid edge of the TI0n0 pin as an external trigger.

The valid edge of the TI0n0 pin is specified by the PRM0n.ES0n0 and PRM0n.ES0n1 bits. The rising, falling, or both the rising and falling edges can be specified.

When the valid edge of the TI0n0 pin is detected, 16-bit timer/event counter 0n is cleared and started, and the output becomes active at the count value set in advance to the CR0n1 register. After that, the output becomes inactive at the count value set in advance to the CR0n0 register[Note].

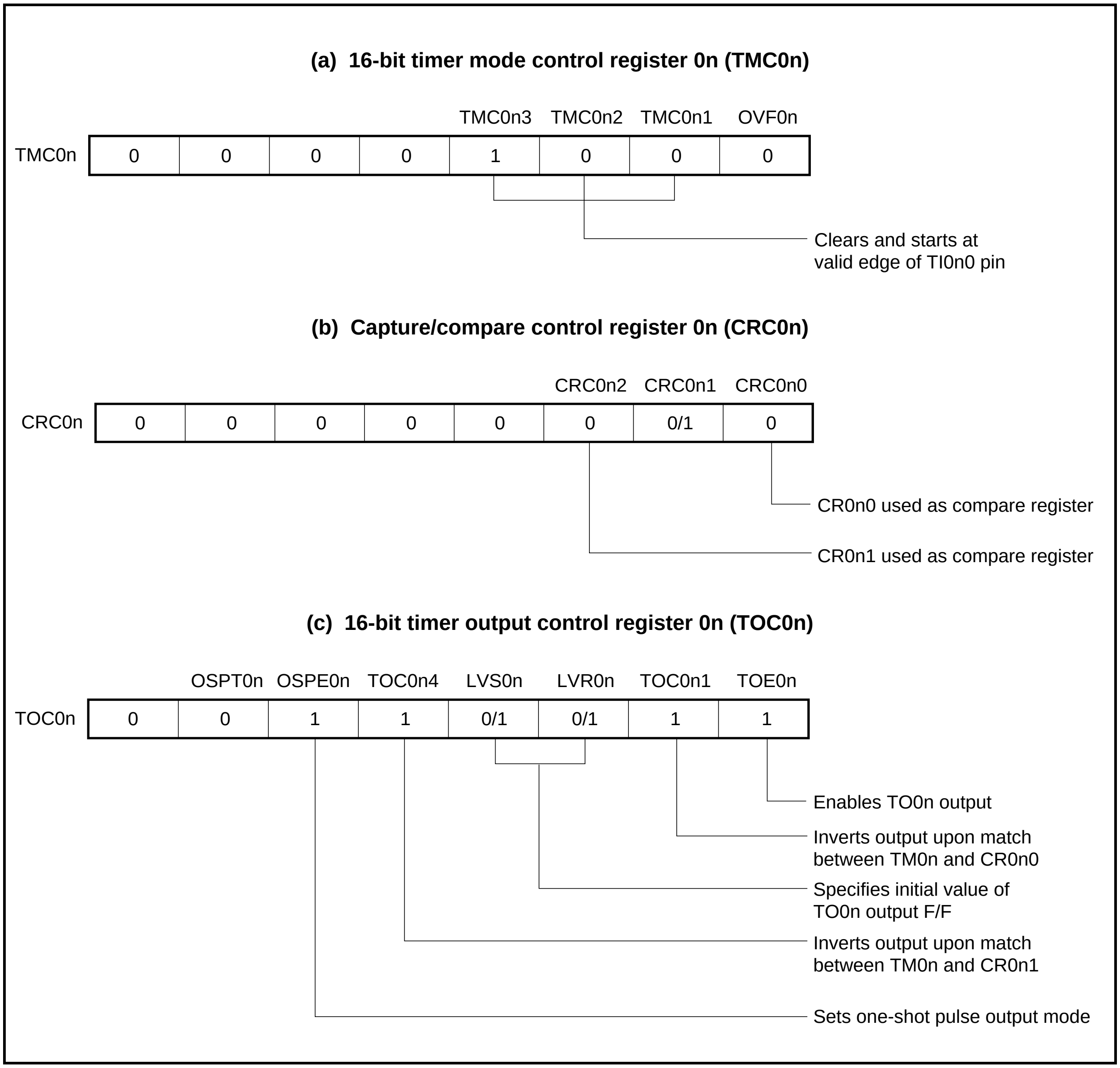
Note The case where N < M is described here. When N > M, the output becomes active with the CR0n0 register and inactive with the CR0n1 register.

Cautions 1. If the external trigger is generated again while a one-shot pulse is being output, the timer is cleared and started.

2. The value of the CR0n0 and CR0n1 registers cannot be changed during timer count operation.

Remark n = 0 to 3

Figure 7-24. Control Register Settings for One-Shot Pulse Output with External Trigger (with Rising Edge Specified) (1/2)



(a) 16-bit timer mode control register 0n (TMC0n)

Clears and starts at valid edge of TI0n0 pin

(b) Capture/compare control register 0n (CRC0n)

CR0n0 used as compare register

CR0n1 used as compare register

(c) 16-bit timer output control register 0n (TOC0n)

Enables TO0n output

Inverts output upon match between TM0n and CR0n0

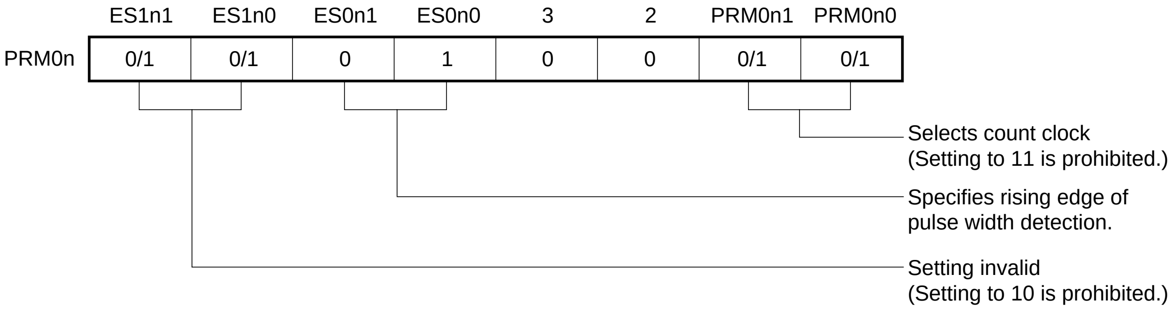
Specifies initial value of TO0n output F/F

Inverts output upon match between TM0n and CR0n1

Sets one-shot pulse output mode

Figure 7-24. Control Register Settings for One-Shot Pulse Output with External Trigger (with Rising Edge Specified) (2/2)

(d) Prescaler mode register 0n (PRM0n)

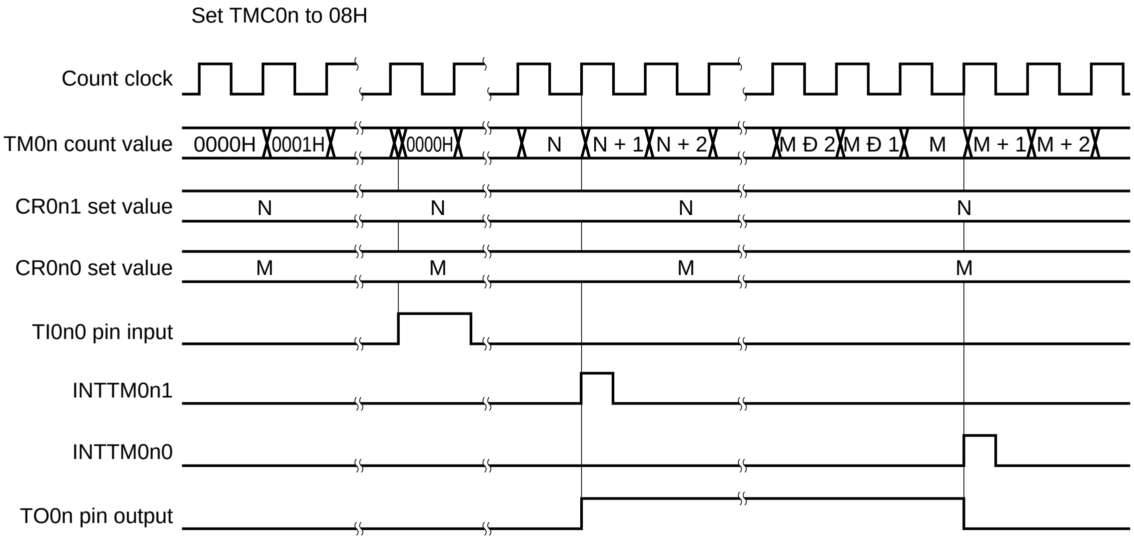


Caution **Do not set the CR0n0 register to 0000H.**

Remarks 1. For details, refer to **7.3 (2) Capture/compare control register 0n (CRC0n)** and **7.3 (3) 16-bit timer output control register 0n (TOC0n)**.

2. n = 0 to 3

Figure 7-25. Timing of One-Shot Pulse Output Operation with External Trigger (with Rising Edge Specified)



Caution **The TM0n register starts operating as soon as a value other than 00 (operation stop mode) is set to the TMC0n2 and TMC0n3 bits.**

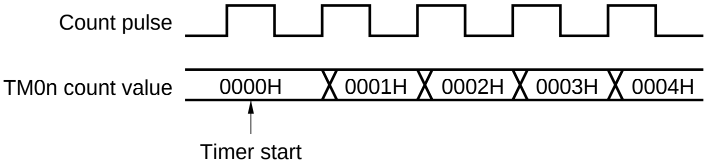
Remark n = 0 to 3
N < M

7.4.7 Cautions

(1) Error on starting timer

An error of up to 1 clock occurs before the match signal is generated after the timer has been started. This is because the TM0n register is started asynchronously to the count pulse.

Figure 7-26. Start Timing of TM0n Register



Remark n = 0 to 3

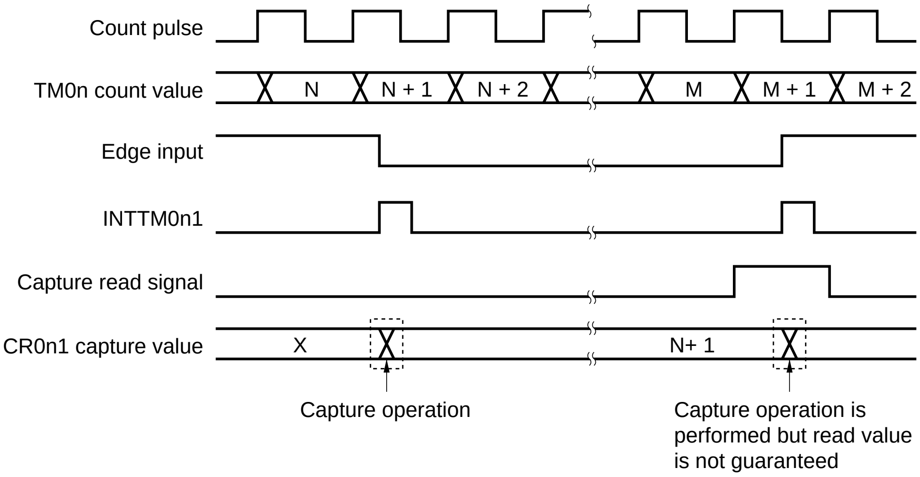
(2) Setting 16-bit timer capture/compare register (in the mode in which clear & start occurs upon match between TM0n register and CR0n0 register)

Set the CR0n0 register to a value other than 0000H (when using this register as an event counter, one-pulse count operation is not possible).

(3) Data hold timing of capture register

<1> If the valid edge of the TI0n0 pin is input while the CR0n1 register is being read, the CR0n1 register performs the capture operation. At this time, the captured value is guaranteed but the read value is not. However, an interrupt request signal (INTTM0n1) is generated as a result of detection of the valid edge.

Figure 7-27. Data Hold Timing of Capture Register



Remark n = 0 to 3

★ <2> The values of the CR0n0 and CR0n1 registers are not guaranteed after 16-bit timer/event counter 0n has stopped.

(4) Setting valid edge

Before setting the valid edge of the TI0n0 and TI0n1 pins, stop the timer operation by setting the TMC0n.TMC0n2 and TMC0n.TMC0n3 bits to 00. Set the valid edge by using the PRM0n.ES0n0, PRM0n.ES0n1, PRM0n.ES1n0, and PRM0n.ES1n1 bits.

The TI0n0 and TI0n1 pins function alternately as the P98/A8 to P915/A15 pins. To use the TI0n0 and TI0n1 pins, select the timer input function by using the PMC9m and PFC9m bits before enabling the timer operation with the TMC0n register. If the PMC9m and PFC9m bits are manipulated after the timer operation, the edge cannot be detected correctly.

Remark n = 0 to 3, m = 8 to 15

(5) Re-triggering one-shot pulse

(a) One-shot pulse output by software (TM00 to TM03)

When a one-shot pulse is output, do not set the TOC0n.OSPT0n bit (1). To output the one-shot pulse again, wait until the current one-shot pulse output is completed.

Remark n = 0 to 3

(b) One-shot pulse output with external trigger

If an external trigger is generated again while a one-shot pulse is being output, the timer is cleared and started.

(c) One-shot pulse output function

When using the one-shot pulse output of timer 0 with a software trigger, do not change the level of the TI0n0 pin or its alternate function port pin.

Because the external trigger is effective even in this case, the timer is cleared and started even with the TI0n0 pin or its alternate function port pin level, resulting in the output of a pulse at an undesired timing.

With the V850ES/PM1, the internal TI0n0 signal level is fixed to low level when a function other than “timer input function” is selected by using the PMC9 and PFC9 registers.

Remark n = 0 to 3

(6) Operation of OVF0n flag

(a) Setting of OVF0n flag

The OVF0n flag is set to 1 in the following case.

Select either the mode in which clear & start occurs upon match between the TM0n register and the CR0n0 register, the mode in which clear & start occurs upon detection of the valid edge of the TI0n0 register, or the free-running mode.

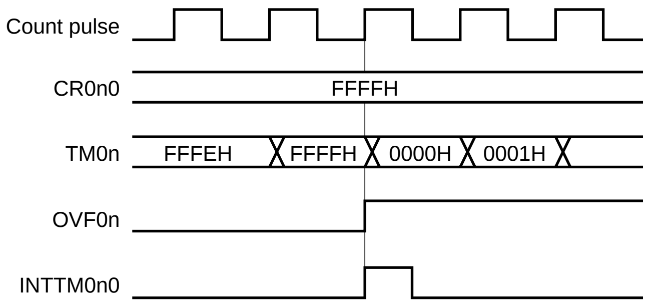
↓

Set the CR0n0 register to FFFFH

↓

When the TM0n register counts up from FFFFH to 0000H

Figure 7-28. Operation Timing of OVF0n Flag



Remark n = 0 to 3

(b) Clearing of OVF0n flag

After the TM0n register overflows, clearing OVF0n flag is invalid and set again even if the OVF0n flag is cleared before the next count clock is counted (before TM0n register becomes 0001H).

Remark n = 0 to 3

(7) Timer operation

(a) CR0n1 register capture

Even if the TM0n register is read, the read data cannot be captured into the CR0n1 register.

(b) TI0n0, TI0n1 pin acknowledgement

Regardless of the CPU's operation mode, if the timer is stopped, signals input to the TI0n0 and TI0n1 pins are not acknowledged.

(c) One-shot pulse output (TM00 to TM03)

One-shot pulse output operates normally in either the free-running mode or the mode in which clear & start occurs on the valid edge of the TI0n0 pin. Because no overflow occurs in the mode in which clear & start occurs upon match between the TM0n register and the CR0n0 register, one-shot pulse output is not possible.

Remark n = 0 to 3

(8) Capture operation

(a) If valid edge of TI0n0 pin is specified for count clock

If the valid edge of the TI0n0 pin is specified for the count clock, the capture register that specified the TI0n0 pin as the trigger does not operate normally.

(b) If both rising and falling edges are specified as valid edge of TI0n0 pin

★ If both the rising and falling edges are specified as the valid edge of the TI0n0 pin and the capture trigger of the CR0n0 register is specified as the inverse edge of the TI0n0 valid edge, the capture operation is not performed.

(c) To ensure that signals from TI0n1 and TI0n0 pins are correctly captured

For the capture trigger to capture the signals from the TI0n1 and TI0n0 pins correctly, a pulse longer than two of the count clocks selected by the PRM0n register is required.

(d) Interrupt request input

Although a capture operation is performed at the falling edge of the count clock, an interrupt request signal (INTTM0n0, INTTM0n1) is generated at the rising edge of the next count clock.

Remark n = 0 to 3

(9) Compare operation

When set in the compare mode, the CR0n0 and CR0n1 registers do not perform capture operation even if a capture trigger is input.

Remark n = 0 to 3

(10) Edge detection

The sampling clock for noise elimination differs depending on whether the valid edge of the TI0n0 pin is used for the count clock or as a capture trigger. In the former case, sampling is performed using f_{XX}, and in the latter case, sampling is performed using the count clock selected by the PRM0n register. The first capture operation does not start until the valid edges are sampled and two valid levels are detected, thus eliminating noise with a short pulse width.

Remarks 1. f_{XX}: Main clock frequency

2. n = 0 to 3

CHAPTER 8 16-BIT TIMER/EVENT COUNTERS 10 AND 11

8.1 Features

16-bit timer/event counters 10 and 11 can perform the following operations.

- Interval timer function
- PWM output
- External signal cycle measurement

8.2 Function Overview

- 16-bit timer/counter
- Capture/compare common registers: 2 × 2 channels
- Interrupt request sources
  - Capture/match interrupt requests: 2 × 2 channels
  - Overflow interrupt requests: 1 × 2 channels
- Timer/counter count clock sources: 2
  (Selection of external pulse input or internal system clock division)
- Either free-running mode or overflow stop mode can be selected as the operation mode when the timer/counter overflows
- Timer/counter can be cleared by a match of the timer/counter and a compare register
- External pulse outputs: 1 × 2 channels

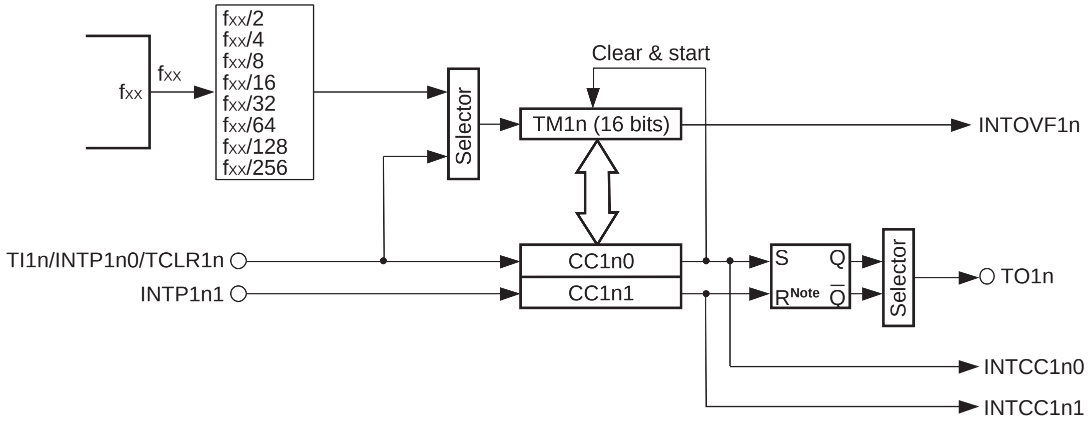
8.3 Configuration

Table 8-1. Configuration of 16-Bit Timer/Event Counters 10, 11

Timer	Count Clock	Register	Read/Write	Generated Interrupt Signal	Capture Trigger	Timer Output S/R
TM10, TM11	fxx/2, fxx/4, fxx/8, fxx/16, fxx/32, fxx/64, fxx/128, fxx/256	TM10	Read	INTOVF10	–	–
		CC100	Read/write	INTCC100	INTP100	TO10 (S)
		CC101	Read/write	INTCC101	INTP101	TO10 (R)
		TM11	Read	INTOVF11	–	–
		CC110	Read/write	INTCC110	INTP110	TO11 (S)
		CC111	Read/write	INTCC111	INTP111	TO11 (R)

Remark fxx: Main clock frequency
S/R: Set/reset

Figure 8-1. Block Diagram of 16-Bit Timer/Event Counter 1n



Note Reset priority

Remarks 1. n = 0, 1

2. fxx: Main clock frequency

(1) 16-bit timer counters 10 and 11 (TM10 and TM11)

The TM1n register functions as a 16-bit free-running timer or as an event counter for an external signal. Besides being used for cycle measurement, TM1n can be used for pulse output (n = 0, 1).

The TM1n register is read-only, in 16-bit units.

Cautions 1. The TM1n register can only be read. If the TM1n register is written, the subsequent operation is undefined.

2. If the TMC1n0.TM1CAEn bit is cleared (0), a reset is performed asynchronously.

★ **3. When the main clock is stopped and the CPU is operating on the subclock, do not access the TM1n register using an access method that causes a wait.**
For details, refer to 3.4.8 (2).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
TM10																	FFFFF600H	0000H
TM11																	FFFFF610H	0000H

(a) Selection of count clock

The TM1n register performs the count-up operation of an internal count clock or external count clock. Timer start and stop are controlled by the TMC1n0.TM1CEn bit (n = 0, 1).

The internal or external count clock is selected by the TMC1n1.ETI1n bit (n = 0, 1).

(i) Selection of the external count clock

The TM1n register operates as an event counter.

When the TMC1n1.ETI1n bit is set (1), the TM1n register counts the valid edges of the external clock input (TI1n), synchronized with the internal count clock. The valid edge is specified by the SES1n register (n = 0, 1).

Caution When the INTP1n0/TI1n/TCLR1n pin is used as TI1n (external clock input pin), disable the INTP1n0 pin interrupt and set the CC1n0 register to compare mode (n = 0, 1).

(ii) Selection of the internal count clock

The TM1n register operates as a free-running timer.

When the internal clock is specified as the count clock by the TMC1n1 register, TM1n is counted up for each input clock cycle specified by the TMC1n0.CS1n0 to TMC1n0.CS1n2 bits (n = 0, 1).

Division by the prescaler can be selected for the count clock from among $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$, and $f_{XX}/256$ by the TMC1n0 register (f_{XX}: Internal system clock).

An overflow interrupt can be generated if the timer overflows. Also, the timer can be stopped following an overflow by setting the TMC1n1.OST1n bit (1).

Caution The count clock cannot be changed while the timer is operating.

(b) Conditions when TM1n register becomes 0000H.

(i) Asynchronous reset

- TMC1n0.TM1CAEn bit = 0
- Reset input

(ii) Synchronous reset

- TMC1n0.TM1CEn bit = 0
- The CC1n0 register is used as a compare register, and the TM1n and CC1n0 registers match when clearing the TM1n register is enabled (TMC1n1.CCLR1n bit = 1)

(2) 16-bit timer capture/compare registers 1n0 and 1n1 (CC1n0 and CC1n1) (n = 0, 1)

The CCIn0 and CC1n1 registers are 16-bit registers.

They can be used as capture registers or compare registers according to the TMC1n1.CMS1n0 and TMC1n1.CMS1n1 bit specifications (n = 0, 1).

These registers can be read or written in 16-bit units. (However, write operations can only be performed in compare mode.)

These registers are set to 0000H after reset.

★ **Caution When the main clock is stopped and the CPU is operating on the subclock, do not access the CC1n0 and CC1n1 registers using an access method that causes a wait.**
For details, refer to 3.4.8 (2).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CC1n0																	CRC100 FFFFF602H, CRC110 FFFFF612H	0000H
CC1n1																	CRC101 FFFFF604H, CRC111 FFFFF614H	0000H

Remark n = 0, 1

(a) Setting these registers as capture registers (TMC1n1.CMS1n0 and TMC1n1.CMS1n1 = 0)

When these registers are set as capture registers, the valid edges of the corresponding external interrupt signals INTP1n0 and INTP1n1 are detected as capture triggers. The timer TM1n is synchronized with the capture trigger, and the value of TM1n is latched in the CC1n0 and CC1n1 registers (capture operation).

The valid edge of the INTP1n0 pin is specified (rising, falling, or both rising and falling edges) according to the SES1n.IES1n01 and SES1n.IES1n00 bits, and the valid edge of the INTP1n1 pin is specified according to the IES1n11 and IES1n10 bits of the SES1n register (n = 0, 1).

The capture operation is performed asynchronously to the count clock. The latched value is held in the capture register until another capture operation is performed (n = 0, 1).

When the TMC1n0.TM1CAEn bit is 0, 0000H is read (n = 0, 1).

If these registers are specified as capture registers, an interrupt is generated by detecting the valid edge of signals INTP1n0 and INTP1n1 (n = 0, 1).

Caution If the capture operation conflicts with the timing of disabling the TM1n register from counting (when the TM1CEn bit of the TMC1n0 register = 0), the captured data becomes undefined. In addition, the INTCC1n0 and INTCC1n1 interrupts do not occur (n = 0, 1).

(b) Setting these registers as compare registers (TMC1n1.CMS1n0 and TMC1n1.CMS1n1 = 1)

When these registers are set as compare registers, the TM1n register and compare register values are compared for each count clock, and an interrupt is generated by a match. If the TMC1n1.CCLR1n bit is set (1), the TM1n value is cleared (0000H) at the same time as a match with the CC1n0 register (it is not cleared (0000H) by a match with the CC1n1 register) (n = 0, 1).

Compare registers are equipped with a set/reset function. The corresponding timer output (TO1n) is set or reset, in synchronization with the generation of a match signal (n = 0, 1).

The interrupt selection source differs according to the function of the selected register.

Cautions 1. When writing to the CC1n0 and CC1n1 registers, always set the TM1CAEn bit to 1 first. If the TM1CAEn bit is 0, the data that is written will be invalid.

2. Write to the CC1n0 and CC1n1 registers after setting them as compare registers via TMC1n0 and TMC1n1 register settings. If they are set as capture registers (TMC1n1.CMS1n0 and TMC1n1.CMS1n1 bits = 0), no data is written even if a write operation is performed to the CC1n0 and CC1n1 registers.

3. When these registers are set as compare registers, the INTP1n0 and INTP1n1 pins cannot be used as capture trigger input pins (n = 0, 1).

8.4 Registers

(1) 16-bit timer mode control registers 100 and 110 (TMC100 and TMC110)

The TMC1n0 registers control the operation of 16-bit timer/event counter 1n (n = 0, 1).

These registers can be read or written in 8-bit or 1-bit units.

These resisters are set to 00H after reset.

Be sure to set bits 3 and 2 to 0. If they are set to 1, the operation is not guaranteed.

Cautions 1. The TM1CAEn bit cannot be set at the same time as the other bits. The other bits and the registers of the other TM1n units should always be set after the TM1CAEn bit has been set. Also, to use external pins related to the timer function when the 16-bit timer/event counter is used, be sure to set (1) the TM1CAEn bit after setting the external pins to control mode.

2. When conflict occurs between an overflow and a TMC1n0 register write, the OVF1n bit value is not guaranteed (n = 0, 1).

★ **3. When the main clock is stopped and the CPU is operating on the subclock, do not access the TMC1n0 register using an access method that causes a wait.**
For details, refer to 3.4.8 (2).

(1/2)

After reset: 00H R/W Address: TMC100 FFFFF606H, TMC110 FFFFF616H

	<7>	6	5	4	3	2	<1>	<0>
TMC1n0 (n = 0, 1)	OVF1n	CS1n2	CS1n1	CS1n0	0	0	TM1CEn	TM1CAEn

OVF1n	TM1n register overflow detection
0	No overflow occurred
1	Overflow occurred

When TM1n has counted up from FFFFH to 0000H, the OVF1n bit becomes 1 and an overflow interrupt request (INTOVF1n) is generated at the same time. However, if TM1n is cleared to 0000H after a match at FFFFH when the CC1n0 register is set to compare mode (TMC1n1.CMS1n0 bit = 1) and clearing is enabled for a match when TM1n and CC1n0 are compared (TMC1n1.CCLR1n bit = 1), then TM1n is considered to be cleared and the OVF1n bit does not become 1. Also, no INTOVF1n interrupt is generated.
The OVF1n bit retains the value 1 until 0 is written directly or until an asynchronous reset is performed because the TM1CAEn bit is 0. An interrupt operation due to an overflow is independent of the OVF1n bit, and the interrupt request flag (OVFIF1n) for INTOVF1n is not affected even if the OVF1n bit is manipulated. If an overflow occurs while the OVF1n bit is being read, the flag value changes, and the change is reflected when the next read operation occurs.

(2/2)

CS1n2	CS1n1	CS1n0	Internal count clock selection
0	0	0	$f_{XX}/2$
0	0	1	$f_{XX}/4$
0	1	0	$f_{XX}/8$
0	1	1	$f_{XX}/16$
1	0	0	$f_{XX}/32$
1	0	1	$f_{XX}/64$
1	1	0	$f_{XX}/128$
1	1	1	$f_{XX}/256$

TM1CEn	TM1n register operation control
0	Count disabled (stops at 0000H and does not operate).
1	Counting operation is performed.

When TM1CEn = 0, the external pulse output (TO1n) becomes inactive (the active level of TO1n output is set by the ALV1n bit of the TMC1n1 register).

TM1CAEn	Internal count clock control
0	The entire TM1n unit is asynchronously reset. The supply of clocks to the TM1n unit stops.
1	Clocks are supplied to the TM1n unit.

- When the TM1CAEn bit is set to 0, the TM1n unit can be asynchronously reset.
- When TM1CAEn = 0, the TM1n unit is in a reset state. Therefore, to operate TM1n, the TM1CAEn bit must be set to 1.
- When the TM1CAEn bit is changed from 1 to 0, all registers of the TM1n unit are initialized. When the TM1CAEn bit is set to 1 again, the TM1n unit registers must be set again.

(2) 16-bit timer mode control registers 101 and 111 (TMC101 and TMC111)

The TMC1n1 registers control the operation of 16-bit timer/event counter 1n (n = 0, 1).

These registers can be read or written in 8-bit or 1-bit units.

Cautions 1. The various bits of the TMC1n1 register must not be changed during timer operation. If they are to be changed, they must be changed after setting the TM1CEn bit of the TMC1n0 register to 0. If these bits are overwritten during timer operation, operation cannot be guaranteed (n = 0, 1).

2. If the ENTO1n and ALV1n bits are changed at the same time, a glitch (spike shaped noise) may be generated in the TO1n pin output. Either create a circuit configuration that will not malfunction even if a glitch is generated or make sure that the ENTO1n and ALV1n bits do not change at the same time (n = 0, 1).

3. TO1n output is not changed by an external interrupt signal (INTP1n0 or INTP1n1). To use the TO1n signal, specify that the capture/compare registers are compare registers (CMS1n0 and CMS1n1 bits of TMC1n1 register = 1) (n = 0, 1).

(1/2)

After reset: 20H R/W Address: TMC101 FFFFF608H, TMC111 FFFFF618H

	7	6	5	4	3	2	1	0
TMC1n1 (n = 0, 1)	OST1n	ENTO1n	ALV1n	ETI1n	CCLR1n	ECLR1n	CMS1n1	CMS1n0

OST1n	Setting of operation when TM1n register overflowed
0	After the overflow, counting continues (free-running mode).
1	After the overflow, the timer maintains the value 0000H, and counting stops (overflow stop mode).
When OST1n bit = 1, the TMC1n0.TM1CEn bit remains at 1. Counting is restarted by writing 1 to the TM1CEn bit.	

ENTO1n	External pulse output (TO1n) enable/disable
0	External pulse output is disabled.
1	External pulse output is enabled.
• When ENTO1n bit = 0, output of the ALV1n bit inactive level to the TO1n pin is fixed. The TO1n pin level is not changed even if a match signal from the corresponding compare register is generated. • When ENTO1n bit = 1, a compare register match causes TO1n output to change. However, if capture mode is set, TO1n output does not change. The ALV1n bit inactive level is output from the time when timer output is enabled until a match signal is first generated. • If either CC1n0 or CC1n1 is specified as a capture register, the ENTO1n bit must be set to 0.	

(2/2)

ALV1n	External pulse output (TO1n) active level specification
0	Low level
1	High level
The initial value of the ALV1n bit is 1.	

ETI1n	Count clock external/internal switch specification
0	Specifies the input clock (internal).
1	Specifies the external clock (TI1n0).
• When ETI1n bit = 0, the internal count clock can be selected according to the TMC1n0.CS1n2 to TMC1n0.CS1n0 bits. • When ETI1n bit = 1, the valid edge can be selected according to the SES1n.TES1n1 and SES1n.TES1n0 bit specifications.	

CCLR1n	TM1n register clear enable/disable specification during compare operation
0	Clearing is disabled
1	Clearing is enabled (if CC1n0 and TM1n match during a compare operation, TM1n is cleared)

ECLR1n	TM1n register clear enable/disable specification by external clear input (TCLR1n)
0	Clearing is disabled
1	Clearing is enabled (after the clearing, restarts counting)

CMS1n1	16-bit timer capture/compare register (CC1n1) operation mode selection
0	The register operates as a capture register.
1	The register operates as a compare register.

CMS1n0	16-bit timer capture/compare register (CC1n0) operation mode selection
0	The register operates as a capture register.
1	The register operates as a compare register.

Remark A reset takes precedence for the flip-flop of the TO1n output (n = 0, 1).

(3) Valid edge select registers 10 and 11 (SES10 and SES11)

These registers specify the valid edge of an external interrupt request (INTP100, INTP101, INTP110, INTP111, TI10, TI11, TCLR10, and TCLR11) from an external pin.

The rising edge, the falling edge, or both rising and falling edges can be specified as the valid edge independently for each pin.

Each of these registers can be read or written in 8-bit units.

After reset, these registers are set to 00H.

Caution The various bits of the SES1n register must not be changed during timer operation. If they are to be changed, they must be changed after setting the TMC1n0.TM1CEn bit to 0. If the SES1n register is overwritten during timer operation, operation cannot be guaranteed.

After reset: 00H R/W Address: SES10 FFFFF609H, SES11 FFFFF619H

	7	6	5	4	3	2	1	0
SES1n (n = 0, 1)	TES1n1	TES1n0	CES1n1	CES1n0	IES1n11	IES1n10	IES1n01	IES1n00

TES1n1	TES1n0	Valid edge of TI1n pin
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

CES1n1	CES1n0	Valid edge of TCLR1n pin
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

IES1n11	IES1n10	Valid edge of INTP1n1 pin
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

IES1n01	IES1n00	Valid edge of INTP1n0 pin
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

8.5 Operation

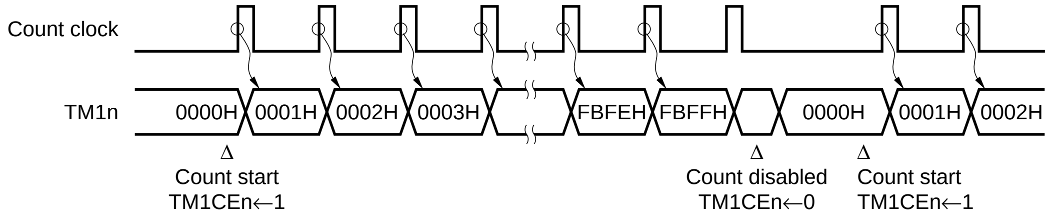
(1) Count operation

16-bit timer/event counter 1n can function as a 16-bit free-running timer or as an external signal event counter. The setting for the type of operation is specified by the TMC1n0 and TMC1n1 registers (n = 0, 1).

When it operates as a free-running timer, if the CC1n0 or CC1n1 register and the TM1n register count value match, an interrupt signal is generated and the timer output signal (TO1n) can be set or reset. Also, a capture operation that holds the TM1n register count value in the CC1n0 or CC1n1 register is performed, in synchronization with the valid edge that was detected from the external interrupt request input pin as an external trigger. The capture value is held until the next capture trigger is generated.

Caution When using the INTP1n0/TI1n0 pin as an external clock input pin (TI1n0), be sure to disable the INTP1n0 interrupt and set CC1n0 to compare mode (n = 0, 1).

Figure 8-2. Basic Operation of 16-Bit Timer/Event Counter



Remark n = 0, 1

(2) Overflow

When the TM1n register has counted the count clock from FFFFH to 0000H, the OVF1n bit of the TMC1n0 register is set (1), and an overflow interrupt (INTOVF1n) is generated at the same time (n = 0, 1). However, if the CC1n0 register is set to compare mode (TMC1n1.CMS1n0 bit = 1) and to the value FFFFH when match clearing is enabled (TMC1n1.CCLR1n bit = 1), then the TM1n register is considered to be cleared and the OVF1n bit is not set (1) when the TM1n register changes from FFFFH to 0000H. Also, the overflow interrupt (INTOVF1n) is not generated .

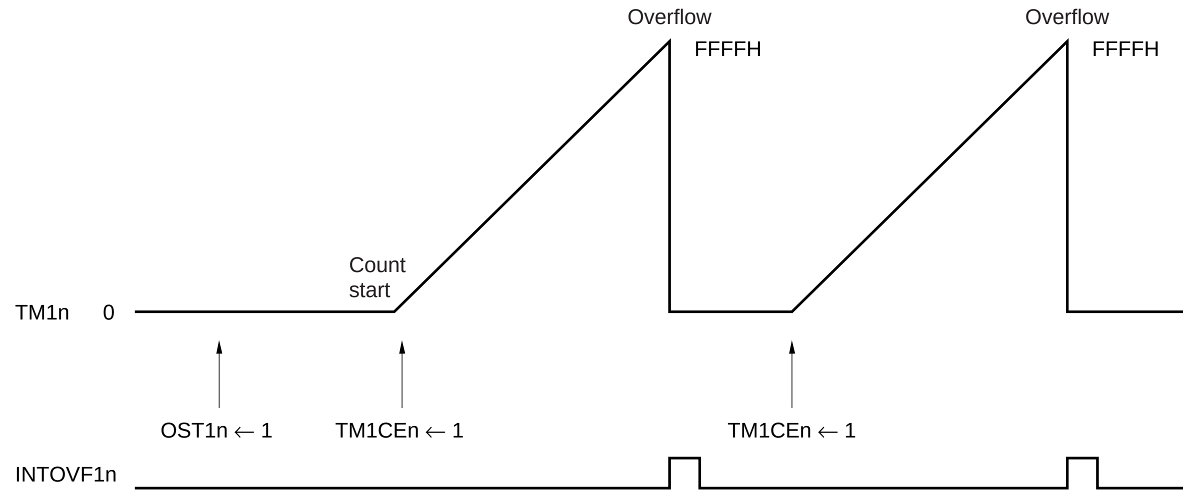
When the TM1n register is changed from FFFFH to 0000H because the TMC1n0.TM1CEn bit changes from 1 to 0, the TM1n register is considered to be cleared, but the OVF1n bit is not set (1) and no INTOVF1n interrupt is generated.

Also, timer operation can be stopped after an overflow by setting the TMC1n1.OST1n bit to 1. When the timer is stopped due to an overflow, the count operation is not restarted until the TM1CEn bit is set (1).

Operation is not affected even if the TM1CEn bit is set (1) during a count operation.

Remark n = 0, 1

Figure 8-3. Operation After Overflow (When OST1n = 1)



Remark n = 0, 1

(3) Capture operation

The TM1n register has two capture/compare registers. These are the CC1n0 register and the CC1n1 register. A capture operation or a compare operation is performed according to the settings of both the TMC1n1.CMS1n1 and TMC1n1.CMS1n0 bits. If the CMS1n1 and CMS1n0 bits of the TMC1n1 register are set to 0, the register operates as a capture register.

A capture operation that captures and holds the TM1n register count value asynchronously to the count clock is performed in synchronization with an external trigger. The valid edge that is detected from an external interrupt request input pin (INTP1n0 or INTP1n1) is used as an external trigger (capture trigger). The TM1n register count value during counting is captured and held in the capture register, in synchronization with that capture trigger signal. The capture register value is held until the next capture trigger is generated.

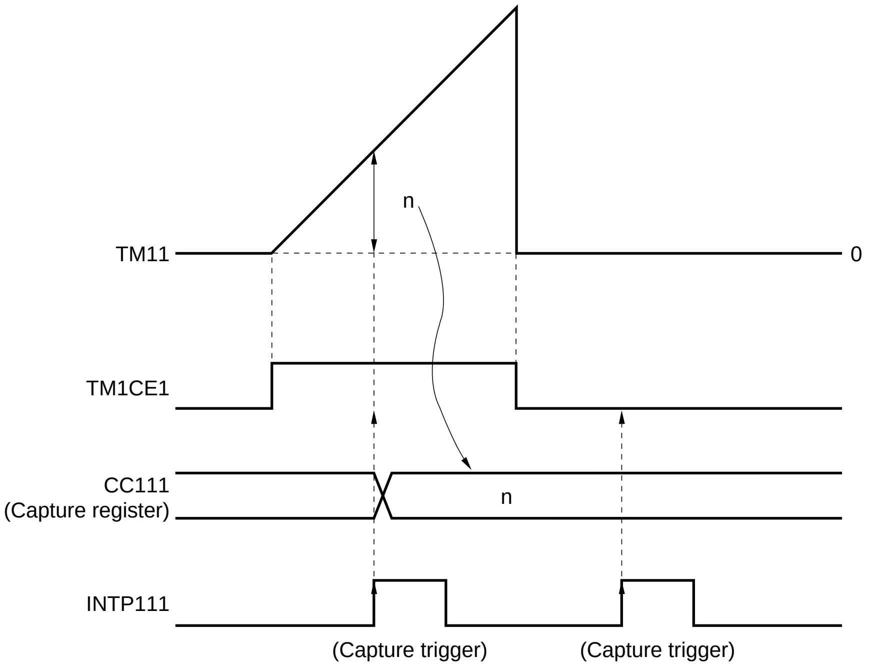
Also, an interrupt request (INTCC1n0 or INTCC1n1) is generated by INTP1n0 or INTP1n1 signal input.

The valid edge of the capture trigger is set by valid edge select register n (SES1n).

If both the rising and falling edges are set as capture triggers, the input pulse width from an external source can be measured. Also, if only one of the edges is set as the capture trigger, the input pulse cycle can be measured.

Remark n = 0, 1

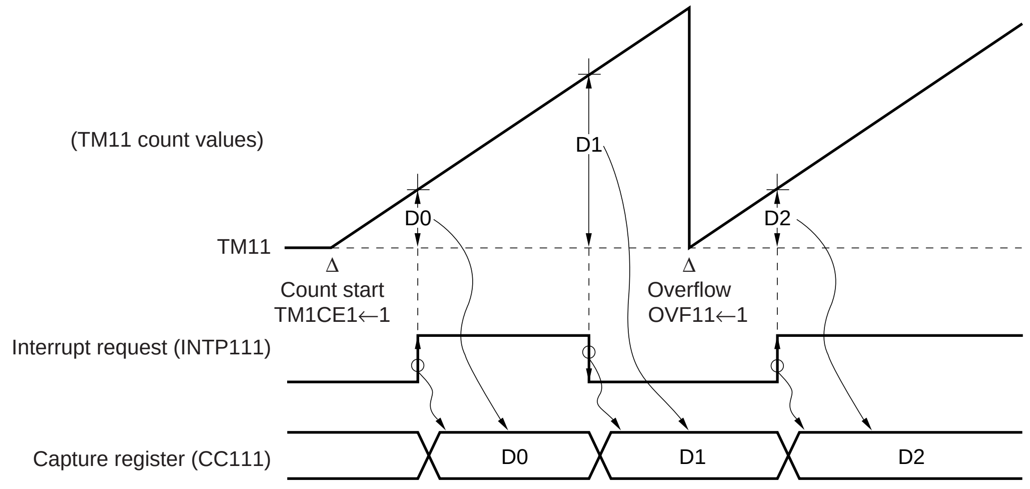
Figure 8-4. Capture Operation Example (TM11)



Remarks 1. When the TM1CE1 bit is 0, no capture operation is performed even if INTP111 is input.

2. Valid edge of INTP111: Rising edge

Figure 8-5. TM11 Capture Operation Example (When Both Edges Are Specified)



Remark D0 to D2: TM11 register count values

(4) Compare operation

16-bit timer/event counter 1n has two 16-bit timer capture/compare registers. These are the CC1n0 register and the CC1n1 register. A capture operation or a compare operation is performed according to the settings of both the TMC1n1.CMS1n1 and TMC1n1.CMS1n0 bits. If the TMC1n1.CMS1n1 and TMC1n1.CMS1n0 bits are set to 1, the register operates as a compare register.

A compare operation that compares the value that was set in the compare register and the TM1n register count value is performed.

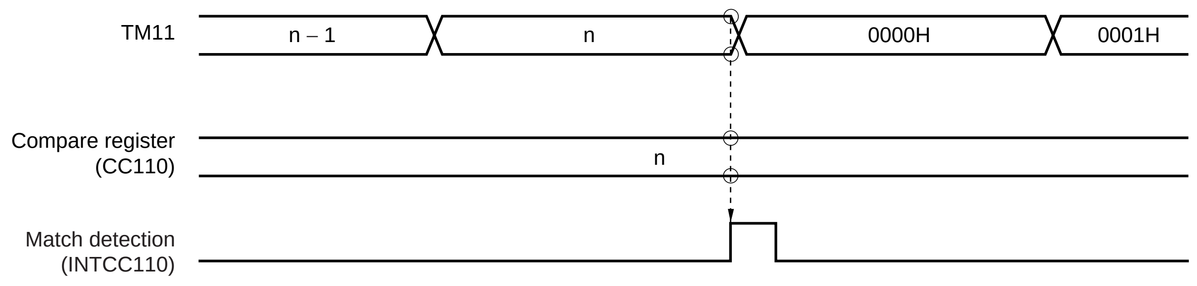
If the TM1n register count value matches the value of the compare register, which had been set in advance, a match signal is sent to the output controller. The match signal causes the timer output pin (TO1n) to change and an interrupt request signal (INTCC1nn) to be generated at the same time.

If the CC1n0 and CC1n1 registers are set to 0000H, the 0000H after the TM1n register counts up from FFFFH to 0000H is judged as a match. In this case, the TM1n register value is cleared (0000H) at the next count timing, however, this 0000H is not judged as a match. Also, the 0000H when the TM1n register begins counting is not judged as a match.

If match clearing is enabled (TMC1n1.CCLR1n bit = 1) for the CC1n0 register, the TM1n register is cleared when a match with the TM1n register occurs during a compare operation.

Remark n = 0, 1

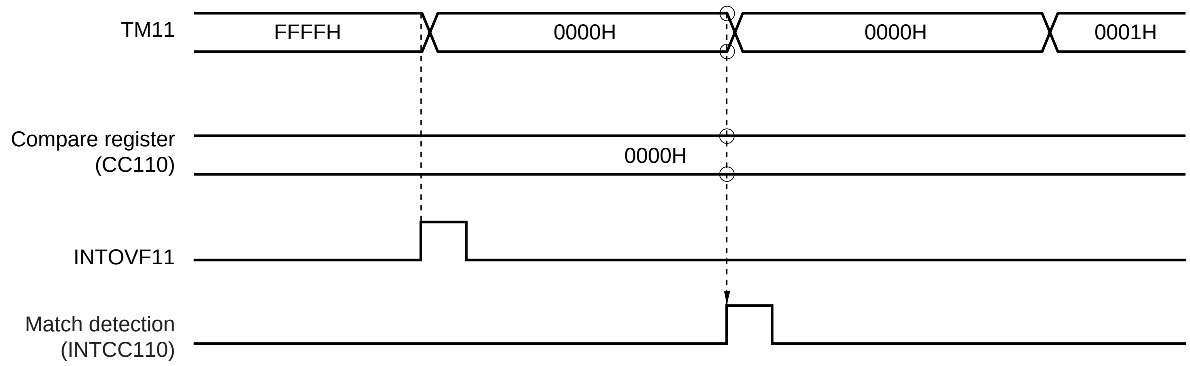
Figure 8-6. Compare Operation Example (When CCLR11 = 1 and CC110 Is Other Than 0000H)



Remarks 1. A match is detected immediately after the timer/event counter has counted up and a match detection signal is generated when the timer/event counter counts up next time.

2. n ≠ 0000H

Figure 8-7. Compare Operation Example (When CCLR11 = 1 and CC110 Is 0000H)



Remark A match is detected immediately after the timer/event counter has counted up and a match detection signal is generated when the timer/event counter counts up next time.

(5) External pulse output

16-bit timer/event counter 1n has two timer output pins (TO1n).

An external pulse output (TO1n) is generated when a match of the two compare registers (CC1n0 and CC1n1) and the TM1n register is detected.

If a match is detected when the TM1n register count value and the CC1n0 register value are compared, the output level of the TO1n pin is set. Also, if a match is detected when the TM1n register count value and the CC1n1 register value are compared, the output level of the TO1n pin is reset.

The output level of the TO1n pin can be specified by the TMC1n1 register.

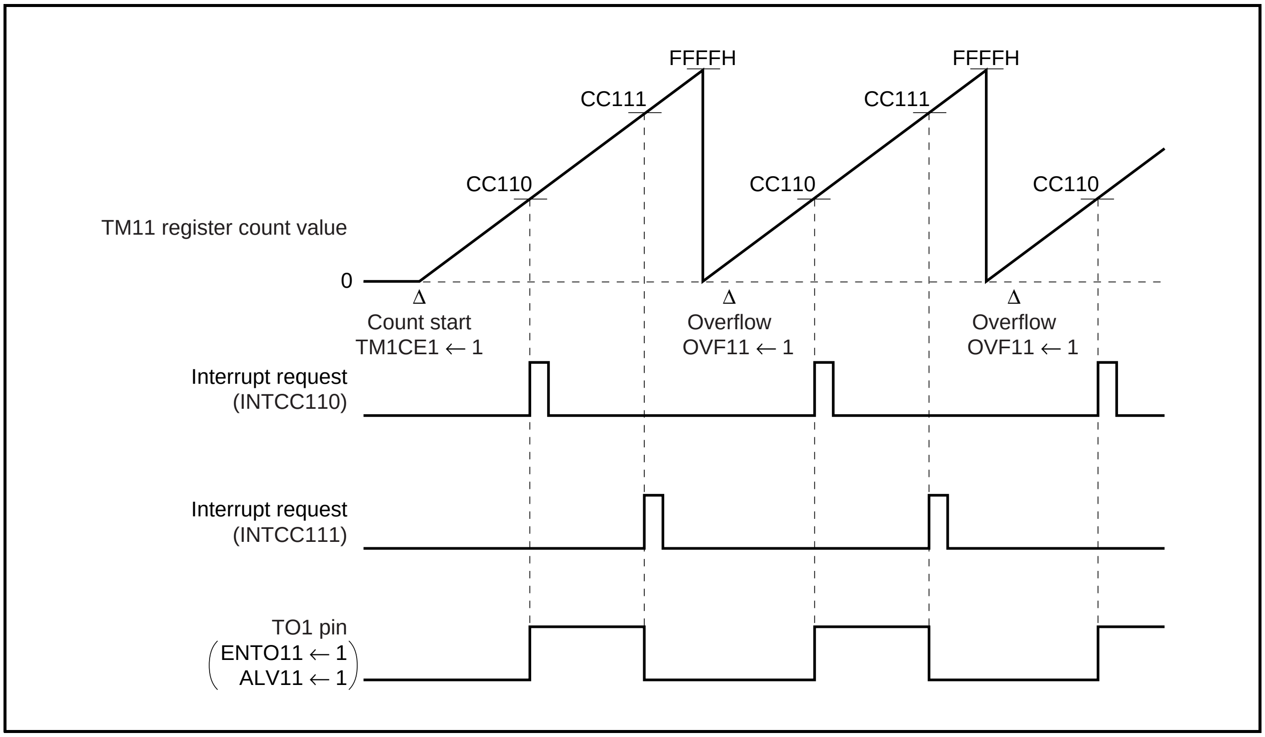
Remark n = 0, 1

Table 8-2. TO1n Output Control

ETI1n	ALV1n	TO1n Output	
		External Pulse Output	Output Level
0	0	Disable	High level
0	1	Disable	Low level
1	0	Enable	When the CC1n0 register is matched: low level When the CC1n1 register is matched: high level
1	1	Enable	When the CC1n0 register is matched: high level When the CC1n1 register is matched: low level

Remark n = 0, 1

Figure 8-8. TM11 Compare Operation Example (Set/Reset Output Mode)



8.6 Application Examples

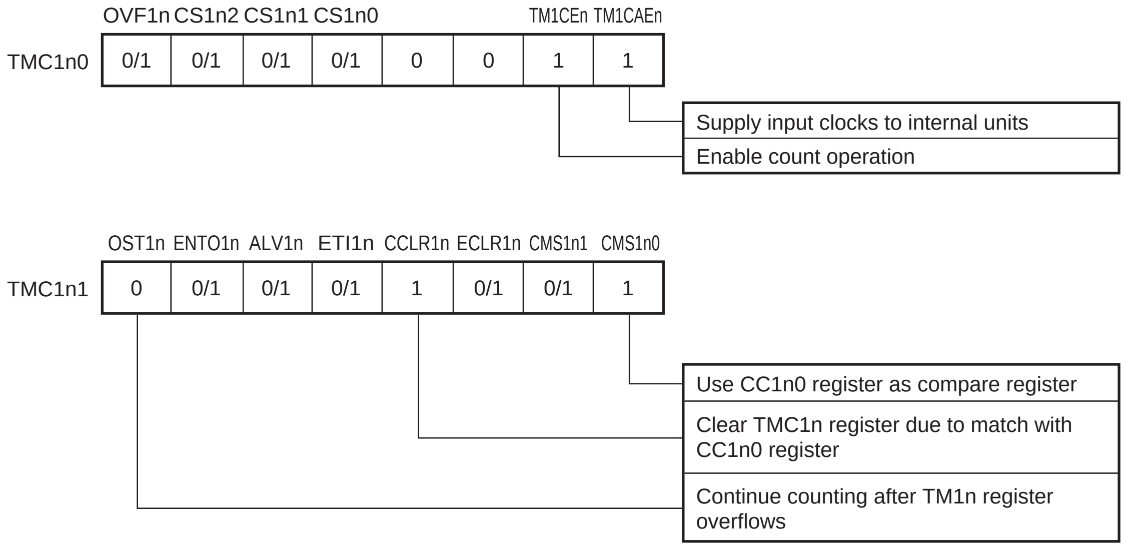
(1) Interval timer

By setting the TMC1n0 and TMC1n1 registers as shown in Figure 8-9, the 16-bit timer/event counter operates as an interval timer that repeatedly generates interrupt requests with the value that was preset in the CC1n0 register as the interval.

When the count value of the TM1n register matches the setting value of the CC1n0 register, the TM1n register is cleared (0000H) and an interrupt request signal (INTCC1n0) is generated at the same time that the count operation resumes.

Remark n = 0, 1

Figure 8-9. Register Settings When 16-Bit Timer/Event Counter Is Used as Interval Timer



	OVF1n	CS1n2	CS1n1	CS1n0			TM1CEn	TM1CAEn
TMC1n0	0/1	0/1	0/1	0/1	0	0	1	1

Supply input clocks to internal units

Enable count operation

	OST1n	ENTO1n	ALV1n	ETI1n	CCLR1n	ECLR1n	CMS1n1	CMS1n0
TMC1n1	0	0/1	0/1	0/1	1	0/1	0/1	1

Use CC1n0 register as compare register

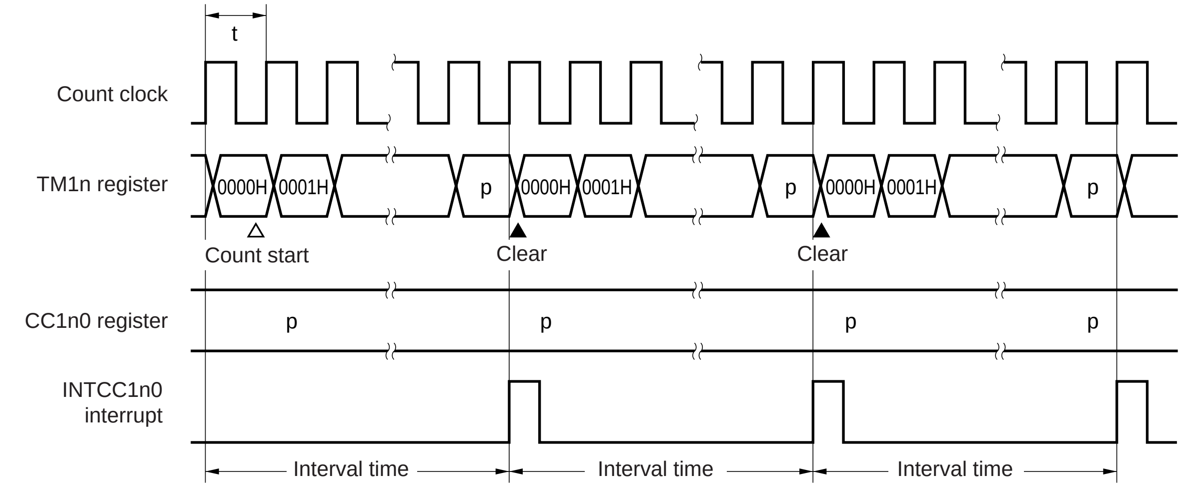
Clear TMC1n register due to match with CC1n0 register

Continue counting after TM1n register overflows

Remarks 1. 0/1: Set to 0 or 1 as necessary

2. n = 0, 1

Figure 8-10. Interval Timer Operation Timing Example



Remarks 1. p: Setting value of CC1n0 register (0000H to FFFFH)

t: Count clock cycle

Interval time = (p + 1) × t

2. n = 0, 1

(2) PWM output

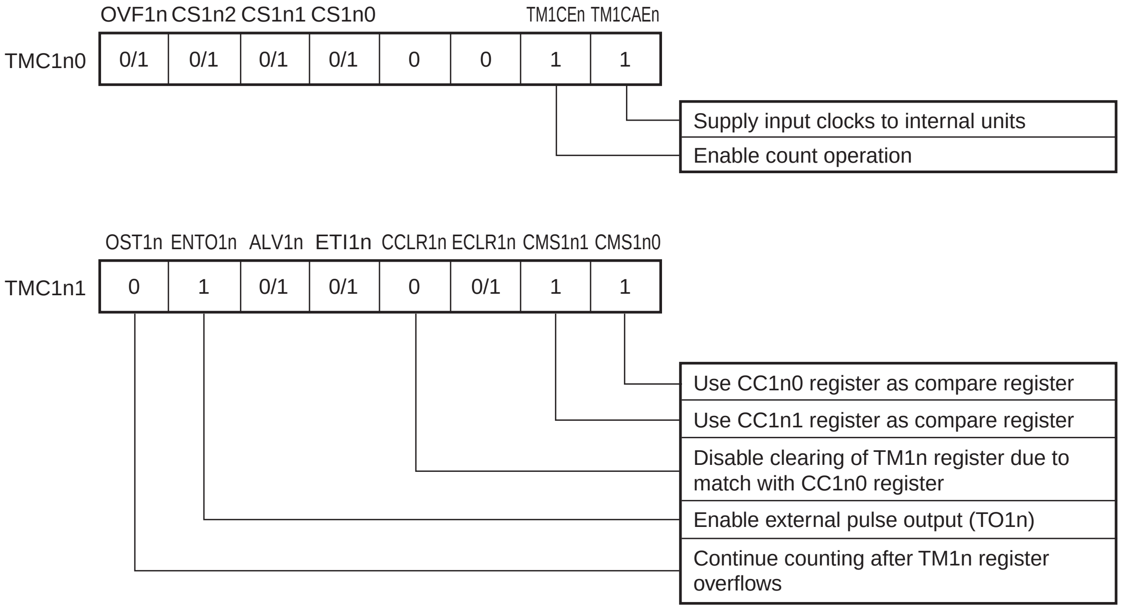
By setting the TMC1n0 and TMC1n1 registers as shown in Figure 8-11, the 16-bit timer/event counter can output a PWM signal, whose frequency is determined according to the setting of the TMC1n0.CS1n2 to TMC1n0.CS1n0 bits with the values that were preset in the CC1n0 and CC1n1 registers determining the intervals.

When the count value of the TM1n register matches the setting value of the CC1n0 register, the TO1n output becomes active. Then, when the counter value of the TM1n register matches the setting value of the CC1n1 register, the TO1n output becomes inactive. The TM1n register continues counting. When it overflows, its count value is cleared to 0000H, and the register continues counting. In this way, a PWM signal whose frequency is determined according to the setting of the CS1n2 to CS1n0 bits can be output. When the setting value of the CC1n0 register and the setting value of the CC1n1 register are the same, the TO1n output remains inactive and does not change.

The active level of the TO1n output can be set by the TMC1n1.ALV1n bit.

Remark n = 0, 1

Figure 8-11. Register Settings When 16-Bit Timer/Event Counter Is Used for PWM Output



	OVF1n	CS1n2	CS1n1	CS1n0			TM1CEn	TM1CAEn
TMC1n0	0/1	0/1	0/1	0/1	0	0	1	1

Supply input clocks to internal units

Enable count operation

	OST1n	ENTO1n	ALV1n	ETI1n	CCLR1n	ECLR1n	CMS1n1	CMS1n0
TMC1n1	0	1	0/1	0/1	0	0/1	1	1

Use CC1n0 register as compare register

Use CC1n1 register as compare register

Disable clearing of TM1n register due to match with CC1n0 register

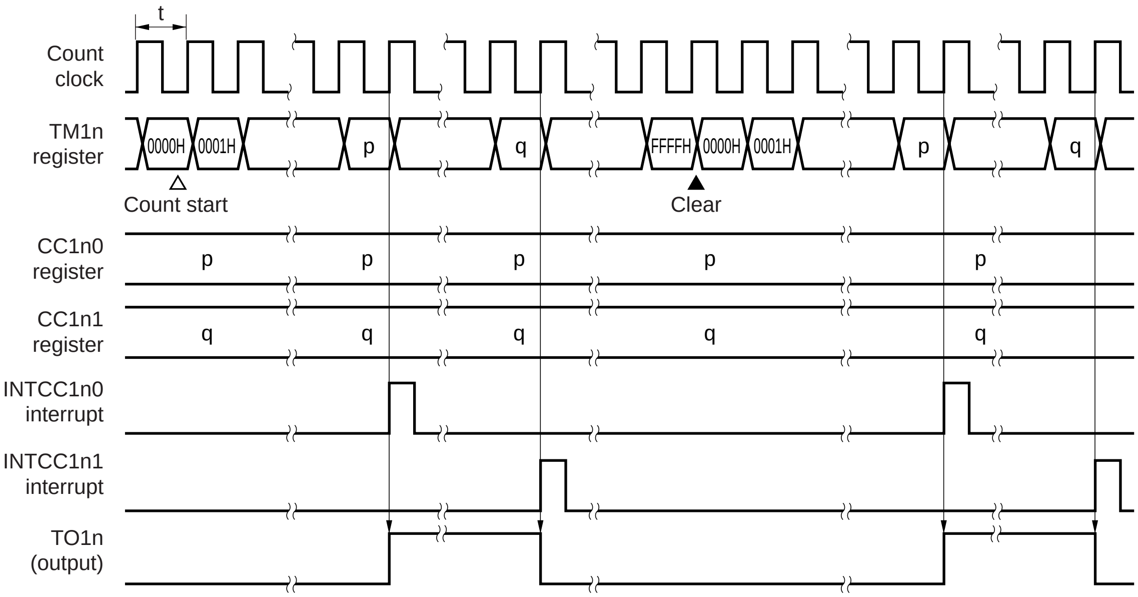
Enable external pulse output (TO1n)

Continue counting after TM1n register overflows

Remarks 1. 0/1: Set to 0 or 1 as necessary

2. n = 0, 1

Figure 8-12. PWM Output Operation Timing Example



Remarks 1. p: Setting value of CC1n0 register (0000H to FFFFH)

q: Setting value of CC1n1 register (0000H to FFFFH)

$p \neq q$

t: Count clock cycle

PWM cycle = 65,536 × t

$$\text{Duty} = \frac{q - p}{65{,}536}$$

2. In this example, the active level of the TO1n output is set to the high level.

3. n = 0, 1

(3) One-shot pulse output

By setting the TMC1n0 and TMC1n1 registers as shown in Figure 8-13, the 16-bit timer/event counter can output a one-shot pulse from the TO1n pin by using the valid edge of the TCLR1n pin as an external trigger.

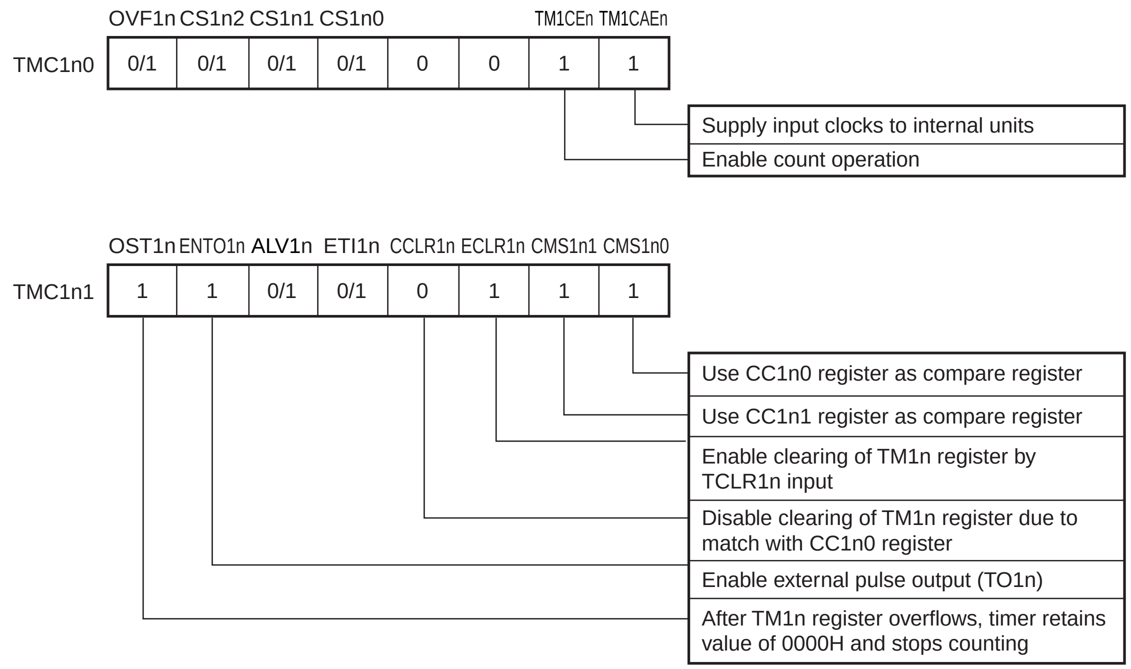
The valid edge of the TCLR1n pin is selected according to the SES1n.CES1n0 and SES1n.CES1n1 bits. The rising edge, falling edge, or both rising and falling edges can be selected as the valid edge.

The TM1n register is cleared and started by setting a valid edge to the TCLR1n pin. TO1n output becomes active at the count value set in advance to the CC1n0 register. After that, the TO1n output becomes inactive at the count value set in advance to CC1n1 register. The active level of the TO1n output can be set by the TMC1n1.ALV1n bit. When the setting value of the CC1n0 register and the setting value of the CC1n1 register are the same, the TO1n output remains inactive and does not change.

The active level of the TO1n output can be set by the TMC1n1.ALV1n bit.

Remark n = 0, 1

Figure 8-13. Register Settings When 16-Bit Timer/Event Counter Is Used for One-Shot Pulse Output



	OVF1n	CS1n2	CS1n1	CS1n0			TM1CEn	TM1CAEn
TMC1n0	0/1	0/1	0/1	0/1	0	0	1	1

Supply input clocks to internal units
Enable count operation

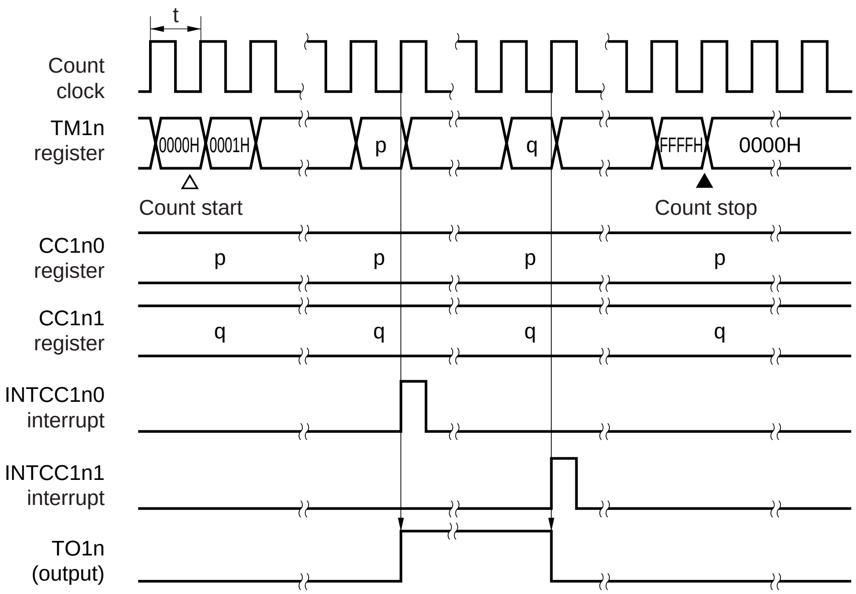
	OST1n	ENTO1n	ALV1n	ETI1n	CCLR1n	ECLR1n	CMS1n1	CMS1n0
TMC1n1	1	1	0/1	0/1	0	1	1	1

Use CC1n0 register as compare register
Use CC1n1 register as compare register
Enable clearing of TM1n register by TCLR1n input
Disable clearing of TM1n register due to match with CC1n0 register
Enable external pulse output (TO1n)
After TM1n register overflows, timer retains value of 0000H and stops counting

Remarks 1. 0/1: Set to 0 or 1 as necessary

2. n = 0, 1

Figure 8-14. One-Shot Pulse Output Operation Timing Example



Remarks 1. p: Setting value of CC1n0 register (0000H to FFFFH)

q: Setting value of CC1n1 register (0000H to FFFFH)

p ≠ q

t: Count clock cycle

2. In this example, the valid edge of the TCLR1n input is set to the rising edge and the active level of the TO1n output is set to the high level.

3. n = 0, 1

(4) Cycle measurement

By setting the TMC1n0 and TMC1n1 registers as shown in Figure 8-15, the 16-bit timer/event counter can measure the cycle of signals input to the INTP1n0 or INTP1n1 pin.

The valid edge of the INTP1n0 pin is selected according to the SES1n.IES1n01 and SES1n.IES1n00 bits, and the valid edge of the INTP1n1 pin is selected according to the SES1n.IES1n11 and SES1n.IES1n10 bits. Either the rising edge, the falling edge, or both edges can be selected as the valid edges of both pins.

If the CC1n0 register is set as a capture register, the valid edge input of the INTP1n0 pin is set as the trigger for capturing the TM1n register value in the CC1n0 register. When this value is captured, an INTCC1n0 interrupt is generated.

Similarly, if the CC1n1 register is set as a capture register, the valid edge input of the INTP1n1 pin is set as the trigger for capturing the TM1n register value in the CC1n1 register. When this value is captured, an INTCC1n1 interrupt is generated.

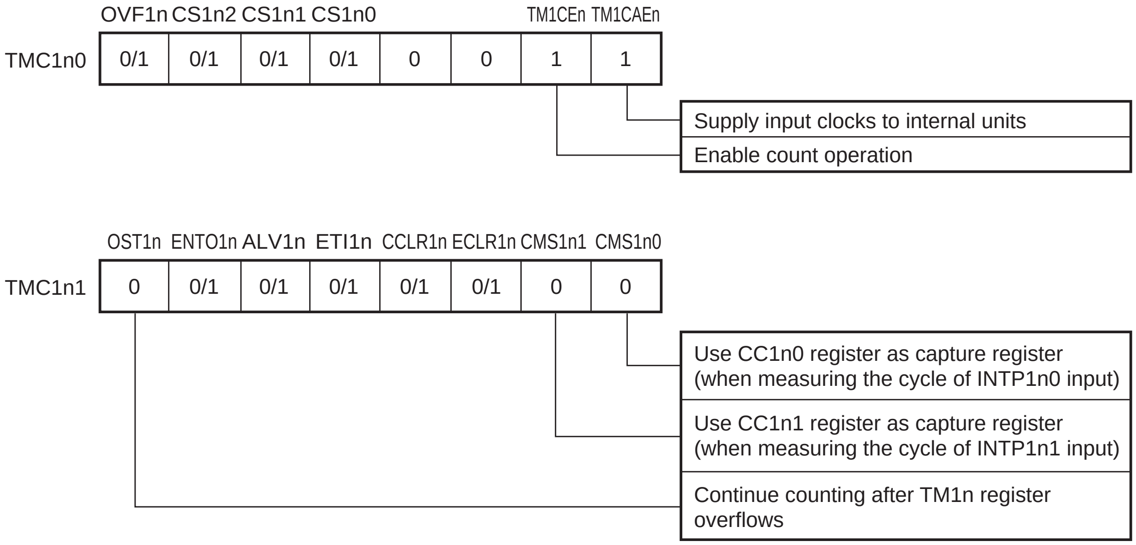
The cycle of signals input to the INTP1n0 pin is calculated by obtaining the difference between the TM1n register's count value (Dx) that was captured in the CC1n0 register according to the x-th valid edge input of the INTP1n0 pin and the TM1n register's count value (D(x+1)) that was captured in the CC1n0 register according to the (x+1)-th valid edge input of the INTP1n0 pin and multiplying the value of this difference by the cycle of the internal count clock[Note].

The cycle of signals input to the INTP1n1 pin is calculated by obtaining the difference between the TM1n register's count value (Dx) that was captured in the CC1n1 register according to the x-th valid edge input of the INTP1n1 pin and the TM1n register's count value (D(x+1)) that was captured in the CC1n1 register according to the (x+1)-th valid edge input of the INTP1n1 pin and multiplying the value of this difference by the cycle of the internal count clock[Note].

★ **Note** This calculation assumes that the rising/falling edges are selected.

Remark n = 0, 1

Figure 8-15. Register Settings When 16-Bit Timer/Event Counter Is Used for Cycle Measurement



	OVF1n	CS1n2	CS1n1	CS1n0			TM1CEn	TM1CAEn
TMC1n0	0/1	0/1	0/1	0/1	0	0	1	1

TM1CAEn: Supply input clocks to internal units

TM1CEn: Enable count operation

	OST1n	ENTO1n	ALV1n	ETI1n	CCLR1n	ECLR1n	CMS1n1	CMS1n0
TMC1n1	0	0/1	0/1	0/1	0/1	0/1	0	0

CMS1n0: Use CC1n0 register as capture register (when measuring the cycle of INTP1n0 input)

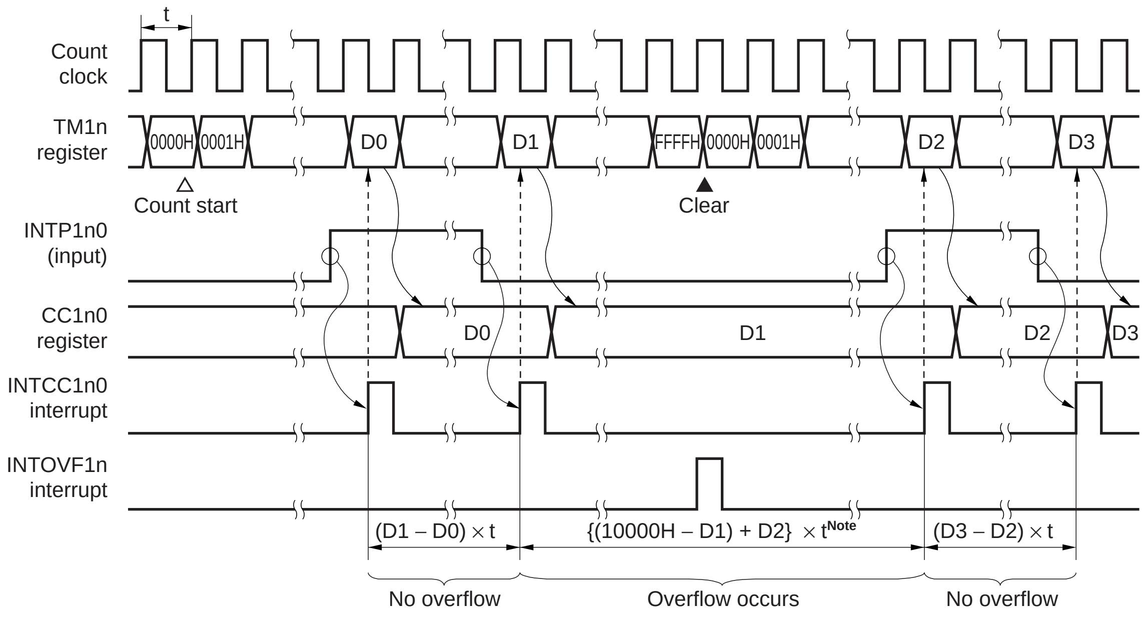
CMS1n1: Use CC1n1 register as capture register (when measuring the cycle of INTP1n1 input)

OST1n: Continue counting after TM1n register overflows

Remarks 1. 0/1: Set to 0 or 1 as necessary

2. n = 0, 1

Figure 8-16. Cycle Measurement Operation Timing Example



Note When an overflow occurs once.

Remarks 1. D0 to D3: TM1n register count values

t: Count clock cycle

2. In this example, the valid edge of the INTP1n0 input has been set to both edges (rising and falling).

3. n = 0, 1

8.7 Cautions

Various cautions concerning the 16-bit timer/event counter are shown below.

(1) If a conflict occurs between the reading of the CC1n0 register and a capture operation when the CC1n0 register is used in capture mode, an external trigger (INTP1n0) valid edge is detected and an interrupt request signal (INTCC1n0) is generated, however, the timer value is not stored in the CC1n0 register.

(2) If a conflict occurs between the reading of the CC1n1 register and a capture operation when the CC1n1 register is used in capture mode, an external trigger (INTP1n1) valid edge is detected and an interrupt request signal (INTCC1n1) is generated, however, the timer value is not stored in the CC1n1 register.

(3) The following bits and registers must not be rewritten during operation (TMC1n0.TM1CEn bit = 1).
- TMC1n0.CS1n2 to TMC1n0.CS1n0 bits.
- TMC1n1 register
- SES1n register

(4) The TMC1n0.TM1CAEn bit is a reset signal of 16-bit timer/event counter 1n. To use 16-bit timer/level counter 1n, first set (1) the TM1CAEn bit.

(5) The analog noise elimination time + two cycles of the input clock are required to detect the valid edge of the external trigger signal (INTP1n0 or INTP1n1) or the external clock input (TI1n). Therefore, edge detection will not be performed normally for changes that are less than the analog noise elimination time + two cycles of the input clock. Only two f_{XX} clocks are necessary for detecting the valid edge of the external clear input (TCLR1n).

(6) The operation of an interrupt request signal (INTCC1n0 or INTCC1n1) is automatically determined according to the operating state of the capture/compare register. When the capture/compare register is used for a capture operation, the external trigger signal is used for a valid edge detection interrupt. When the capture/compare register is used for a compare operation, the external interrupt request signal is used for an interrupt indicating a match with the TM1n register.

(7) If the TMC1n1.ENTO1n and TMC1n1.ALV1n bits are changed at the same time, a glitch (spike shaped noise) may be generated in the TO1n pin output. Either create a circuit configuration that will not malfunction even if a glitch is generated or make sure that the ENTO1n and ALV1n bits are not changed at the same time.

Remark n = 0, 1

9.1 Function Overview

8-bit timer/event counter 2n has the following two modes (n = 0, 1).

- Mode using 8-bit timer/event counter alone (individual mode)
- Mode using cascade connection (16-bit resolution: cascade connection mode)

These two modes are described below.

(1) Mode using 8-bit timer/event counter alone (individual mode)

8-bit timer/event counter 2n operates as an 8-bit timer/event counter.
The following functions can be used.

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

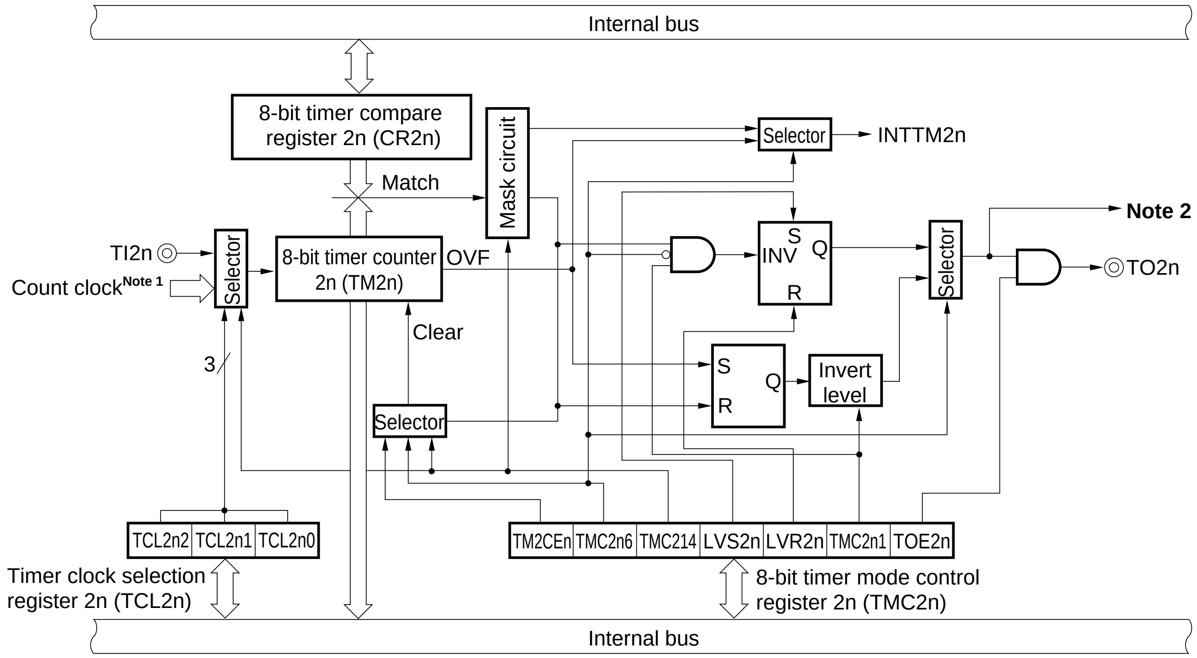
(2) Mode using cascade connection (16-bit resolution: cascade connection mode)

TM20 and TM21 can be used as 16-bit timer/event counters when they are connected in cascade. The following functions can be used.

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

The block diagram of 8-bit timer/event counter 2n is shown next.

Figure 9-1. Block Diagram of 8-Bit Timer/Event Counter 2n



Notes 1. The count clock is set by the TCL2n register.

2. Serial interface clock

Remark n = 0, 1

9.2 Configuration

8-bit timer/event counter 2n consists of the following hardware (n = 0, 1).

Table 9-1. Configuration of 8-Bit Timer/Event Counter 2n

Item	Configuration
Timer registers	8-bit timer counters 20, 21 (TM20, TM21) 16-bit timer counter 2 (TM2): Only when using cascade connection
Registers	8-bit timer compare registers 20, 21 (CR20, CR21) 16-bit timer compare register 2 (CR2): Only when using cascade connection
Timer output	TO20, TO21
Control registersNote	Timer clock selection registers 20, 21 (TCL20, TCL21) Timer clock selection register 2 (TCL2): Only when using cascade connection 8-bit timer mode control registers 20, 21 (TMC20, TMC21) 16-bit timer mode control register 2 (TMC2): Only when using cascade connection

Note When using the functions of the TI2n and TO2n pins, refer to **Table 4-15 Settings When Port Pins Are Used for Alternate Functions**.

(1) 8-bit timer counters 20, 21 (TM20, TM21)

The TM2n register is an 8-bit read-only register that counts the count pulses.

The counter is incremented in synchronization with the rising edge of the count clock.

The TM20 and TM21 registers can be used as 16-bit timers when they are connected in cascade. When these timers are used as 16-bit timers, their values can be read by using a 16-bit memory manipulation instruction.

After reset: 00H R Address: TM20 FFFFF640H, TM21 FFFFF641H

	7	6	5	4	3	2	1	0
TM2n (n = 0, 1)								

In the following cases, the count value becomes 00H.

- Reset
- When the TMC2n.TM2CEn bit is cleared (0)
- TM2n register and CR2n register match in the mode in which clear & start occurs on a match between the TM2n register and CR2n register

Caution When connected in cascade, these registers become 0000H even when the TM2CE0 bit in the lowest timer (TM20) is cleared.

Remark n = 0, 1

(2) 8-bit timer compare registers 20, 21 (CR20, CR21)

The CR2n register can be read and written by an 8-bit memory manipulation instruction.

In a mode other than the PWM mode, the value set to the CR2n register is always compared to the count value of the TM2n register, and if the two values match, an interrupt request signal (INTTM2n) is generated.

In the PWM mode, TM2n register overflow causes the TO2n pin output to change to the active level, and when the values of the TM2n register and the CR2n register match, the TO2n pin output changes to the inactive level.

The value of the CR2n register can be set in the range of 00H to FFH.

When the TM20 and TM21 registers are connected in cascade as 16-bit timers, the CR20 register and CR21 register function as 16-bit timer compare register 2 (CR2). The count value and register value are compared in 16-bit lengths, and if they match, an interrupt request (INTTM20) is generated.

These registers are set to 00H after reset.

After reset: 00H R/W Address: CR20 FFFFF642H, CR21 FFFFF643H

	7	6	5	4	3	2	1	0
CR2n (n = 0, 1)								

Cautions 1. In the mode in which clear & start occurs upon a match of the TM2n register and CR2n register (TMC2n.TMC2n6 bit = 0), do not write a different value to the CR2n register during the count operation.

2. In the PWM mode, set the CR2n register rewrite interval to three or more count clocks (clock selected with the TCL2n register.

3. Before changing the value of the CR2n register when using a cascade connection, be sure to stop the timer operation.

Remark n = 0, 1

9.3 Registers

The following two registers are used to control 8-bit timer/event counter 2n.

- Timer clock selection register 2n (TCL2n)
- 8-bit timer mode control register 2n (TMC2n)

Remark To use the functions of the TI2n and TO2n pins, refer to **Table 4-15 Settings When Port Pins Are Used for Alternate Functions**.

(1) Timer clock selection registers 20, 21 (TCL20, TCL21)

TCL20 and TCL21 set the count clock of 8-bit timer/event counter 2n and the valid edge of the TI2n pin input.
These registers are set by an 8-bit memory manipulation instruction.
These registers are set to 00H after reset.

After reset: 00H R/W Address: TCL20 FFFFF644H, TCL21 FFFFF645H

	7	6	5	4	3	2	1	0
TCL2n	0	0	0	0	0	TCL2n2	TCL2n1	TCL2n0

(n = 0, 1)

TCL2n2	TCL2n1	TCL2n0	Count clock selection		
			Clock	f_{XX}	
				20 MHz	10 MHz
0	0	0	Falling edge of TI2n	–	–
0	0	1	Rising edge of TI2n	–	–
0	1	0	$f_{XX}/4$	200 ns	400 ns
0	1	1	$f_{XX}/8$	400 ns	800 ns
1	0	0	$f_{XX}/16$	800 ns	1.60 μs
1	0	1	$f_{XX}/32$	1.60 μs	3.20 μs
1	1	0	$f_{XX}/128$	6.40 μs	12.8 μs
1	1	1	$f_{XX}/512$	25.6 μs	51.2 μs

Cautions 1. Before overwriting the TCL2n register with different data, stop the timer operation.

2. Because the TI2n pin functions alternately as P03/INTP2 and P14/TO21, select the timer input function by setting the PMC0, PFC0, PMC1, and PFC1 registers before starting the timer operation when using the TI2n pin function. If the TI2n pin is manipulated after the timer operation, the edge detection operation is not performed correctly.

Remark When TCL2n is connected in cascade, the TCL1 register settings are invalid.

(2) 8-bit timer mode control registers 20, 21 (TMC20, TMC21)

The TMC2n register performs the following six settings.

- Controls counting by 8-bit timer counters 20, 21 (TM20, TM21)
- Selects the operation mode of the TM2n register
- Selects the individual mode or cascade connection mode
- Sets the status of the timer output flip-flop
- Controls the timer output flip-flop or selects the active level in the PWM (free-running) mode
- Controls timer output

The TMC2n register is set by an 8-bit or 1-bit memory manipulation instruction.
These registers are set to 00H after reset.

Remark n = 0, 1

After reset: 00H R/W Address: TMC20 FFFFF646H, TMC21 FFFFF647H

	<7>	6	5	4	<3>	<2>	1	<0>
TMC2n (n = 0, 1)	TM2CEn	TMC2n6	0	TMC214[Note]	LVS2n	LVR2n	TMC2n1	TOE2n

TM2CEn	Control of count operation of 8-bit timer/event counter 2n
0	Counting is disabled after the counter is cleared to 0 (counter disabled)
1	Start count operation

TMC2n6	Selection of operation mode of 8-bit timer/event counter 2n
0	Mode in which clear & start occurs on match between TM2n register and CR2n register
1	PWM (free-running) mode

TMC214	Selection of individual mode or cascade connection mode
0	Individual mode
1	Cascade connection mode (connected with TM20)

LVS2n	LVR2n	Setting of status of timer output F/F
0	0	Unchanged
0	1	Reset timer output F/F to 0
1	0	Set timer output F/F to 1
1	1	Setting prohibited

TMC2n1	Other than PWM (free-running) mode (TMC2n6 = 0)	PWM (free-running) mode (TMC2n6 = 1)
	Controls timer F/F	Selects active level
0	Disable inversion operation	High active
1	Enable inversion operation	Low active

TOE2n	Timer output control
0	Disable output (TO2n pin is low level)
1	Enable output

Note Bit 4 of the TMC20 register is fixed to 0.

Cautions 1. The LVS2n and LVR2n bit settings are valid in other than the PWM mode.

2. Do not rewrite the following bits at the same time.

- **TMC2n1 bit and TOE2n bit**
- **TMC2n6 bit and TOE2n bit**
- **TMC2n1 bit and TMC2n6 bit**
- **TMC2n6 bit and LVS2n bit or LVR2n bit**
- **TOE2n bit and LVS2n bit or LVR2n bit**

Remarks 1. In the PWM mode, the PWM output is set to the inactive level by setting TM2CEn bit = 0.

2. When the LVS2n and LVR2n bits are read, 0 is read.

3. The values of the TMC2n6, LVS2n, LVR2n, TMC2n1, and TOE2n bits are reflected to the TO2n output regardless of the TM2CEn bit value.

9.4 Operation

9.4.1 Operation as interval timer (8 bits)

8-bit timer/event counter 2n operates as an interval timer that repeatedly generates interrupts at the interval of the count value preset in the CR2n register.

If the count value in the TM2n register matches the value set in the CR2n register, the value of the TM2n register is cleared to 00H and counting is continued, and at the same time, an interrupt request signal (INTTM2n) is generated.

Setting method

<1> Set each register.

- TCL2n register: Selects the count clock (t).
- CR2n register: Compare value (N)
- TMC2n register: Stops count operation and selects the mode in which clear & start occurs on a match between the TM2n register and CR2n register (TMC2n register = 0000xx11B, x: don't care).

<2> When the TMC2n.TM2CEn bit is set to 1, the count operation starts.

<3> When the values of the TM2n register and CR2n register match, the INTTM2n signal is generated (TM2n register is cleared to 00H).

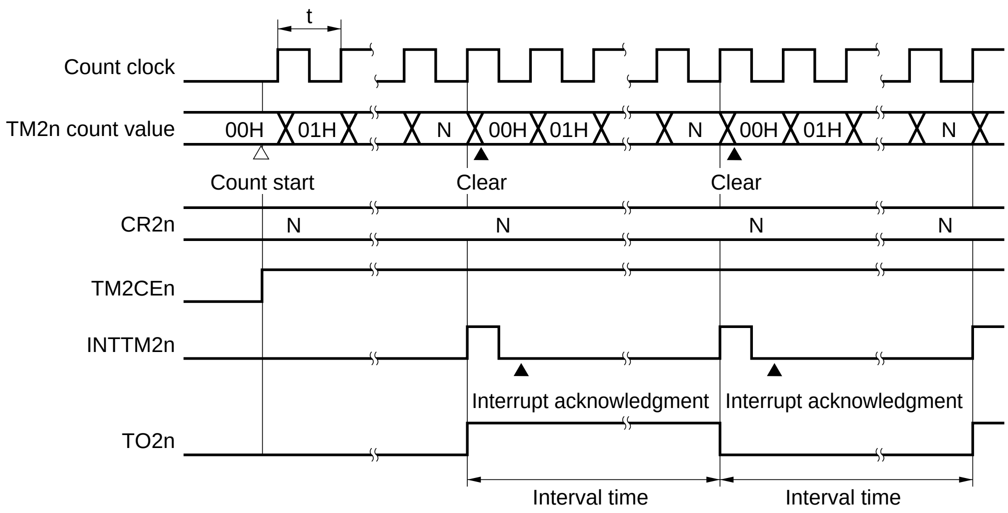
<4> Then, the INTTM2n signal is repeatedly generated at the same interval. To stop counting, set the TM2CEn bit to 0.

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

Caution During interval timer operation, do not rewrite the value of the CR2n register.

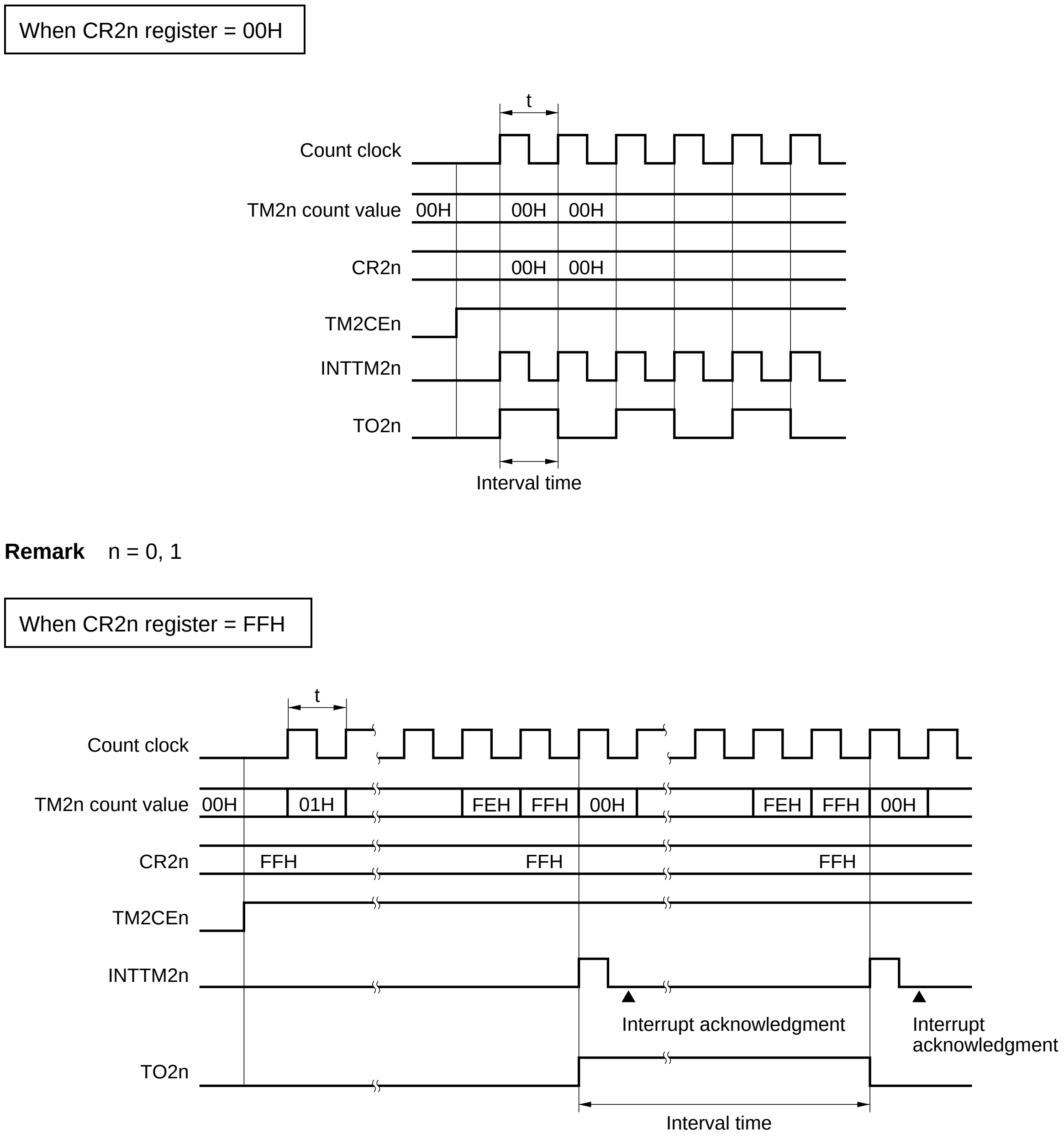
Figure 9-2. Timing of Interval Timer Operation (1/2)

Basic operation



Remark n = 0, 1

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



When CR2n register = 00H

Remark n = 0, 1

When CR2n register = FFH

Remark n = 0, 1

9.4.2 Operation as external event counter (8 bits)

The external event counter counts the number of clock pulses input to the TI2n pin from an external source by using the TM2n register.

Each time the valid edge specified by the TCL2n register is input to the TI2n pin, the TM2n register is incremented. Either the rising edge or the falling edge can be specified as the valid edge.

When the count value of the TM2n register matches the value of the CR2n register, the TM2n register is cleared to 00H and an interrupt request signal (INTTM2n) is generated.

Setting method

<1> Set each register.

- TCL2n register: Selects the TI2n input edge.
  Falling edge of TI2n pin → TCL2n register = 00H
  Rising edge of TI2n pin → TCL2n register = 01H
- CR2n register: Compare value (N)
- TMC2n register: Stops count operation, selects the mode in which clear & start occurs on a match between the TM2n register and CR2n register, disables timer output F/F inversion operation, and disables timer output.
  (TMC2n register = 0000xx00B, x: don’t care)
- For the alternate-function pin settings, refer to **Table 4-15 Settings When Port Pins Are Used for Alternate Functions**.

<2> When the TMC2n.TM2CEn bit is set to 1, the counter counts the number of pulses input from the TI2n pin.

<3> When the values of the TM2n register and CR2n register match, the INTTM2n signal is generated (TM2n register is cleared to 00H).

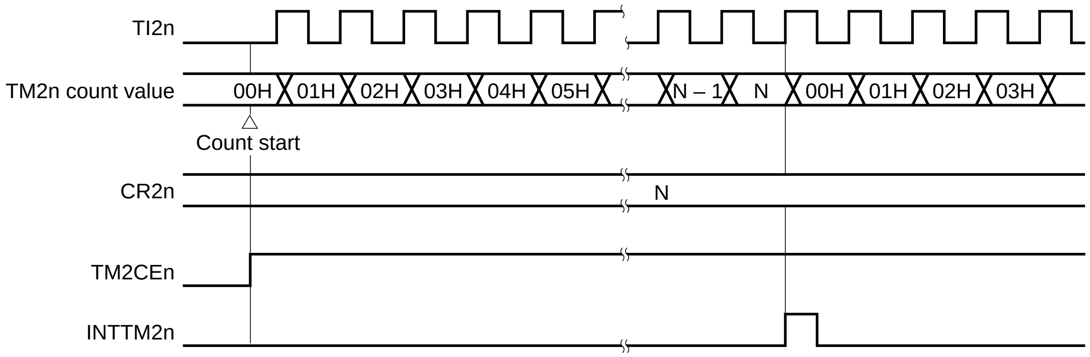
<4> Then, the INTTM2n signal is generated each time the values of the TM2n register and CR2n register match.

INTTM2n is generated when the valid edge of TI2n is input N + 1 times: N = 00H to FFH

Caution During external event counter operation, do not rewrite the value of the CR2n register.

Remark n = 0, 1

Figure 9-3. Timing of External Event Counter Operation (with Rising Edge Specified)



Remark n = 0, 1

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

A square wave with any frequency can be output at an interval determined by the value preset in the CR2n register.

By setting the TMC2n.TOE2n bit to 1, the output status of the TO2n pin is inverted at an interval determined by the count value preset in the CR2n register. In this way, a square wave of any frequency can be output (duty = 50%) (n = 0, 1).

Setting method

<1> Set each register.

- TCL2n register: Selects the count clock (t).
- CR2n register: Compare value (N)
- TMC2n register: Stops count operation, selects the mode in which clear & start occurs on a match between the TM2n register and CR2n register, makes the timer output initial setting, enables timer output F/F inversion operation, and enables timer output.
  (TMC2n register = 00001011B or 00000111B)
- For the alternate-function pin settings, refer to **Table 4-15 Settings When Port Pins Are Used for Alternate Functions**.

<2> When the TMC2n.TM2CEn bit is set to 1, counting starts.

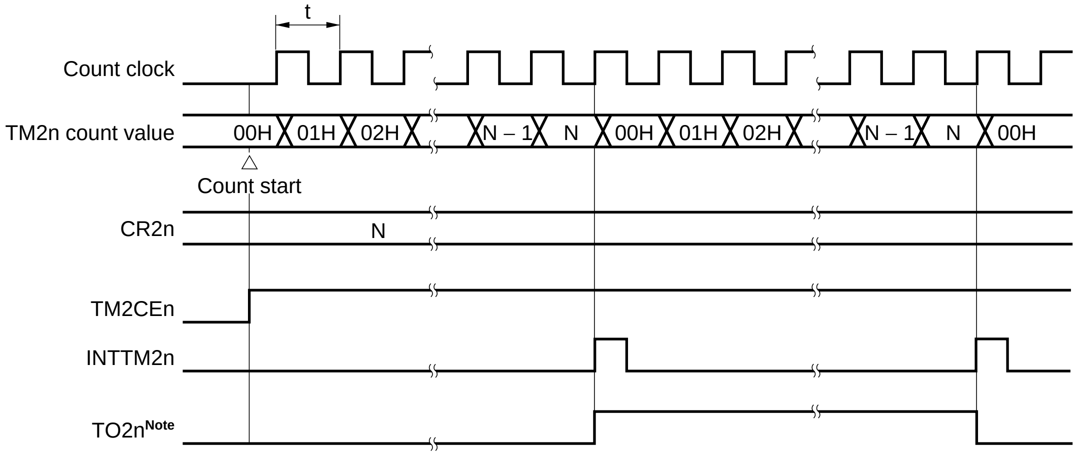
<3> When the values of the TM2n register and CR2n register match, the timer output F/F is inverted.
Moreover, the INTTM2n signal is generated and the TM2n register is cleared to 00H.

<4> Then, the timer F/F is inverted during the same interval and a square wave is output from the TO2n pin.

Frequency = 1/2t (N + 1): N = 00H to FFH

Caution Do not rewrite the value of the CR2n register during square-wave output.

Figure 9-4. Timing of Square-Wave Output Operation



Note The initial value of the TO2n output can be set using the TMC2n.LVS2n and TMC2n.LVR2n bits.

Remark n = 0, 1

9.4.4 8-bit PWM output operation

By setting the TMC2n.TMC2n6 bit to 1, 8-bit timer/event counter 2n performs PWM output.

Pulses with the duty factor determined by the value set in the CR2n register are output from the TO2n pin.

Set the width of the active level of the PWM pulse in the CR2n register. The active level can be selected using the TMC2n.TMC2n1 bit.

The count clock can be selected using the TCL2n register.

PWM output can be enabled/disabled by the TMC2n.TOE2n bit.

Caution The CR2n register rewrite interval must be three or more operation clocks (set by the TCL2n register).

Usage method

<1> Set each register.

- TCL2n register: Selects the count clock (t).
- CR2n register: Compare value (N)
- TMC2n register: Stops count operation, selects PWM mode, leaves timer output F/F unchanged, sets active level, and enables timer output. (TMC2n register = 01000001B or 01000011B)
- For the alternate-function pin settings, refer to **Table 4-15 Settings When Port Pins Are Used for Alternate Functions**.

<2> When the TMC2n.TM2CEn bit is set to 1, counting starts.

PWM output operation

<1> When counting starts, PWM output (output from the TO2n pin) outputs the inactive level until an overflow occurs.

<2> When an overflow occurs, the active level set by setting method <1> is output. The active level is output until the value of the CR2n register and the count value of the TM2n register match.

<3> When the value of the CR2n register and the count value match, the inactive level is output and continues to be output until an overflow occurs again.

<4> Then, steps <2> and <3> are repeated until counting is stopped.

<5> When counting is stopped by setting the TM2CEn register to 0, PWM output becomes inactive.

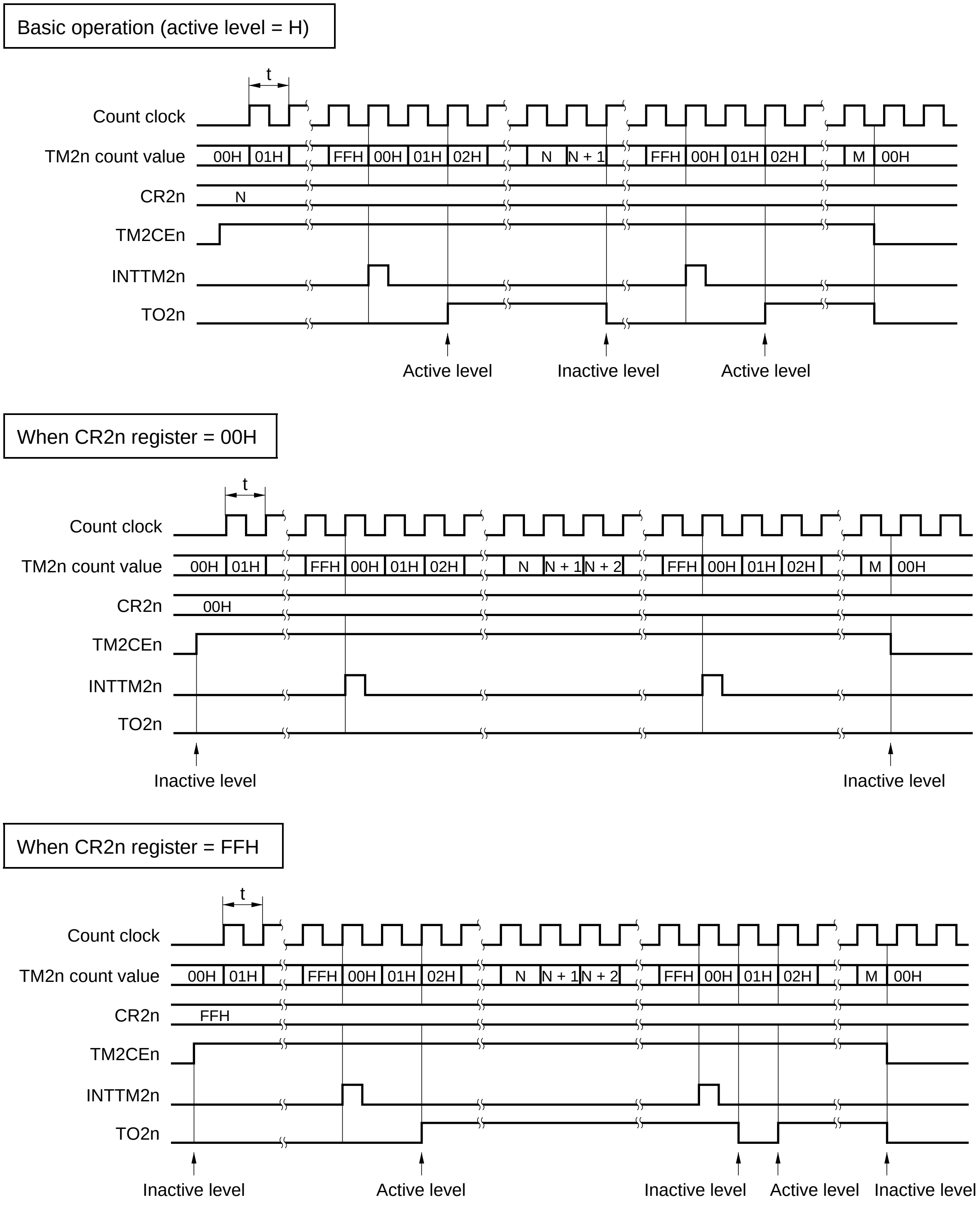
Cycle = 2^8t, active level width = Nt, duty = N/2^8: N = 00H to FFH

Remarks 1. n = 0, 1

2. For the detailed timing, refer to **Figure 9-5 Timing of PWM Output Operation** and **Figure 9-6 Timing of Operation Based on CR2n Register Transitions**.

(a) Basic operation of PWM output

Figure 9-5. Timing of PWM Output Operation



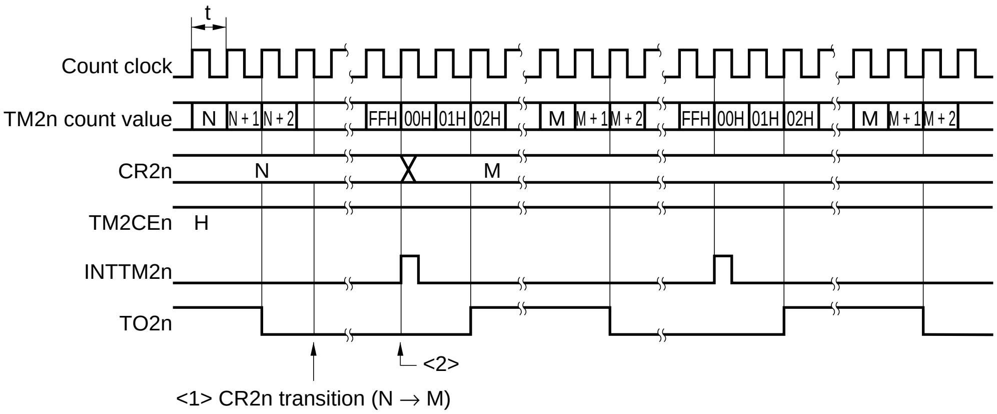
Remark n = 0, 1

(b) Operation based on CR2n register transitions

Figure 9-6. Timing of Operation Based on CR2n Register Transitions

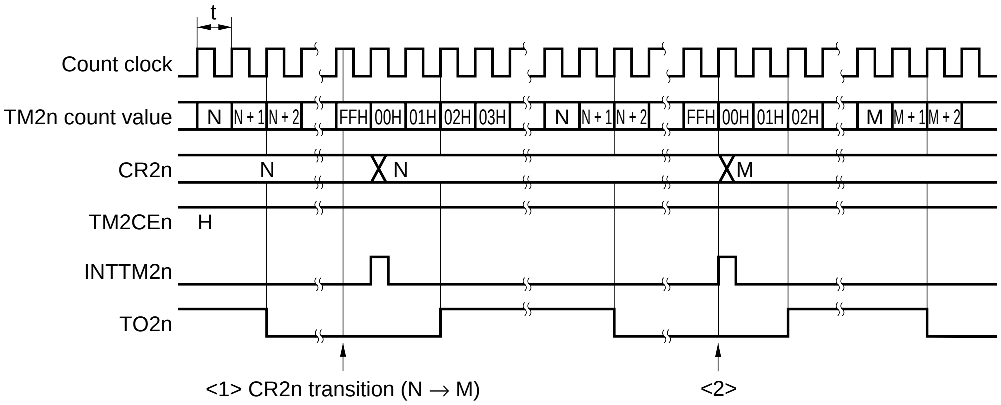
When the value of the CR2n register changes from N to M before the rising edge of the FFH clock

→ The value is transferred to the CR2n register at the overflow that occurs immediately after.



When the value of the CR2n register changes from N to M after the rising edge of the FFH clock

→ The value is transferred to the CR2n register at the second overflow.



Caution In the case of read from the CR2n register between <1> and <2>, the value that is actually used differs (read value: M; actual value of CR2n register: N).

Remark n = 0, 1

9.4.5 Operation as interval timer (16 bits)

The 16-bit resolution timer/event counter mode is selected by setting the TMC21.TMC214 bit to 1.

8-bit timer/event counter 2n operates as an interval timer by repeatedly generating interrupts using the count value preset in the CR2 register as the interval.

Setting method

<1> Set each register.

- TCL20 register: Selects the count clock (t)
  (The TCL21 register does not need to be set in cascade connection)
- CR20 register: Compare value (N) ... Lower 8 bits (settable from 00H to FFH)
- CR21 register: Compare value (N) ... Higher 8 bits (settable from 00H to FFH)
- TMC20, TMC21 registers: Selects the mode in which clear & start occurs on a match between TM2 register and CR2 register (x: don't care)
  (TMC20 register = 0000xx11B
  TMC21 register = 0001xx00B)

<2> Set the TMC21.TM2CE1 bit to 1. Then set the TMC20.TM2CE0 bit to 1 to start the count operation.

<3> When the values of the TM2 register and CR2 register connected in cascade match, the INTTM20 signal is generated (the TM2 register is cleared to 0000H).

<4> The INTTM20 signal is then generated repeatedly at the same interval.

Interval time = (N + 1) × t: N = 0000H to FFFFH

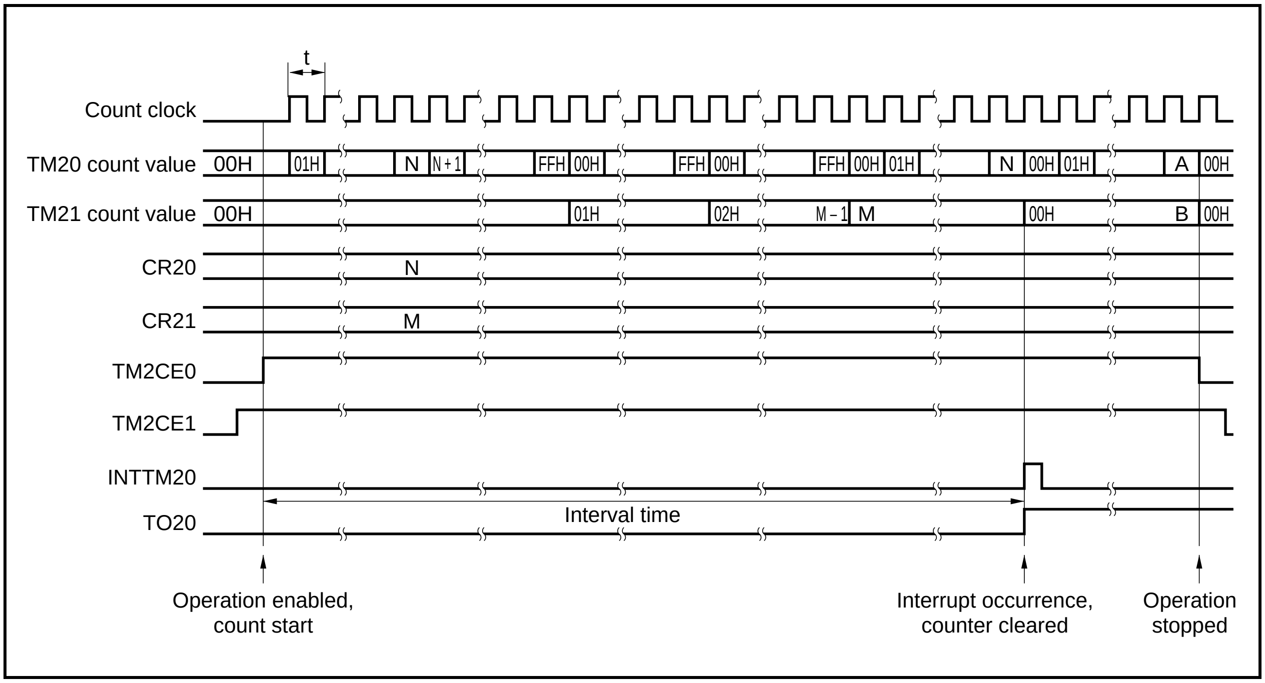
Cautions 1. To write using 8-bit access during cascade connection, set the TM2CE1 bit to 1 at operation start and then set the TM2CE0 bit to 1. When operation is stopped, set the TM2CE0 bit to 0 and then set the TM2CE1 bit to 0.

2. During cascade connection, use TI20 input, TO20 output, and INTTM20 and do not use and mask TI21 input, TO21 output, and INTTM21 (for details, refer to CHAPTER 16 INTERRUPT/EXCEPTION PROCESSING FUNCTION). Set bits LVS21, LVR21, TMC211, and TOE21 to 0.

3. Do not change the value of the CR2 register during timer operation.

Figure 9-7 shows a timing example of the cascade connection mode with 16-bit resolution.

Figure 9-7. Cascade Connection Mode with 16-Bit Resolution



9.4.6 Operation as external event counter (16 bits)

The 16-bit resolution timer/event counter mode is selected by setting the TMC21.TMC214 bit to 1.

The external event counter counts the number of clock pulses input to the TI20 pin from an external source using the TM2 register.

Setting method

<1> Set each register.

- TCL20 register: Selects the TI20 input edge.
  (The TCL21 register does not have to be set during cascade connection.)
  Falling edge of TI20 → TCL20 register = 00H
  Rising edge of TI20 → TCL20 register = 01H
- CR20 register: Compare value (N) ... Lower 8 bits (settable from 00H to FFH)
- CR21 register: Compare value (N) ... Higher 8 bits (settable from 00H to FFH)
- TMC20, TMC21 registers: Stops count operation, selects the clear & start mode entered on a match between the TM2 register and CR2 register, disables timer output F/F inversion, and disables timer output.
  (x: don’t care)
  TMC20 register = 0000xx00B
  TMC21 register = 0001xx00B
- For the alternate-function pin settings, refer to **Table 4-15 Settings When Port Pins Are Used for Alternate Functions**.

<2> Set the TMC21.TM2CE1 bit to 1. Then set the TMC20.TM2CE0 bit to 1 and count the number of pulses input from TI20.

<3> When the values of the TM2 register and CR2 register connected in cascade match, the INTTM20 signal is generated (the TM2 register is cleared to 0000H).

<4> INTTM20 is then generated each time the values of the TM2 register and CR2 register match.

INTTM20 is generated when the valid edge of TI20 is input N + 1 times: N = 0000H to FFFFH

Cautions 1. During external event counter operation, do not rewrite the value of the CR2n register.

2. To write using 8-bit access during cascade connection, set the TM2CE1 bit to 1 and then set the TM2CE0 bit to 1. When operation is stopped, set the TM2CE0 bit to 0 and then set the TM2CE1 bit to 0.

3. During cascade connection, use TI20 input and INTTM20 and do not use and mask TI21 input, TO21 output, and INTTM21 (for details, refer to CHAPTER 16 INTERRUPT/EXCEPTION PROCESSING FUNCTION). Set bits LVS21, LVR21, TMC211, and TOE21 to 0.

4. Do not change the value of the CR2 register during external event counter operation.

9.4.7 Square-wave output operation (16-bit resolution)

The 16-bit resolution timer/event counter mode is selected by setting the TMC21.TMC214 bit to 1.

8-bit timer/event counter 2n outputs a square wave of any frequency using the interval preset in the CR2 register.

Setting method

<1> Set each register.

- TCL20 register: Selects the count clock (t)
  (The TCL21 register does not have to be set in cascade connection)
- CR20 register: Compare value (N) ... Lower 8 bits (settable from 00H to FFH)
- CR21 register: Compare value (N) ... Higher 8 bits (settable from 00H to FFH)
- TMC20, TMC21 registers: Stops count operation, selects the mode in which clear & start occurs on a match between the TM2 register and CR2 register.

LVS20	LVR20	Timer Output F/F Status Settings
1	0	High-level output
0	1	Low-level output

Enables timer output F/F inversion, and enables timer output.

(TMC20 register = 00001011B or 00000111B
TMC21 register = 00010000B)

- For the alternate-function pin settings, refer to **Table 4-15 Settings When Port Pins Are Used for Alternate Functions**.

<2> Set the TMC21.TM2CE1 bit to 1. Then set the TMC20.TM2CE0 bit to 1 to start the count operation.

<3> When the values of the TM2 register and the CR2 register connected in cascade match, the TO20 timer output F/F is inverted. Moreover, the INTTM20 signal is generated and the TM2 register is cleared to 0000H.

<4> Then, the timer F/F is inverted during the same interval and a square wave is output from the TO20 pin.

Frequency = 1/2t (N + 1): N = 0000H to FFFFH

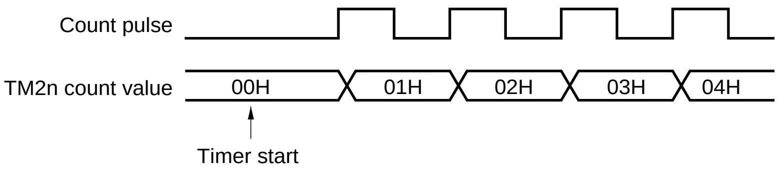
Caution Do not write a different value to the CR2 register.

9.5 Cautions

(1) Error on starting timer

An error of up to 1 clock occurs before the match signal is generated after the timer has been started. This is because the TM2n register is started asynchronously to the count pulse.

Figure 9-8. Start Timing of Timer 2n



Remark n = 0, 1

CHAPTER 10 REAL-TIME COUNTER FUNCTION

10.1 Functions

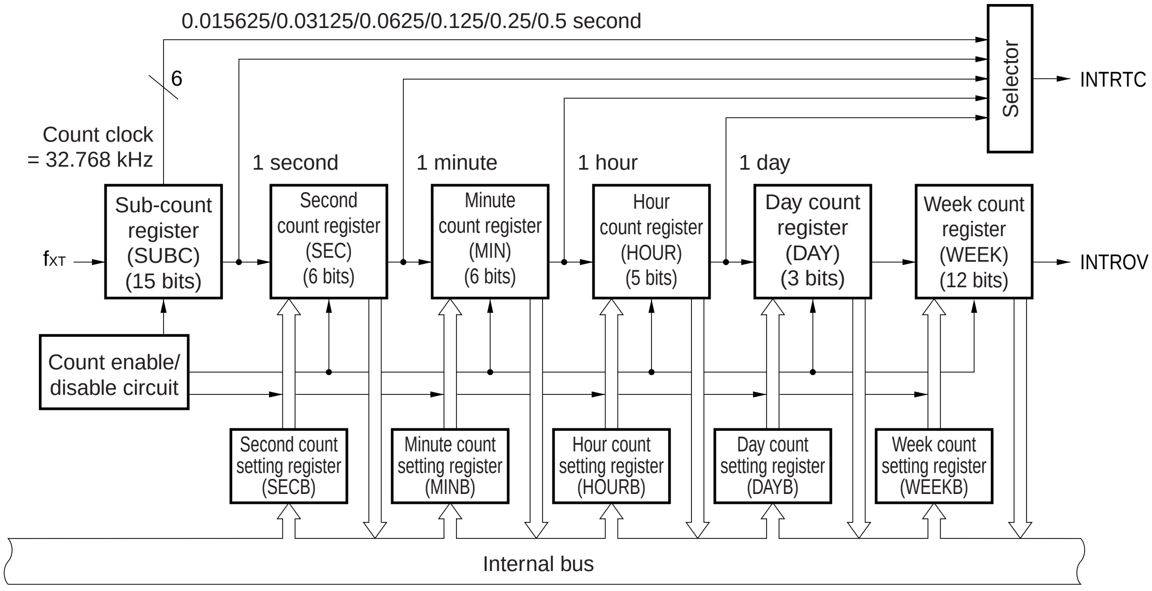
The real-time counter has the following functions.

- Week, day, hour, minute, and second counters that can count up to 4,095 weeks
- Week, day, hour, minute, and second counters can be read while they are operating/stopped
- Generates overflow interrupt request signal (INTROV) from week counter.
- Generates interval interrupt request signal (INTRTC) at intervals of 0.015625, 0.03125, 0.0625, 0.125, 0.25, 0.5, or 1 second, 1 minute, 1 hour, or 1 day.

10.2 Configuration

The block diagram of the real-time counter is shown below.

Figure 10-1. Block Diagram of Real-Time Counter


0.015625/0.03125/0.0625/0.125/0.25/0.5 second
6
Selector
INTRTC
Count clock
= 32.768 kHz
1 second
1 minute
1 hour
1 day
fXT
Sub-count register (SUBC) (15 bits)
Second count register (SEC) (6 bits)
Minute count register (MIN) (6 bits)
Hour count register (HOUR) (5 bits)
Day count register (DAY) (3 bits)
Week count register (WEEK) (12 bits)
INTROV
Count enable/ disable circuit
Second count setting register (SECB)
Minute count setting register (MINB)
Hour count setting register (HOURB)
Day count setting register (DAYB)
Week count setting register (WEEKB)
Internal bus

Remark f_{XT}: Subclock frequency

Table 10-1. Configuration of Real-Time Counter

Item	Configuration
Registers	RTC control register 0 (RTCC0) RTC control register 1 (RTCC1) Sub-count register (SUBC) Second count register (SEC) Second count setting register (SECB) Minute count register (MIN) Minute count setting register (MINB) Hour count register (HOUR) Hour count setting register (HOURB) Day count register (DAY) Day count setting register (DAYB) Week count register (WEEK) Week count setting register (WEEKB)

10.3 Registers

The registers listed in the table below control the real-time counter.

(1) RTC control register 0 (RTCC0)

RTCC0 is an 8-bit register that controls the operation of the real-time counter. This register can be read or written in 8-bit or 1-bit units.

This register is set to 80H after reset.

After reset: 80H R/W Address: FFFFF680H

	<7>	6	5	4	3	2	1	0
RTCC0	RTCAE	0	0	0	0	0	0	0

RTCAE	Enables/disables RTC operation
0	Stops RTC clock operation and resets sub-count value.
1	Enables RTC clock operation.

(2) RTC control register 1 (RTCC1)

RTCC1 is an 8-bit register that controls the operation of the real-time counter. This register can be read or written in 8-bit or 1-bit units.

This register is set to 8xH after reset.

After reset: 8xH[Note 1] R/W Address: FFFFF681H

	<7>	6	5	4	3	2	1	<0>
RTCC1	RTCE	INTS3	INTS2	INTS1	INTS0	0	0	RTCF[Note 2]

RTCE	Enables/disables RTC count-up operation
0	Disables RTC count operation.
1	Enables RTC count operation.

INTS3	INTS2	INTS1	INTS0	Specifies interrupt request signal generation timing
0	0	0	0	Does not generate interrupt request signal.
0	0	0	1	Generates interrupt request signal every 0.015625 second.
0	0	1	0	Generates interrupt request signal every 0.03125 second.
0	0	1	1	Generates interrupt request signal every 0.0625 second.
0	1	0	0	Generates interrupt request signal every 0.125 second.
0	1	0	1	Generates interrupt request signal every 0.25 second.
0	1	1	0	Generates interrupt request signal every 0.5 second.
0	1	1	1	Generates interrupt request signal every 1 second.
1	0	0	0	Generates interrupt request signal every 1 minute.
1	0	0	1	Generates interrupt request signal every 1 hour.
1	0	1	0	Generates interrupt request signal every 1 day.
Other than above				Setting prohibited

RTCF	RTC operation flag
0	Count operation is stopped
1	Count-up operation is in progress.

Notes 1. 80H or 81H, depending on the value of the RTCF bit.

2. The RTCF bit is a read-only bit.

(3) Sub-count register (SUBC)

SUBC is a 15-bit register that counts the reference time of the real-time counter. It counts 1 second using the 32.768 kHz clock. This register is read-only, in 16-bit or 8-bit units.

This register is not initialized after reset or when RTCC1.RTCE bit = 0.

After reset: Undefined R Address: FFFFF682H

	15	14	0
SUBC	0	SUBC14 to SUBC0	

(4) Second count register (SEC)

SEC is an 8-bit register that uses a value of 0 to 59 (decimal) to indicate the count value in seconds. This register is read-only, in 8-bit units.

This register is not initialized after reset or when RTCC1.RTCE bit = 0.

After reset: Undefined R Address: FFFFF684H

	7	6	5	4	3	2	1	0
SEC	0	0	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0

(5) Second count setting register (SECB)

SECB is an 8-bit register for setting the second count. This register is write-only, in 8-bit units. Set a count value in a range of 0 to 59 (decimal) to this register. Do not set a count value of 60 (decimal) or greater.

This register is set to 00H after reset.

After reset: 00H W Address: FFFFF68AH

	7	6	5	4	3	2	1	0
SECB	0	0	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0

(6) Minute count register (MIN)

MIN is an 8-bit register that uses a value of 0 to 59 (decimal) to indicate the count value in minutes. This register is read-only, in 8-bit units.

This register is not initialized after reset or when RTCC1.RTCE bit = 0.

After reset: Undefined R Address: FFFFF685H

	7	6	5	4	3	2	1	0
MIN	0	0	MIN5	MIN4	MIN3	MIN2	MIN1	MIN0

(7) Minute count setting register (MINB)

MINB is an 8-bit register for setting the minute count. This register is write-only, in 8-bit units. Set a count value in a range of 0 to 59 (decimal) to this register. Do not set a count value of 60 (decimal) or greater.

This register is set to 00H after reset.

After reset: 00H W Address: FFFFF68BH

	7	6	5	4	3	2	1	0
MINB	0	0	MIN5	MIN4	MIN3	MIN2	MIN1	MIN0

(8) Hour count register (HOUR)

HOUR is an 8-bit register that uses a value of 0 to 23 (decimal) to indicate the count value in hours. This register is read-only, in 8-bit units.

This register is not initialized after reset or when RTCC1.RTCE bit = 0.

After reset: Undefined R Address: FFFFF686H

	7	6	5	4	3	2	1	0
HOUR	0	0	0	HOUR4	HOUR3	HOUR2	HOUR1	HOUR0

(9) Hour count setting register (HOURB)

HOURB is an 8-bit register for setting the hour count. This register is write-only, in 8-bit units. Set a count value in a range of 0 to 23 (decimal) to this register. Do not set a count value of 24 (decimal) or greater.

This register is set to 00H after reset.

After reset: 00H W Address: FFFFF68CH

	7	6	5	4	3	2	1	0
HOURB	0	0	0	HOUR4	HOUR3	HOUR2	HOUR1	HOUR0

(10) Day count register (DAY)

DAY is an 8-bit register that uses a value of 0 to 6 (decimal) to indicate the count value in days. This register is read-only, in 8-bit units.

This register is not initialized after reset or when RTCC1.RTCE bit = 0.

After reset: Undefined R Address: FFFFF687H

	7	6	5	4	3	2	1	0
DAY	0	0	0	0	0	DAY2	DAY1	DAY0

(11) Day count setting register (DAYB)

DAYB is an 8-bit register for setting the day count. This register is write-only, in 8-bit units. Set a count value in a range of 0 to 6 (decimal) to this register. Do not set a count value of 7 (decimal) or greater.

This register is set to 00H after reset.

After reset: 00H W Address: FFFFF68DH

	7	6	5	4	3	2	1	0
DAYB	0	0	0	0	0	DAY2	DAY1	DAY0

(12) Week count register (WEEK)

WEEK is a 16-bit register that uses a value of 0 to 4,095 (decimal) to indicate the count value in weeks. This register is read-only, in 8-bit or 16-bit units.

This register is not initialized after reset or when RTCC1.RTCE bit = 0.

After reset: Undefined R Address: FFFFF688H

	15 12	11 0
WEEK	0000	WEEK11 to WEEK0

(13) Week count setting register (WEEKB)

WEEKB is a 16-bit register for setting the week count. This register is write-only, in 8-bit or 16-bit units. Set a count value in a range of 0 to 4,095 (decimal) to this register.

This register is set to 0000H after reset.

After reset: 0000H W Address: FFFFF68DH

	15 12	11 0
WEEKB	0000	WEEK11 to WEEK0

10.4 Operation

10.4.1 Initializing counter and count-up

<1> After reset, the values of the RTCC0 and RTCC1 registers are initialized. Real-time counter clock operation is enabled when the RTCC0.RTCAE bit is set to 1, and real-time counter count operation is enabled when the RTCC1.RTCE of bit is set to 1.

<2> The sub-count register (SUBC) is reset if the real-time count clock operation is stopped when the RTCAE bit is 0.

<3> The real-time counter clock operation is started when the RTCAE bit is set to 1.

<4> After 3 internal clocks, the values of all the count setting registers are reflected on the corresponding count registers at all once, and each count register starts counting up.

<5> Each time a count register overflows, the higher count register starts counting up.

<6> At the clock after the one at which the overflow conditions of all the count registers have been satisfied, all the count registers are cleared to “0”. The INTROV signal is asserted active for the duration of one cycle of the real-time count clock after the WEEK register overflows.

10.4.2 Rewriting counter

<1> After reset, the values of the RTCC0 and RTCC1 registers are initialized. Real-time counter clock operation is enabled when the RTCC1.RTCAE bit is set to 1, and real-time counter count operation is enabled when the RTCC1.RTCE bit is set to 1.

<2> Write a value to each count setting register.

<3> The value of all the count setting registers are reflected on the corresponding count registers all at once two internal clocks after the RTCE bit is set to 1, and the real-time counter starts counting up 3 internal clocks after that.

10.4.3 Controlling interrupt request signal output

This section explains how to control interrupt request signals, taking the RTCC1.INTS0 to RTCC1.INTS3 bits = 0111B (every second) and the RTCC1.INTS0 to RTCC1.INTS3 bits = 1000B (every minute) as an example.

<1> After reset, the values of the RTCC0 and RTCC1 registers are initialized. Real-time counter clock operation is enabled when the RTCC1.RTCAE bit is set to 1, and real-time counter count operation is enabled when RTCE is set to 1. The SUBC register is reset.

<2> Clear the RTCAE bit to 0.

<3> The internal clock operation is started when the RTCAE bit = 1.

<4> After 3 internal clocks, the value of all the count setting registers are reflected on the corresponding count registers at all once, and the real-time counter starts counting up.

<5> Set the INTS0 to INTS3 bits to 0111B (1000B).

<6> Because the INTS0 to INTS3 bits = 0111B, the INTRTC signal is asserted each time 1 second is counted (because the INTS0 to INTS3 bits = 1000B, the INTRTC signal is asserted each time 1 minute is counted).

<7> The INTROV signal is asserted when the overflow conditions of all the count registers have been satisfied.

10.4.4 Notes

(1) If the real-time counter is not used, clear RTCC0.RTCAE to 0 after the reset signal has been cleared.

(2) Perform initialization after clearing the RTCAE bit to 0 when the reset signal has been cleared for the first time. For initialization, set each count setting register, count clock, and interrupt request signal generation timing using the procedure described in (4) and (5) below, and clear the ROVIC.ROVIF bit and the RTCIC.RTCIF bit to 0.

(3) Read each count register using the following procedure:

<1> Read the second, minute, hour, day, and week count registers in that order, and then read the second count register again.

<2> Compare the value of the second count register read first with the value of the second count register read last.
If the two values do not match, the chances are that the counter counted up while it was being read. If so, repeat steps <1> and <2> again.

(4) Write data to each count setting register using the following procedure:

- To clear the SUBC register
  <1> Using the procedure described in (3) above, read the values of all the count registers (this may be omitted), and clear the RTCAE bit to 0.
  <2> Write a value to one of the count setting registers. Write the value read in step <1> to the other count setting registers.
  <3> Set the RTCAE bit to 1. The values of the count setting registers will be transferred to the count registers, and the real-time counter will start counting (after 2 or 3 count clocks).
- To not clear the SUBC register (to hold the value)
  <1> Clear the RTCC1.RTCE bit to 0, and check if the RTCC1.RTCF bit is cleared to 0 (count stops).
  <2> Read the values of all the count registers (this may be omitted).
  <3> Write a value to one of the count setting registers. Write the value read in <2> to the other count setting registers.
  <4> Set the RTCE bit to 1. The values of the count setting registers will be transferred to the count registers, and the real-time counter will start counting (after 2 or 3 count clocks).

(5) To change the interrupt request signal generation timing, be sure to set the RTCIC.RTCMK bit to 1. After changing the timing, clear the RTCIC.RTCIF bit to 0.

(6) To change the count clock, be sure to clear the RTCAE bit to 0.

CHAPTER 11 WATCHDOG TIMER FUNCTIONS

11.1 Functions

The watchdog timer has the following operation modes.

- Watchdog timer
- Interval timer

The following functions are realized from the above-listed operation modes.

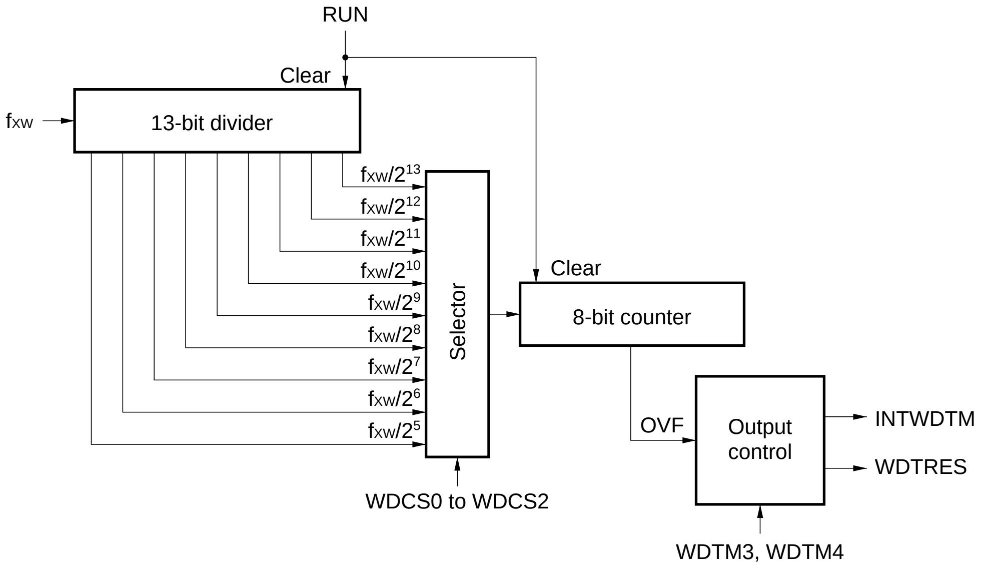
- Generation of system reset signal (WDTRES) upon overflow of watchdog timer
- Generation of maskable interrupt request signal (INTWDTM) upon overflow of interval timer

Remark Select whether to use the watchdog timer in the watchdog timer mode or the interval timer mode using the WDTM register.

11.2 Configuration

The watchdog timer consists of the following hardware.

Figure 11-1. Block Diagram of Watchdog Timer



Remark INTWDTM: Request signal for maskable interrupt through WDT overflow

WDTRES: Reset signal through WDT overflow

$f_{XW} = f_{XX}/16$: Watchdog timer clock frequency

Table 11-1. Configuration of Watchdog Timer

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

11.3 Registers

The registers that control the watchdog timer are as follows.

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

(1) Watchdog timer clock select register (WDCS)

WDCS is a register that sets the overflow time of the watchdog timer and the interval timer.

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

WDCS is set to 00H after reset.

After reset: 00H R/W Address: FFFFF6C1H

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

WDCS2	WDCS1	WDCS0	Overflow time of watchdog timer/interval timer		
				f_{XX}	
				20 MHz	10 MHz
0	0	0	$2^{17}/f_{XX}$	6.554 ms	13.11 ms
0	0	1	$2^{18}/f_{XX}$	13.11 ms	26.21 ms
0	1	0	$2^{19}/f_{XX}$	26.21 ms	52.43 ms
0	1	1	$2^{20}/f_{XX}$	52.43 ms	104.9 ms
1	0	0	$2^{21}/f_{XX}$	104.9 ms	209.7 ms
1	0	1	$2^{22}/f_{XX}$	209.7 ms	419.4 ms
1	1	0	$2^{23}/f_{XX}$	419.4 ms	838.9 ms
1	1	1	$2^{25}/f_{XX}$	1.678 s	3.355 s

Remark $f_{XW} = f_{XX}/16$: Watchdog timer clock frequency

(2) Watchdog timer mode register (WDTM)

WDTM is a register that sets the watchdog timer operation mode and enables/disables count operations.

This register is a special register that can be written only in a special sequence (refer to **3.4.7 Special registers**).

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

WDTM is set to 00H after reset.

★ **Caution When the main clock is stopped and the CPU is operating on the subclock, do not access the WDTM register using an access method that causes a wait. For details, refer to 3.4.8 (2).**

After reset: 00H R/W Address: FFFFF6C2H

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

RUN	Selection of operation mode of watchdog timer[Note 1]
0	Stops counting
1	Clears counter and starts counting

WDTM4	WDTM3	Selection of operation mode of watchdog timer[Note 2]
0	0	Interval timer mode (Upon overflow, maskable interrupt INTWDTM is generated.)
0	1	
1	0	Setting prohibited
1	1	Watchdog timer mode (Upon overflow, reset operation WDTRES is started.)

Notes 1. Once the RUN bit is set (to 1), it cannot be cleared (to 0) by software.
Therefore, when counting is started, it cannot be stopped except through $\overline{\text{RESET}}$ input.

2. Once the WDTM3 and WDTM4 bits are set (to 1), they cannot be cleared (to 0) by software and can be cleared only through $\overline{\text{RESET}}$ input.

Caution It takes up to 2 μs (at f_{XX} = 20 MHz) to write the WDTM register because of synchronization control with the WDT operation clock.

11.4 Operation

11.4.1 Operation as watchdog timer

Watchdog timer operation to detect a program loop is selected by setting the WDTM.WDTM4 and WDTM.WDTM3 bits to 11.

The count clock (program loop detection time interval) of the watchdog timer can be selected with the WDCS.WDCS0 to WDCS.WDCS2 bits. The count operation is started by setting the WDTM.RUN bit to 1. When, after the count operation is started, the RUN bit is again set to 1 within the set program loop detection time interval, the watchdog timer is cleared and the count operation starts again.

If the program loop detection time is exceeded without the RUN bit being set to 1, a reset signal (WDTRES) is generated.

The count operation of the watchdog timer stops in the software STOP mode and IDLE mode. Set the RUN bit to 1 before the software STOP mode or IDLE mode is entered in order to clear the watchdog timer.

Because the watchdog timer operates in the HALT mode, make sure that an overflow will not occur during HALT.

Cautions 1. Do not change the mode to the watchdog timer mode after clearing the WDTM4 bit to 0 (selecting the interval timer mode) and setting the RUN bit to 1.

2. When the subclock is selected for the CPU clock, the count operation of the watchdog timer stops (the value of the watchdog timer is maintained).

Table 11-2. Program Loop Detection Time of Watchdog Timer

Clock	Program Loop Detection Time	
	f_{XX} = 20 MHz	f_{XX} = 10 MHz
$2^{17}/f_{XX}$	6.554 ms	13.11 ms
$2^{18}/f_{XX}$	13.11 ms	26.21 ms
$2^{19}/f_{XX}$	26.21 ms	52.43 ms
$2^{20}/f_{XX}$	52.43 ms	104.9 ms
$2^{21}/f_{XX}$	104.9 ms	209.7 ms
$2^{22}/f_{XX}$	209.7 ms	419.4 ms
$2^{23}/f_{XX}$	419.4 ms	838.9 ms
$2^{25}/f_{XX}$	1.678 s	3.355 s

Remark $f_{XW} = f_{XX}/16$: Watchdog timer clock frequency

11.4.2 Operation as interval timer

The watchdog timer can be made to operate as an interval timer that repeatedly generates interrupts using the count value set in advance as the interval, by setting the WDTM.WDTM4 bit to 0.

When the watchdog timer operates as an interval timer, the WDTIC.WDTMK flag and priority specification flags (WDTIC.WDTPR0 to WDTIC.WDTPR2 bits) are valid and maskable interrupt request signals (INTWDTM) can be generated. The default priority of the INTWDTM signal is set to the highest level among the maskable interrupt request signals.

The interval timer continues to operate in the HALT mode, but it stops operating in the software STOP mode and the IDLE mode.

Cautions 1. Once the WDTM4 bit is set to 1 (thereby selecting the watchdog timer mode), the interval timer mode is not entered as long as $\overline{\text{RESET}}$ is not input.

2. When the subclock is selected for the CPU clock, the count operation of the watchdog timer stops (the value of the watchdog timer is maintained).

Table 11-3. Interval Time of Interval Timer

Clock	Interval Time	
	f_{XX} = 20 MHz	f_{XX} = 10 MHz
$2^{17}/f_{XX}$	6.554 ms	13.11 ms
$2^{18}/f_{XX}$	13.11 ms	26.21 ms
$2^{19}/f_{XX}$	26.21 ms	52.43 ms
$2^{20}/f_{XX}$	52.43 ms	104.9 ms
$2^{21}/f_{XX}$	104.9 ms	209.7 ms
$2^{22}/f_{XX}$	209.7 ms	419.4 ms
$2^{23}/f_{XX}$	419.4 ms	838.9 ms
$2^{25}/f_{XX}$	1.678 s	3.355 s

Remark $f_{XW} = f_{XX}/16$: Watchdog timer clock frequency

11.4.3 Monitoring reset by watchdog timer (WDT)

When the V850ES/PM1 has been reset, whether it has been reset by the watchdog timer (WDTRES) can be checked by using the WDRES register.

(1) WDT reset status register (WDRES)

WDRES is an 8-bit register that indicates the status of WDTRES and can be read or written by an 8-bit or 1-bit manipulation instruction.

To write the WDRES register, a specific sequence using the PRCMD register as a command register is required. If the register is written in an illegal sequence, writing is invalid and the protect error flag (bit 0 of SYS register: PRERR) is set to 1, and nothing is written to the register.

This register is undefined after reset.

After reset: Undefined R/W Address: FFFFF82AH

	7	6	5	4	3	2	1	<0>
WDRES	0	0	0	0	0	0	0	WRESF

WRESF	WDTRES detection flag
0	WDTRES did not occur
1	WDTRES occurred
Setting (1) condition: Reset by overflow of watchdog timer (WDT) Clearing (0) condition: Writing “0” by instruction or $\overline{\text{RESET}}$ pin input. Only “0” can be written to the WRESF bit.	

Caution Write “0” to the WRESF bit after confirming (reading) that the WRESF bit is 1 to avoid a conflict with setting the flag.

Remark The WRESF bit can be read or written, but it can only be cleared by writing “0”. “1” cannot be written to it.

CHAPTER 12 A/D CONVERTER

12.1 Functions

The A/D converter converts an analog input signal into a digital value. Its functions are as follows.

○ S/N ratio: 62 dB min. (when gain of ×16 is selected for channels 1, 3, and 5)
○ 16-bit resolution (conversion result register: 16 bits)
○ 6 channels
○ Analog input: 12 (positive, negative input/channel)
○ ΔΣ conversion mode
○ Pre-amplifier gain selectable: ×2 or ×16 (channels 1, 3, and 5)
○ Operating voltage: AV_{DD} = 3.0 to 3.6 V, AV_{SS} = 0 V
○ Analog input voltage: ±0.375 V (channels 0, 2, and 4)
±0.1875 V (channels 1, 3, and 5, when pre-amplifier gain of ×1 is selected)
±23.4 mV (channels 1, 3, and 5, when pre-amplifier gain of ×16 is selected)
○ Reference voltage generation (1.226 V (TYP.) can be output)
○ Conversion rate selectable (4.340 kHz or 2.170 kHz)

12.2 Configuration

The A/D converter consists of the following hardware.

Figure 12-1. Block Diagram of A/D Converter

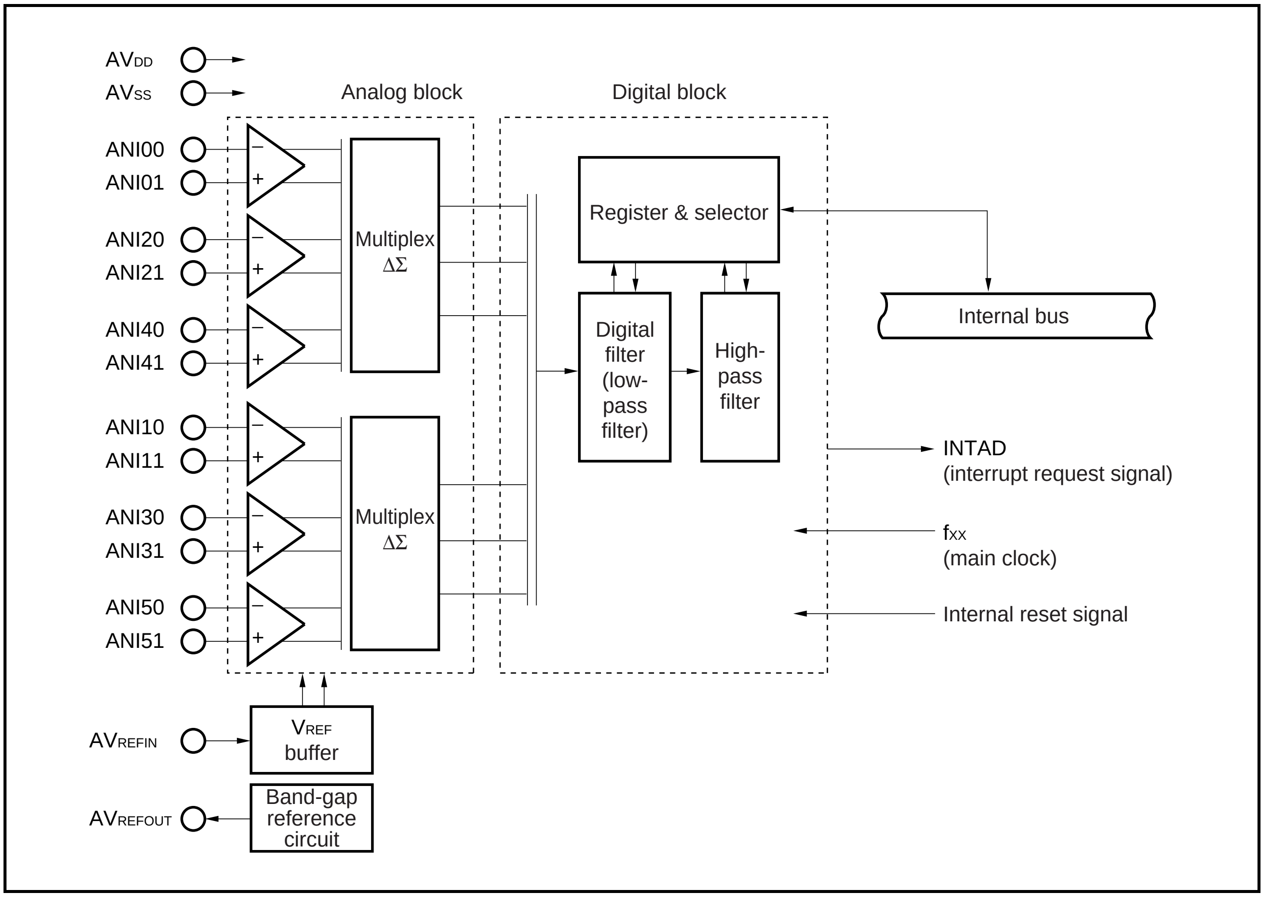


Table 12-1. Configuration of A/D Converter

Item	Configuration
Analog input	6 channels and 12 inputs (ANIn0 and ANIn1 pins (n = 0 to 5)) 2 inputs/channel
Registers	A/D converter mode register (ADM) High-pass filter control register 0 (HPFC0) A/D conversion result register n (ADCRn) (n = 0 to 5) A/D clock delay setting register (ADLY)
Internal units	Pre-amplifier block ΔΣ converter Reference voltage generator Digital filter (DF) High-pass filter (HPF)

(1) Pre-amplifier

This unit shifts the signal input to the ANIn0 and ANIn1 pins with AVSS as the reference voltage into the internal reference voltage, and then amplifies the input signal. It supplies its output signal to the ΔΣ circuit (n = 0 to 5).

(2) Multiplex ΔΣ circuit

Two 3-multiplex ΔΣ circuits are provided so that a total of 6 channels of analog inputs can be converted into digital signals. These two ΔΣ circuits operate synchronously, and one ΔΣ circuit executes analog input conversion of three channels by time division. The input signal is amplified by the pre-amplifier and ΔΣ circuit, and the gain of channels 0, 2, and 4 is fixed to ×1, and that of channels 1, 3, and 5 can be selected from ×2 and ×16. A high-speed mode (4.340 kHz) and a low-speed mode (2.170 kHz) are selectable as the conversion rate, and the over-sampling frequency in the respective modes is 555.6 kHz and 277.8 kHz (at f_X = 20 MHz).

(3) Reference voltage generator

An internal reference voltage source (band-gap reference circuit) is provided and a reference voltage is output from the reference voltage output pin (AV_{REFOUT}). To use the internal reference voltage source, short-circuit the AV_{REFOUT} pin and reference voltage input pin (AV_{REFIN}) outside the V850ES/PM1. To use an external reference voltage source, input its voltage to the AV_{REFIN} pin and leave AV_{REFOUT} open.
It is recommended to connect a capacitor of 10 μF to the AV_{REFIN} pin for stabilization.

(4) Digital filter (DF)

This unit eliminates high harmonic noise included in the ΔΣ circuit and thins out the data rate to 1/128.

(5) High-pass filter

This unit eliminates the DC component included in the input signal and the DC offset generated by the analog circuit. Whether the high-pass filter is inserted or not can be selected for each channel.

(6) ANIn0 to ANIn1 pins (n = 0 to 5)

These are analog input pins of the A/D converter. One channel inputs two signals. The ANIn0 pin is the negative input, while the ANIn1 pin is the positive input.

(7) AV_{DD} pin

This is the analog power supply pin of the A/D converter. Always keep the voltage on this pin the same as that on the V_{DD} pin even when the A/D converter is not used.

(8) AV_{SS} pin

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

(9) AV_{REFIN} pin

This pin inputs a reference voltage to the A/D converter. When the internal reference voltage is to be used, connect this pin to the AV_{REFOUT} pin. To use an external reference voltage, input the voltage from the AV_{REFIN} pin.

(10) AV_{REFOUT} pin

This pin outputs an internally generated reference voltage for the A/D converter.
Leave this pin open when the AV_{REFOUT} pin is not used.

12.3 Registers

The A/D converter is controlled by the following registers.

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

ADM is a register that controls the operation of the A/D converter and specifies the gain of pre-amplifier and conversion rate.

This register can be read or written in 8-bit or 1-bit units.

This register is set to 00H after reset.

After reset: 00H R/W Address: FFFFF200H

	<7>	<6>	5	4	3	<2>	1	<0>
ADM	ADPON	ADCE	0	0	0	PAGS1	0	FR

ADPON	Specification of power to A/D converter
0	Power OFF
1	Power ON

ADCE	Specification of operation of A/D converter
0	Stop conversion operation
1	Enable conversion operation

PAGS1	Specification of programmable amplifier gain of channels 1, 3, and 5
0	×2
1	×16

- Use channels 1, 3, and 5 for current measurement.
- Use channels 0, 2, and 4 for voltage measurement (the gain of these channels is fixed to ×1).

FR	Specification of conversion rate
0	High speed (4.340 kHz)
1	Low speed (2.170 kHz)

Caution Be sure to clear bits 1 and 3 to 5 to “0”.

(2) High-pass filter control register 0 (HPFC0)

HPFC0 is an 8-bit register that specifies insertion of a high-pass filter for each channel.

This register can be read or written in 8-bit or 1-bit units.

This register is set to 00H after reset.

After reset: 00H R/W Address: FFFFF202H

	7	6	<5>	<4>	<3>	<2>	<1>	<0>
HPFC0	0	0	THR5	THR4	THR3	THR2	THR1	THR0

THRn	Specification of insertion of high-pass filter for channel n (n = 0 to 5)
0	Insert high-pass filter
1	Do not insert high-pass filter

- If the high-pass filter is not inserted, the scale of the A/D output is decreased to 1/2 to avoid overload due to the DC element.

(3) A/D conversion result registers 0 to 5 (ADCR0 to ADCR5)

The ADCR0 to ADCR5 registers are 16-bit registers that store the conversion result of each channel.

These registers are read-only, in 16-bit units.

The value of these registers is initialized to 0000H by system reset and when the ADM.ADCE bit is 0.

After reset: 0000H R/W Address: ADCR0 FFFFF204H, ADCR1 FFFFF206H, ADCR2 FFFFF208H,
ADCR3 FFFFF20AH, ADCR4 FFFFF20CH, ADCR3 FFFFF20EH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCRn (n = 0 to 5)																

Caution Read the ADCRn register when the ADCE bit is 1, because the register is initialized when the ADCE bit is 0.

(4) A/D clock delay setting register (ADLY)

ADLY is a register that controls the phase between the A/D operation clock and the digital clock. Be sure to set this register to 00H.

This register can be read or written in 8-bit units.

This register is set to 00H after reset.

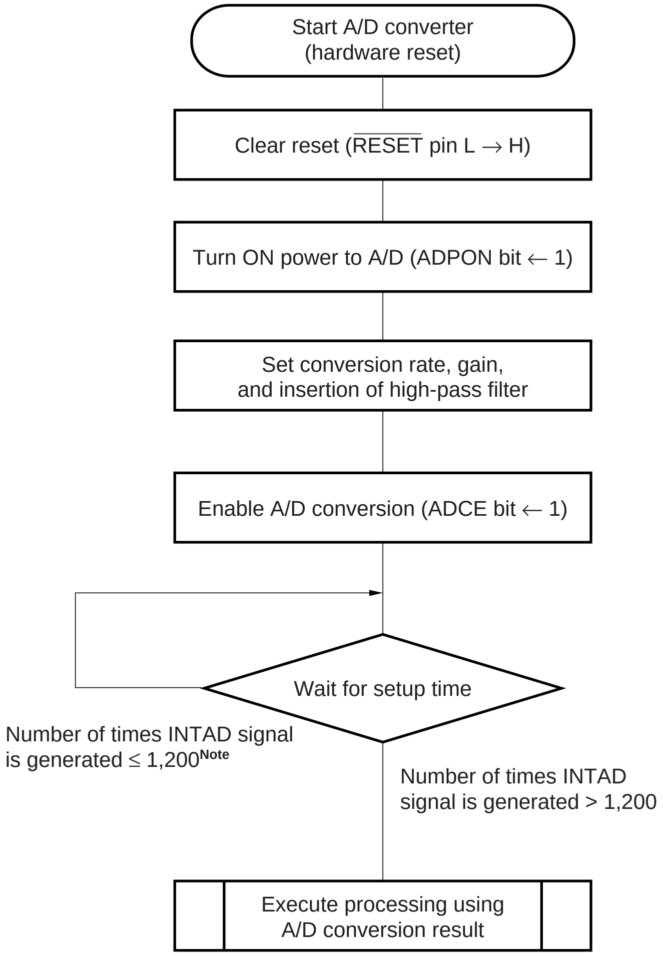
After reset: 00H R/W Address: FFFFF201H

	7	6	5	4	3	2	1	0
ADLY								

12.4 Operation

The A/D converter starts operating when the ADM.ADPON bit and ADM.ADCE bit are set to 1. The setup time of the analog block and digital filter block is required after power application and start of conversion. Perform initialization in accordance with the flowchart below.

Figure 12-2. Initialization Flowchart


Start A/D converter (hardware reset)
Clear reset ($\overline{\text{RESET}}$ pin L → H)
Turn ON power to A/D (ADPON bit ← 1)
Set conversion rate, gain, and insertion of high-pass filter
Enable A/D conversion (ADCE bit ← 1)
Wait for setup time
Number of times INTAD signal is generated ≤ 1,200Note
Number of times INTAD signal is generated > 1,200
Execute processing using A/D conversion result

Note The setup time (the number of times the INTAD signal is to be generated) when the ADPON bit is set to 1 is subject to change. Consult NEC Electronics before using this function.

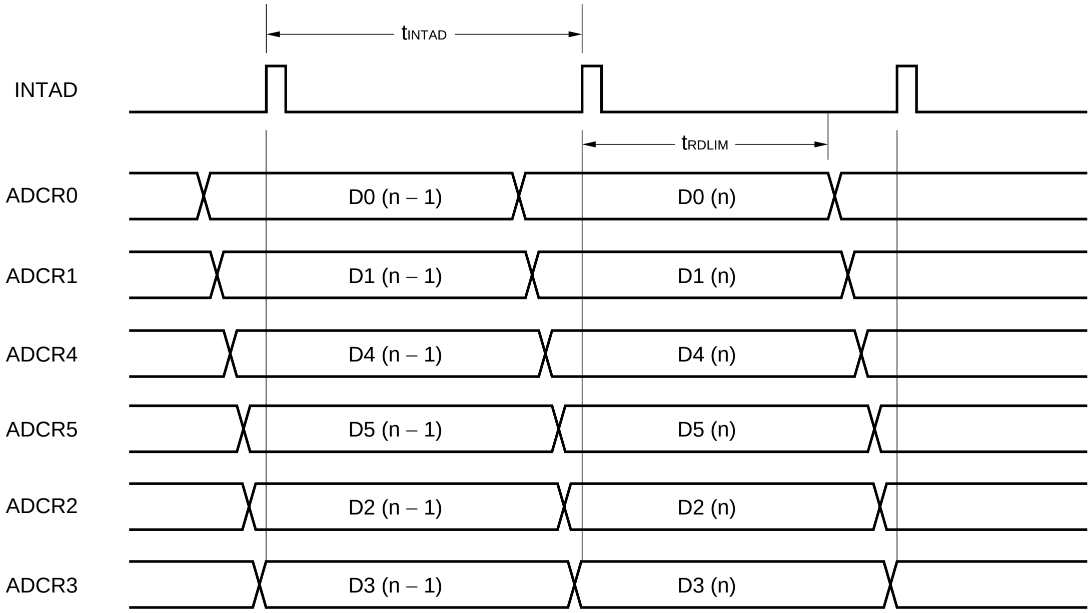
Caution If the A/D converter is temporarily stopped for initialization (ADCE bit ← 0 with ADPON bit set to 1) and then restarted, it is necessary to wait for a certain setup time. In this case, the setup time should be equal to 10 A/D conversion end interrupt request signals (INTAD), which is the delay of the digital filter.

When A/D conversion is enabled, conversion of the signals on the six channels of analog input pins (ANIn0 and ANIn1 pins) is started. Two sets of 3-multiplex ΔΣ circuits are provided, each of which executes conversion of three channels by time division. Each time conversion of all the six channels is completed, the INTAD signal is generated to inform the CPU that the conversion result can be read.

The cycle in which the INTAD signal is to be generated (t_{INTAD}) differs depending on the conversion rate specified by the FR bit of the ADM register. To read the ADCRn register by interrupt servicing, the maximum pending time is as shown in Figure 12-3. Complete reading of the ADCRn register within this time.

Remark n = 0 to 5

Figure 12-3. Timing of Generation of INTAD Signal and Storing in ADCRn Register (f_{XX} = 20 MHz)



t_{INTAD}: Interrupt generation cycle: 230.4 μs when FR bit = 0, 460.8 μs when FR bit = 1
t_{RDLIM}: ADCRn register read pending time (MAX.): t_{INTAD} – 5.5 × A/D system clock (f_{XX}/12)
227.1 μs when FR bit = 0, 454.2 μs when FR bit = 1

12.5 Cautions

(1) Read the ADCRn register by A/D conversion end interrupt (INTAD) servicing. Otherwise, an illegal value may be read because of a conflict between storing the conversion value in the ADCRn register and reading the register. The period of the INTAD processing during which reading the ADCRn register is held pending differs depending on the specified conversion speed, and is 227.1 μs when the ADM.FR bit is 0 and 454.2 μs when the FR bit is 1 (at 20 MHz).

(2) After turning ON power to the A/D converter (ADM.ADPON bit is set to 1), the internal setup time of the A/D converter is necessary. Consequently, the data of the first 1,200 conversions is invalid.

(3) The setup time is also necessary when the A/D converter has been temporarily stopped once for initialization (by clearing the ADM.ADCE bit with the ADPON bit set to 1) and then restarted. Wait for the duration of 10 INTAD signals, which is the delay of the digital filter.

(4) The time required for the correct data to be output after the conversion operation has been enabled (by setting the ADM.ADCE bit to 1) differs depending on the analog input status at that time. This is because the stabilization time of the high-pass filter changes depending on the analog input status.

(5) Be sure to set the conversion speed and gain, and the HPFC0 and ADLY registers while the A/D converter is stopped (ADCE bit = 0).

(6) Because the ADCRn register is initialized when the ADCE bit is 0, read the ADCRn register when the ADCE bit is 1.

★ (7) Set the ADPON bit to 0 before shifting to the software STOP mode. If software STOP mode is entered with the ADPON bit set to 1, a current will flow.

Cautions 1. Count the INTAD signal 1,200 times after the A/D converter is started and then load the converted data when the next INTAD signal is generated. The setup time is subject to change. Consult NEC Electronics before using the setup time.

2. Thoroughly evaluate the stabilization time in the environment in which the A/D converter is used.

Remark n = 0 to 5

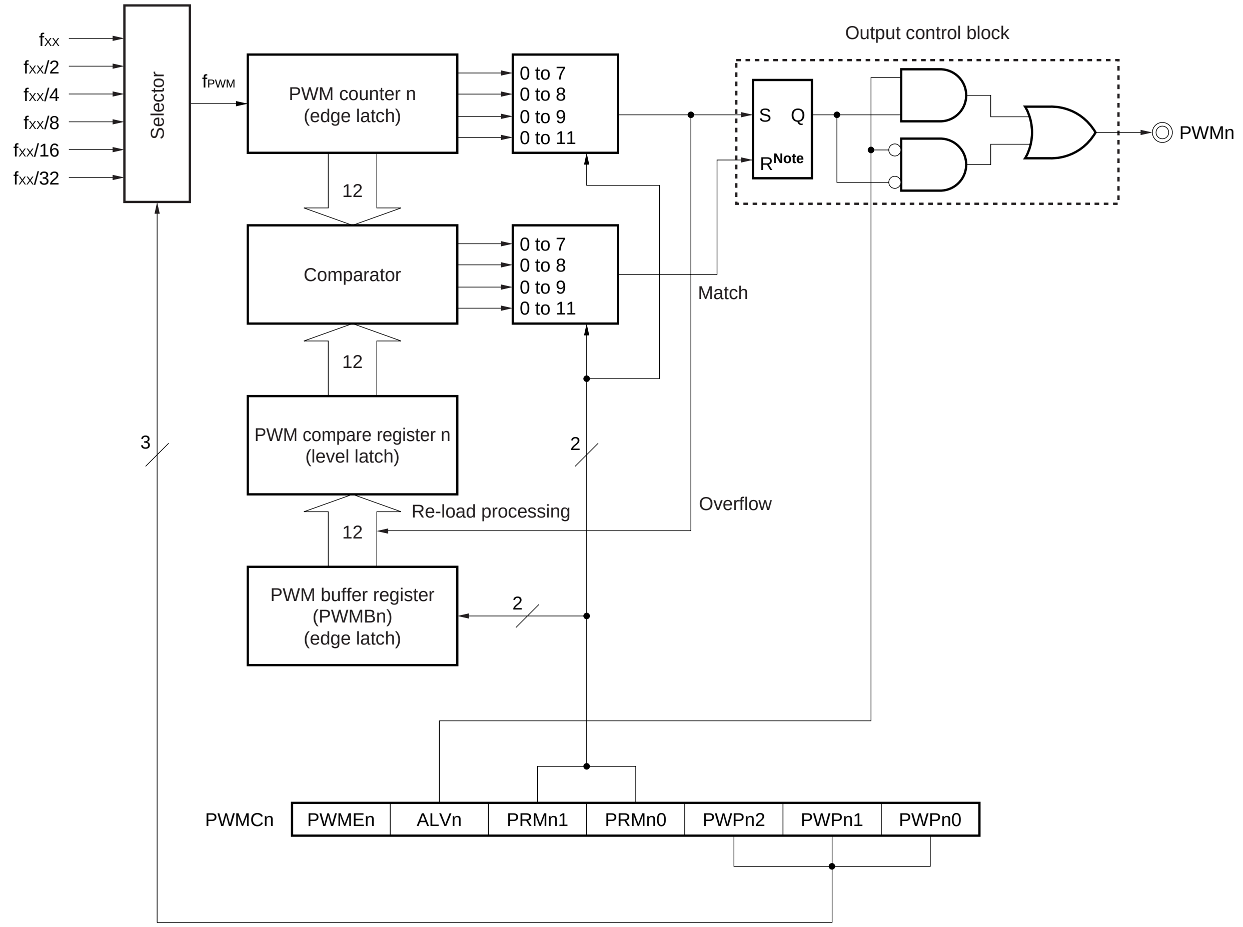
13.1 Features

○ PWMn: 4 channels
○ Active level of PWMn output pulse selectable
○ Operation clock: Selectable from fxx, fxx/2, fxx/4, fxx/8, fxx/16, and fxx/32
○ PWMn output resolution: Selectable from 8, 9, 10, and 12 bits

Remark n = 0 to 3

13.2 Configuration

Figure 13-1. Block Diagram of PWM Function



Note Reset function

Remark n = 0 to 3

13.3 Registers

(1) PWM control register n (PWMCn)

PWMCn is a register that controls the operation of PWMn.

This register can be read or written in 8-bit or 1-bit units.

This register is set to 40H after reset.

Caution To use PWMn, be sure to set the external pins related to PWMn to the control mode. Then set the operation clock by using the PWMCn register, set the PWMBn register, and set the PWMEn bit to 1.

After reset: 40H   R/W   Address: PWMC0 FFFFFB00H, PWMC1 FFFFFB10H,
PWMC2 FFFFFB20H, PWMC3 FFFFFB30H

	<7>	<6>	5	4	3	2	1	0
PWMCn (n = 0 to 3)	PWMEn	ALVn	PRMn1	PRMn0	0	PWPn2	PWPn1	PWPn0

PWMEn	PWMn operation enable/disable
0	Stop PWMn operation
1	Enable PWMn operation

ALVn	Specification of active level of PWMn
0	Active-low
1	Active-high

PRMn1	PRMn0	Specification of bit length of counter and comparator
0	0	8 bits
0	1	9 bits
1	0	10 bits
1	1	12 bits

PWPn2	PWPn1	PWPn0	Specification of operation clock of PWMn
0	0	1	f_{XX}
0	1	0	$f_{XX}/2$
0	1	1	$f_{XX}/4$
0	0	0	$f_{XX}/8$
1	0	1	$f_{XX}/16$
1	0	1	$f_{XX}/32$
Other than above			Setting prohibited

Note When the PWMEn bit is set to 1 from 0, PWM counter n is reset and starts counting from 000H (if the bit length is specified as 12 bits). When the counter overflows for the first time, the PWMn signal is asserted.

PWM counter n cannot be reset by writing 1 to the PWMEn bit while the PWMEn bit is already 1. Clear the bit to 0 once, and then write 1 to it.

Remark n = 0 to 3

(2) PWM buffer register n (PWMBn)

PWMBn is a 12-bit buffer register that sets control data of the active signal width of the PWMn output. Bits 15 to 12 of this register are fixed to 0 by hardware.

The contents of the PWMBn register are transferred to PWM compare register n when PWM counter n, which controls PWMn output, overflows.

This register can be read or written in 16-bit units.

This register is set to 0000H after reset.

★ **Caution When the main clock is stopped and the CPU is operating on the subclock, do not access the PWMBn register using an access method that causes a wait.**
For details, refer to 3.4.8 (2).

Remark n = 0 to 3

After reset: 0000H R/W Address: PWMB0 FFFFFB02H, PWMB1 FFFFFB12H,
PWMB2 FFFFFB22H, PWMB3 FFFFFB32H

	15	14	13	12	11	
PWMBn (n = 0 to 3)	0	0	0	0	0	PWMBn11 to PWMBn0

Caution To execute writing to the PWMBn register during PWMn operation, the access time is extended by a control action to synchronize the operation clock. How much the access time is to be extended differs depending on the specified PWMn operation clock. It is the shortest at 1 μs (f_{XX} = 20 MHz) when f_{XX} is selected. The access time is lengthened as the operating clock frequency decreases, and the maximum access time is about 4 μs (f_{XX} = 20 MHz) when f_{XX}/32 is selected.

13.4 Operation

13.4.1 Basic operation

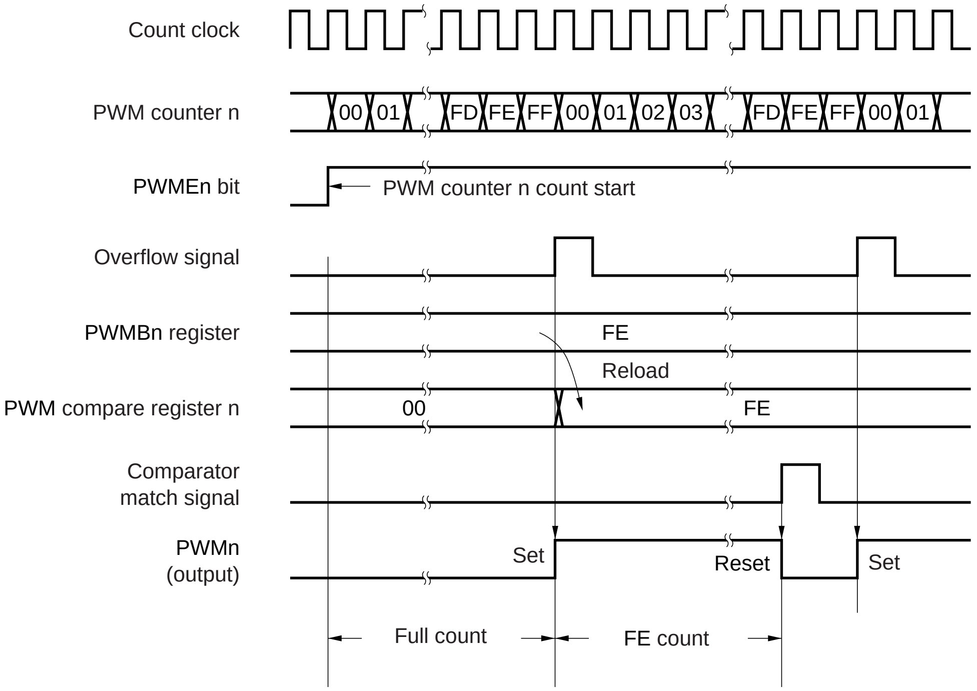
To output the PWMn pulse, set the necessary data to the PWMCn and PWMBn registers, and set the PWMCn.PWMEn bit to 1. As a result, PWM counter n is cleared (to 000H). When the counter overflows for the first time, the active level of the PWMn output is set and the data of the PWMBn register is transferred to PWM compare register n. After that, the PWMn output is deasserted when the value of PWM counter n matches that of PWM compare register n. This is repeated and the PWMn signal of the active level specified by the ALVn bit of the PWMCn register is output from the PWMn pin.

When the PWMCn.PWMEn bit is cleared to 0, the PWMn output is immediately disabled, and the PWMn output goes to the inactive level specified by the PWMCn.ALVn bit.

If the PWMCn.PWPn0 to PWMCn.PWPn2 bits, PWMCn.PRM0 or PWMCn.PRM1 bit, or PWMCn.ALVn bit are changed while the PWMn signal is being output, the cycle width and pulse width of the PWMn signal cannot be guaranteed during the period in which the change is made.

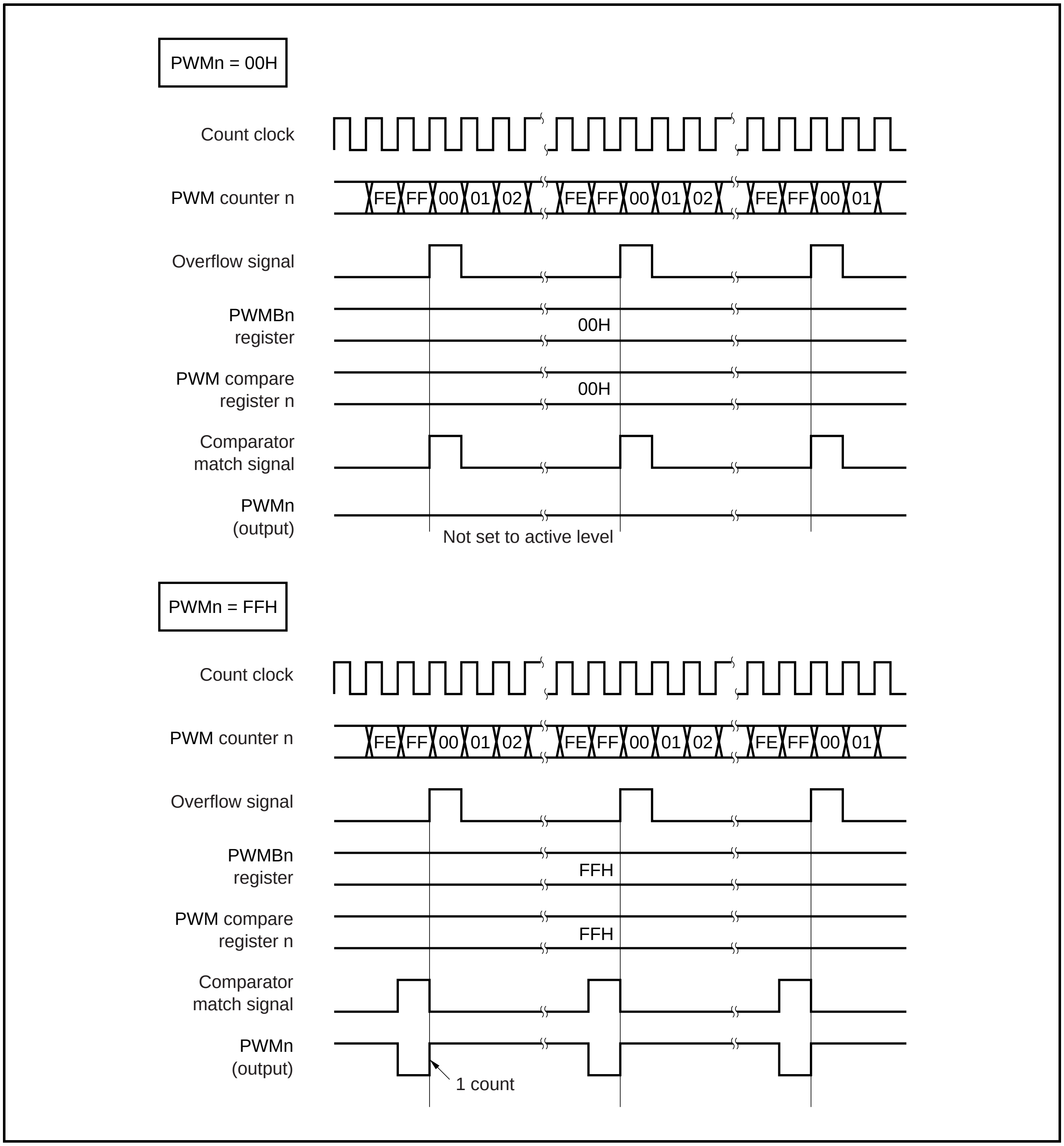
Remark n = 0 to 3

★ **Figure 13-2. PWMn Operation Timing**



Remark n = 0 to 3

★ **Figure 13-3. Operation Timing When PWMBn Register Is Set to 00H/FFH**



13.4.2 Repeat frequency

The repeat frequency of PWMn output is shown below (n = 0 to 3).

PWMn Operation Frequency	Resolution	Repeat Frequency ( ): Value at f_{XX} = 20 MHz
f_{XX}	8 bits	$f_{XX}/2^{8}$ (approx. 78.13 kHz)
	9 bits	$f_{XX}/2^{9}$ (approx. 39.06 kHz)
	10 bits	$f_{XX}/2^{10}$ (approx. 19.53 kHz)
	12 bits	$f_{XX}/2^{12}$ (approx. 4.88 kHz)
$f_{XX}/2$	8 bits	$f_{XX}/2^{9}$ (approx. 39.06 kHz)
	9 bits	$f_{XX}/2^{10}$ (approx. 19.53 kHz)
	10 bits	$f_{XX}/2^{11}$ (approx. 9.77 kHz)
	12 bits	$f_{XX}/2^{13}$ (approx. 2.44 kHz)
$f_{XX}/4$	8 bits	$f_{XX}/2^{10}$ (approx. 19.53 kHz)
	9 bits	$f_{XX}/2^{11}$ (approx. 9.77 kHz)
	10 bits	$f_{XX}/2^{12}$ (approx. 4.88 kHz)
	12 bits	$f_{XX}/2^{14}$ (approx. 1.22 kHz)
$f_{XX}/8$	8 bits	$f_{XX}/2^{11}$ (approx. 9.77 kHz)
	9 bits	$f_{XX}/2^{12}$ (approx. 4.88 kHz)
	10 bits	$f_{XX}/2^{13}$ (approx. 2.44 kHz)
	12 bits	$f_{XX}/2^{15}$ (approx. 610 Hz)
$f_{XX}/16$	8 bits	$f_{XX}/2^{12}$ (approx. 4.88 kHz)
	9 bits	$f_{XX}/2^{13}$ (approx. 2.44 kHz)
	10 bits	$f_{XX}/2^{14}$ (approx. 1.22 kHz)
	12 bits	$f_{XX}/2^{16}$ (approx. 305 Hz)
$f_{XX}/32$	8 bits	$f_{XX}/2^{13}$ (approx. 2.44 kHz)
	9 bits	$f_{XX}/2^{14}$ (approx. 1.22 kHz)
	10 bits	$f_{XX}/2^{15}$ (approx. 610 Hz)
	12 bits	$f_{XX}/2^{17}$ (approx. 153 Hz)

13.5 Cautions

Each PWMn pin (n = 0 to 3) functions alternately as the P1n pin (n = 0 to 3) of port 1. To use the PWMn pin, set the corresponding bit of the PMC1 register to 1. For the pin that also functions alternately as a timer output as well as a port pin (P11, P12, and P13), the PFC1 register is used to specify the alternate function. Set the corresponding bit of this register to 1. The setting values of the PMC1 and PFC1 registers when PWMn is output are shown below.

If the setting of the corresponding bits of the PMC1 and PFC1 registers is changed while the PWMn pulse is being output, the PWMn pulse output cannot be guaranteed.

Pin	Function	PMC1 Register Setting	PFC1 Register Setting
P10	PWM0	PMC10 bit = 1	Setting unnecessary
P11	TO00	PMC11 bit = 1	PFC11 bit = 0
	PWM1	PMC11 bit = 1	PFC11 bit = 1
P12	TO01	PMC12 bit = 1	PFC12 bit = 0
	PWM2	PMC12 bit = 1	PFC12 bit = 1
P13	TO20	PMC13 bit = 1	PFC13 bit = 0
	PWM3	PMC13 bit = 1	PFC13 bit = 1

CHAPTER 14 ASYNCHRONOUS SERIAL INTERFACE n (UARTn)

14.1 Features

- Transfer rate: 300 bps to 312.5 kbps (using a dedicated baud rate generator and an internal system clock of 20 MHz)
- Full-duplex communications
  On-chip receive buffer register n (RXBn)
  On-chip transmit buffer register n (TXBn)
- Two-pin configuration
  TXDn: Transmit data output pin
  RXDn: Receive data input pin
- Reception error detection functions
  - Parity error
  - Framing error
  - Overrun error
- Interrupt sources: 3 types
  - Reception error interrupt (INTSREn): Interrupt is generated according to the logical OR of the three types of reception errors
  - Reception completion interrupt (INTSRn): Interrupt is generated when receive data is transferred from the shift register to receive buffer register n after serial transfer is completed during a reception enabled state
  - Transmission completion interrupt (INTSTn): Interrupt is generated when the serial transmission of transmit data (8 or 7 bits) from the shift register is completed
- Character length: 7 or 8 bits
- Parity functions: Odd, even, 0, or none
- Transmission stop bits: 1 or 2 bits
- On-chip dedicated baud rate generator

Remark n = 0, 1

14.2 Configuration

Table 14-1. Configuration of UART

Item	Configuration
Registers	Receive buffer register n (RXBn) Transmit buffer register n (TXBn) Receive shift register Transmit shift register Asynchronous serial interface mode register n (ASIMn) Asynchronous serial interface status register n (ASISn) Asynchronous serial interface transmit status register n (ASIFn)
Other	Reception control parity check Addition of transmission control parity

Figure 14-1 shows the configuration of asynchronous serial interface n (UARTn).

(1) Asynchronous serial interface mode register n (ASIMn)

ASIMn is an 8-bit register that specifies the operation of the asynchronous serial interface.

(2) Asynchronous serial interface status register n (ASISn)

ASISn consists of a set of flags that indicate the error contents when a reception error occurs. The various reception error flags are set (1) when a reception error occurs and are reset (0) when the ASISn register is read.

(3) Asynchronous serial interface transmit status register n (ASIFn)

ASIFn is an 8-bit register that indicates the status when a transmit operation is performed.

This register consists of a transmit buffer data flag, which indicates the hold status of the data of the TXBn register, and the transmit shift register data flag, which indicates whether transmission is in progress.

(4) Reception control parity check

The receive operation is controlled according to the contents set in the ASIMn register. A check for parity errors is also performed during a receive operation, and if an error is detected, a value corresponding to the error contents is set in the ASISn register.

(5) Receive shift register

This is a shift register that converts the serial data that was input to the RXDn pin into parallel data. One byte of data is received, and if a stop bit is detected, the receive data is transferred to the RXBn register.

This register cannot be directly manipulated.

(6) Receive buffer register n (RXBn)

RXBn is an 8-bit buffer register for holding receive data. When 7 characters are received, 0 is stored in the MSB.

During a reception enabled state, receive data is transferred from the receive shift register to the RXBn register, synchronized with the end of the shift-in processing of one frame.

Also, the reception completion interrupt request (INTSRn) is generated by the transfer of data to the RXBn register.

(7) Transmit shift register

This is a shift register that converts the parallel data that was transferred from the TXBn register into serial data.

When one byte of data is transferred from the TXBn register, the shift register data is output from the TXDn pin.

The transmission completion interrupt request (INTSTn) is generated synchronized with the completion of transmission of one frame.

This register cannot be directly manipulated.

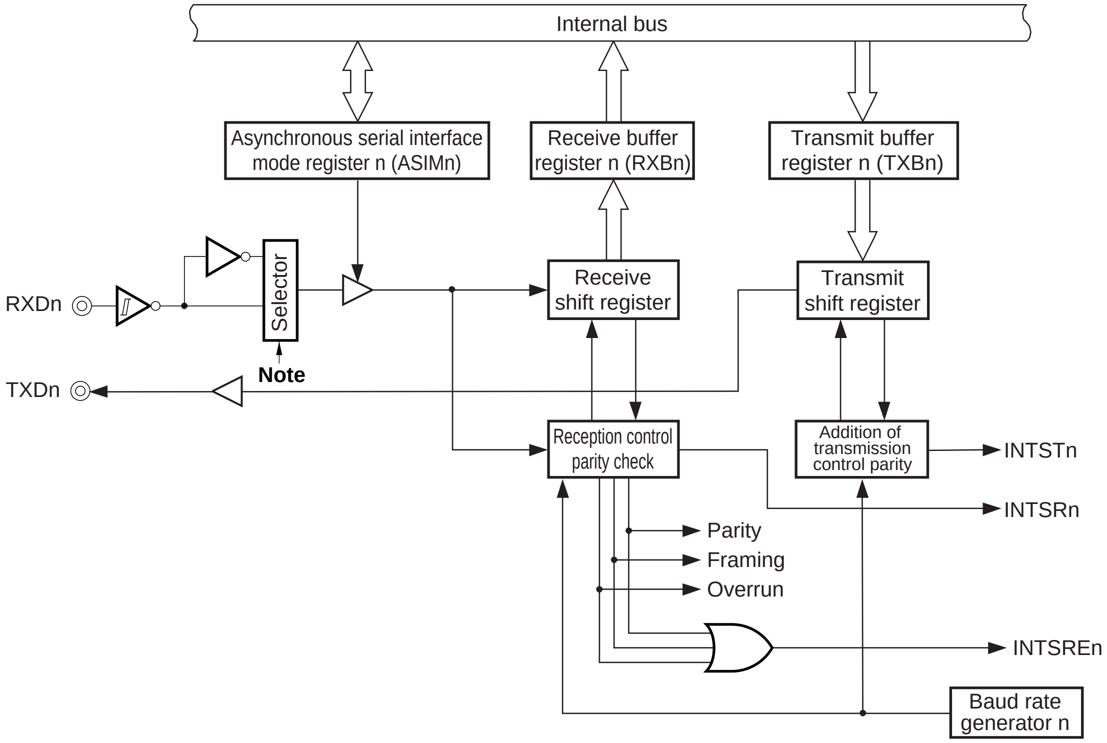
(8) Transmit buffer register n (TXBn)

The TXBn register is an 8-bit buffer for transmit data. A transmit operation is started by writing transmit data to the TXBn register.

(9) Addition of transmission control parity

A transmit operation is controlled by adding a start bit, parity bit, or stop bit to the data that is written to the TXBn register, according to the contents that were set in the ASIMn register.

Figure 14-1. Block Diagram of Asynchronous Serial Interface n


Internal bus
Asynchronous serial interface mode register n (ASIMn)
Receive buffer register n (RXBn)
Transmit buffer register n (TXBn)
Selector
Receive shift register
Transmit shift register
RXDn
TXDn
Note
Reception control parity check
Addition of transmission control parity
INTSTn
INTSRn
Parity
Framing
Overrun
INTSREn
Baud rate generator n

Note Use the PFC30 bit to switch RXD0 and the PFC43 bit to switch RXD1.

Remarks 1. For the configuration of the baud rate generator, refer to **Figure 14-12**.

2. n = 0, 1

14.3 Registers

(1) Asynchronous serial interface mode register n (ASIMn)

ASIMn is an 8-bit register that controls the UARTn transfer operation.
This register can be read or written in 8-bit or 1-bit units.
After reset, this register is set to 01H.

Cautions 1. When using UARTn, be sure to set the external pins related to UARTn functions to the control mode before setting clock select register n (CKSRn) and baud rate generator control register n (BRGCn), and then set the UARTCAEn bit to 1. Then set the other bits.

★ **2. Set the UARTCAEn and RXEn bits to 1 while a high level is being input to the RXDn pin. If these bits are set to 1 while a low level is being input to the RXDn pin, reception will be started.**

(1/3)

After reset: 01H R/W Address: ASIM0 FFFFFA00H, ASIM1 FFFFFA10H

	<7>	<6>	<5>	4	3	2	1	0
ASIMn	UARTCAEn	TXEn	RXEn	PSn1	PSn0	CLn	SLn	ISRMn

UARTCAEn	Controls the operating clock
0	Stops clock supply to UARTn.
1	Supplies clock to UARTn.

- If UARTCAEn bit = 0, UARTn is asynchronously reset[Note].
- If UARTCAEn bit = 0, UARTn is reset. To operate UARTn, first set the UARTCAEn bit to 1.
- If the UARTCAEn bit is changed from 1 to 0, all the registers of UARTn are initialized. To set the UARTCAEn bit to 1 again, be sure to re-set the registers of UARTn.
- The output of the TXDn pin goes high when transmission is disabled, regardless of the setting of the UARTCAEn bit.

TXEn	Enables/disables transmission
0	Disables transmission
1	Enables transmission

- Set the TXEn bit to 1 after setting the UARTCAEn bit to 1 at startup. Set the UARTCAEn bit to 0 after setting the TXEn bit to 0 to stop.
- To initialize the transmission unit, clear (0) the TXEn bit, and after letting 2 Clock cycles (base clock) elapse, set (1) the TXEn bit again. If the TXEn bit is not set again, initialization may not be successful. (For details of the base clock, refer to **14.6 (1) (a) Base clock (Clock)**.)

Note The ASISn, ASIFn, and RXBn registers are reset.

(2/3)

RXEn	Enables/disables reception
0	Disables reception[Note]
1	Enables reception

- Set the RXEn bit to 1 after setting the UARTCAEn bit to 1 at startup. Set the UARTCAEn bit to 0 after setting the RXEn bit to 0 to stop.
- To initialize the reception unit status, clear (0) the RXEn bit, and after letting 2 Clock cycles (base clock) elapse, set (1) the RXEn bit again. If the RXEn bit is not set again, initialization may not be successful. (For details of the base clock, refer to **14.6 (1) (a) Base clock (Clock)**.)

PSn1	PSn0	Transmit operation	Receive operation
0	0	Parity bit not output	Receive with no parity
0	1	Output 0 parity	Receive as 0 parity
1	0	Output odd parity	Judge as odd parity
1	1	Output even parity	Judge as even parity

- To overwrite the PSn1 and PSn0 bits, first clear (0) the TXEn and RXEn bits.
- If “0 parity” is selected for reception, no parity judgment is performed. Therefore, no parity error interrupt is generated because the ASISn.PE bit is not set.

Note When reception is disabled, the receive shift register does not detect a start bit. No shift-in processing or transfer processing to the RXBn register is performed, and the contents of the RXBn register are retained.

When reception is enabled, the receive shift operation starts, synchronized with the detection of the start bit, and when the reception of one frame is completed, the contents of the receive shift register are transferred to the RXBn register. A reception completion interrupt (INTSRn) is also generated in synchronization with the transfer to the RXBn register.

(3/3)

CLn	Specifies character length of 1 frame of transmit/receive data
0	7 bits
1	8 bits
• To overwrite the CLn bit, first clear (0) the TXEn and RXEn bits.	

SLn	Specifies stop bit length of transmit data
0	1 bit
1	2 bits
• To overwrite the SLn bit, first clear (0) the TXEn bit. • Since reception is always performed with a stop bit length of 1, the SL bit setting does not affect receive operations.	

ISRMn	Enables/disables generation of reception completion interrupt requests when an error occurs
0	Generate a reception error interrupt request (INTSREn) as an interrupt when an error occurs. In this case, no reception completion interrupt request (INTSRn) is generated.
1	Generate a reception completion interrupt request (INTSRn) as an interrupt when an error occurs. In this case, no reception error interrupt request (INTSREn) is generated.
• To overwrite the ISRMn bit, first clear (0) the RXEn bit.	

(2) Asynchronous serial interface status register n (ASISn)

The ASISn register, which consists of 3 error flag bits (PEn, FEn and OVEn), indicates the error status when UARTn reception is complete.

The ASISn register is cleared to 00H by a read operation. When a reception error occurs, the RXBn register should be read and the error flag should be cleared after the ASISn register is read.

This register is read-only, in 8-bit units.

This register is set to 00H after reset.

Cautions 1. When the ASIMn.UARTCAEn bit or ASIMn.RXEn bit is set to 00, or when the ASISn register is read, the ASISn.PEn, ASISn.FEn, and ASISn.OVEn bits are cleared (0).

★ **2. Operation using a bit manipulation instruction is prohibited.**

After reset: 00H R Address: ASIS0 FFFFFA03H, ASIS1 FFFFFA13H

	7	6	5	4	3	2	1	0
ASISn	0	0	0	0	0	PEn	FEn	OVEn

PEn	Status flag that indicates a parity error
0	When the ASIMn.UARTCAEn and ASIMn.RXEn bits are both set to 0, or when the ASISn register has been read
1	When reception was completed, the transmit data parity did not match the parity bit
• The operation of the PEn bit differs according to the settings of the ASIMn.PS1 and ASIMn.PS0 bits.	

FEn	Status flag that indicates a framing error
0	When the ASIMn.UARTCAEn and ASIMn.RXEn bits are both set to 0, or when the ASISn register has been read
1	When reception was completed, no stop bit was detected
• For receive data stop bits, only the first bit is checked regardless of the stop bit length.	

OVEn	Status flag that indicates an overrun error
0	When the ASIMn.UARTCAEn and ASIMn.RXEn bits are both 0, or when the ASISn register has been read.
1	UARTn completed the next receive operation before reading the receive data of the RXBn register.
• When an overrun error occurs, the next receive data value is not written to the RXBn register and the data is discarded.	

(3) Asynchronous serial interface transmit status register n (ASIFn)

The ASIFn register, which consists of 2 status flag bits, indicates the status during transmission.

By writing the next data to the TXBn register after data is transferred from the TXBn register to the transmit shift register, transmit operations can be performed continuously without suspension even during an interrupt interval. When transmission is performed continuously, data should be written after referencing the TXBFn bit to prevent writing to the TXBn register by mistake.

This register is read-only, in 8-bit or 1-bit units.

This register is set to 00H after reset.

After reset: 00H R Address: ASIF0 FFFFFA05H, ASIF1 FFFFFA15H

	7	6	5	4	3	2	<1>	<0>
ASIFn	0	0	0	0	0	0	TXBFn	TXSFn

TXBFn	Transmit buffer data flag
0	Data to be transferred next to the TXBn register does not exist (when the ASIMn.UARTCAEn or ASIMn.TXEn bit is 0, or when data has been transferred to the transmit shift register)
1	Data to be transferred next exists in the TXBn register (data exists in the TXBn register when the TXBn register has been written to)

- When transmission is performed continuously, data should be written to the TXBn register after confirming that this flag is 0. If writing to the TXBn register is performed when this flag is 1, transmit data cannot be guaranteed.

TXSFn	Transmit shift register data flag (indicating the transmission status of UARTn)
0	Initial status or awaiting transmission (when the ASIMn.UARTCAEn or ASIMn.TXEn bit is set to 0, or following transfer completion, the next data transmission from the TXBn register is not performed)
1	Transmission in progress (when data has been transferred from the TXBn register)

- When the transmission unit is initialized, initialization should be executed after confirming that this flag is 0 following the occurrence of a transmission completion interrupt. If initialization is performed when this flag is 1, transmit data cannot be guaranteed.

(4) Receive buffer register n (RXBn)

RXBn is an 8-bit buffer register for storing parallel data that had been converted by the receive shift register.

When reception is enabled (ASIMn.RXEn bit = 1), receive data is transferred from the receive shift register to the RXBn register, synchronized with the completion of the shift-in processing of one frame. Also, a reception completion interrupt request (INTSRn) is generated by the transfer to the RXBn register. For information about the timing for generating this interrupt request, refer to **14.5 (4) Receive operation**.

If reception is disabled (ASIMn.RXEn bit = 0), the contents of the RXBn register are retained, and no processing is performed for transferring data to the RXBn register even when the shift-in processing of one frame is completed. Also, no reception completion interrupt is generated.

When 7 bits is specified for the data length, bits 6 to 0 of the RXBn register are transferred for the receive data and the MSB (bit 7) is always 0. However, if an overrun error (OVEn) occurs, the receive data at that time is not transferred to the RXBn register.

Except after reset, the RXBn register becomes FFH even when ASIMn.UARTCAEn = 0.

This register is read-only, in 8-bit units.

After reset: FFH R Address: RXB0 FFFFFA02H, RXB1 FFFFFA12H

	7	6	5	4	3	2	1	0
RXBn	RXBn7	RXBn6	RXBn5	RXBn4	RXBn3	RXBn2	RXBn1	RXBn0

(5) Transmit buffer register n (TXBn)

TXBn is an 8-bit buffer register for setting transmit data.

When transmission is enabled (ASIMn.TXEn bit = 1), the transmit operation is started by writing data to TXBn register.

When transmission is disabled (ASIMn.TXEn bit = 0), even if data is written to TXBn register, the value is ignored.

The TXBn register data is transferred to the transmit shift register, and a transmission completion interrupt request (INTSTn) is generated, synchronized with the completion of the transmission of one frame from the transmit shift register. For information about the timing for generating this interrupt request, refer to **14.5 (2) Transmit operation**.

When ASIFn.TXBFn bit = 1, the TXBn register must not be written.

This register can be read or written in 8-bit units.

This register is set to FFH after reset.

After reset: FFH R/W Address: TXB0 FFFFFA04H, TXB1 FFFFFA14H

	7	6	5	4	3	2	1	0
TXBn	TXBn7	TXBn6	TXBn5	TXBn4	TXBn3	TXBn2	TXBn1	TXBn0

14.4 Interrupt Requests

The following three types of interrupt requests are generated from UARTn.

- Reception error interrupt (INTSREn)
- Reception completion interrupt (INTSRn)
- Transmission completion interrupt (INTSTn)

The default priorities among these three types of interrupt requests is, from high to low, reception error interrupt, reception completion interrupt, and transmission completion interrupt.

Table 14-2. Generated Interrupts and Default Priorities

Interrupt	Priority
Reception error	1
Reception completion	2
Transmission completion	3

(1) Reception error interrupt (INTSREn)

When reception is enabled, a reception error interrupt is generated according to the logical OR of the three types of reception errors explained for the ASISn register. Whether a reception error interrupt (INTSREn) or a reception completion interrupt (INTSRn) is generated when an error occurs can be specified using the ASIMn.ISRMn bit.

When reception is disabled, no reception error interrupt is generated.

(2) Reception completion interrupt (INTSRn)

When reception is enabled, a reception completion interrupt is generated when data is shifted in to the receive shift register and transferred to the receive buffer register (RXBn).

A reception completion interrupt request can be specified to be generated in place of a reception error interrupt using the ASIMn.ISRMn bit even when a reception error has occurred.

When reception is disabled, no reception completion interrupt is generated.

(3) Transmission completion interrupt (INTSTn)

A transmission completion interrupt is generated when one frame of transmit data containing 7-bit or 8-bit characters is shifted out from the transmit shift register.

14.5 Operation

(1) Data format

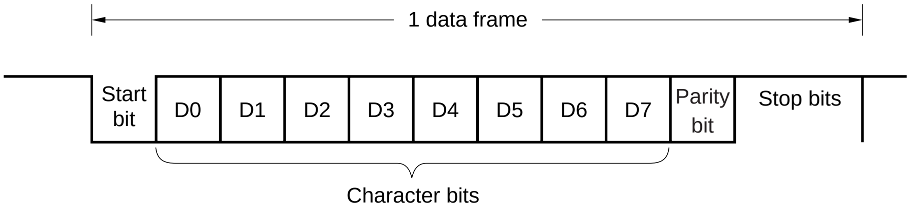
Full-duplex serial data transmission and reception can be performed.

The transmit/receive data format consists of one data frame containing a start bit, character bits, a parity bit, and stop bits as shown in Figure 14-2.

The character bit length within one data frame, the type of parity, and the stop bit length are specified by the ASIMn register.

Also, data is transferred LSB first.

Figure 14-2. Asynchronous Serial Interface Transmit/Receive Data Format



- Start bit .... 1 bit
- Character bits .... 7 bits or 8 bits
- Parity bit .... Even parity, odd parity, 0 parity, or no parity
- Stop bits .... 1 bit or 2 bits

(2) Transmit operation

When the ASIMn.UARTCAEn bit is set to 1, a high level is output from the TXDn pin.
Then, when the ASIMn.TXEn bit is set to 1, transmission is enabled, and the transmit operation is started by writing transmit data to the TXBn register.

(a) Transmission enabled state

This state is set by the ASIMn.TXEn bit.

- TXEn bit = 1: Transmission enabled state
- TXEn bit = 0: Transmission disabled state

Since UARTn does not have a CTS (transmission enabled signal) input pin, a port should be used to confirm whether the destination is in a reception enabled state.

(b) Starting a transmit operation

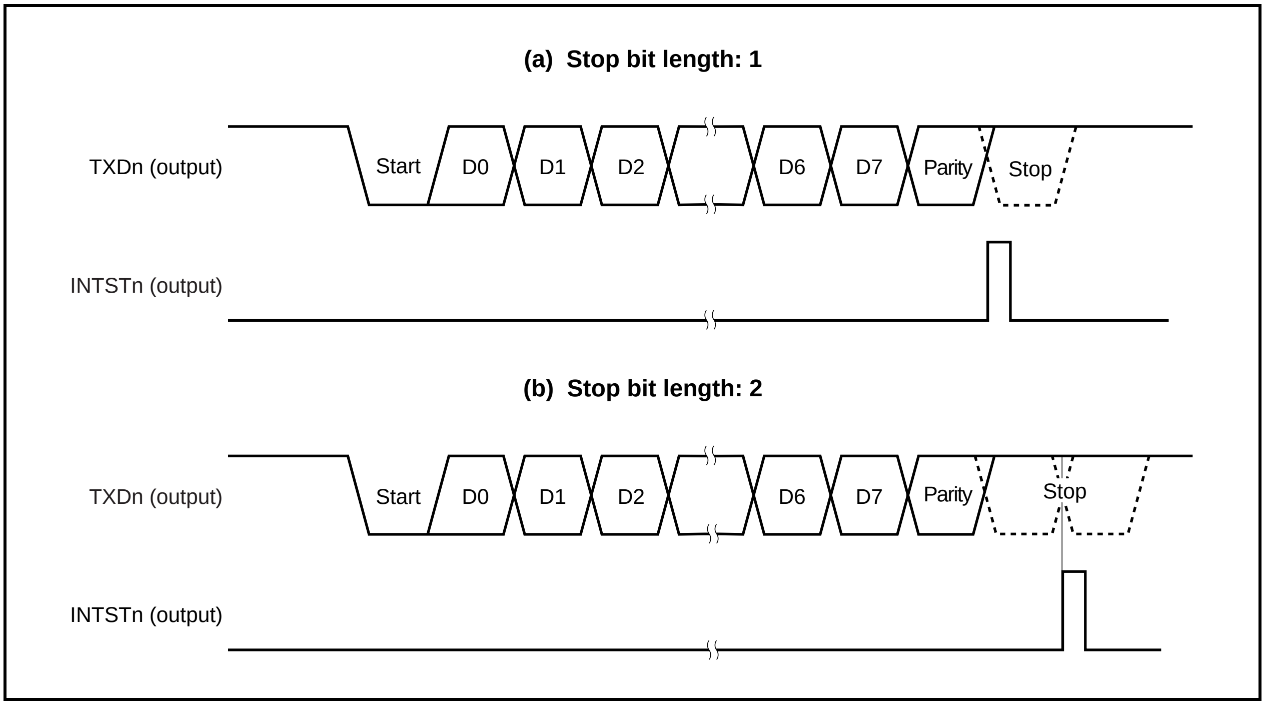
In the transmission enabled state, a transmit operation is started by writing transmit data to the TXBn register. When a transmit operation is started, the data in the TXBn register is transferred to the transmit shift register. Then, the transmit shift register outputs data to the TXDn pin (the transmit data is transferred sequentially starting with the start bit). The start bit, parity bit, and stop bits are added automatically.

(c) Transmission interrupt

When the transmit shift register becomes empty, a transmission completion interrupt (INTSTn) is generated. The timing for generating the INTSTn interrupt differs according to the specification of the stop bit length. The INTSTn interrupt is generated at the same time that the last stop bit is output.
If the data to be transmitted next has not been written to the TXBn register, the transmit operation is suspended.

Caution Normally, when the transmit shift register becomes empty, a transmission completion interrupt (INTSTn) is generated. However, no transmission completion interrupt (INTSTn) is generated if the transmit shift register becomes empty due to a reset.

Figure 14-3. Asynchronous Serial Interface Transmission Completion Interrupt Timing



(3) Continuous transmission operation

UARTn can write the next transmit data to the TXBn register at the timing that the transmit shift register starts the shift operation. This enables an efficient transmission rate to be realized by continuously transmitting data even during the INTSTn interrupt servicing after the transmission of one data frame. In addition, reading the ASIFn.TXSFn bit after the occurrence of a transmission completion interrupt enables the TXBn register to be efficiently written twice (2 bytes) without waiting for the transmission of 1 data frame.

When continuous transmission is performed, data should be written after referencing the ASIFn register to confirm the transmission status and whether or not data can be written to the TXBn register.

★ **Caution The values of the ASIF.TXBFn and ASIS.TXSFn bits change 10 → 11 → 01 in continuous transmission. Therefore, do not confirm the status based on the combination of the TXBFn and TXSFn bits.**

Read only the TXBFn bit during continuous transmission.

TXBFn	Whether or Not Writing to TXBn Register Is Enabled
0	Writing is enabled
1	Writing is not enabled

Caution When transmission is performed continuously, write the first transmit data (first byte) to the TXBn register and confirm that the TXBFn bit is 0, and then write the next transmit data (second byte) to TXBn register. If writing to the TXBn register is performed when the TXBFn bit is 1, transmit data cannot be guaranteed.

While transmission is being performed continuously, whether writing to the TXBn register later is enabled can be judged by confirming the TXSFn bit after the occurrence of a transmission completion interrupt.

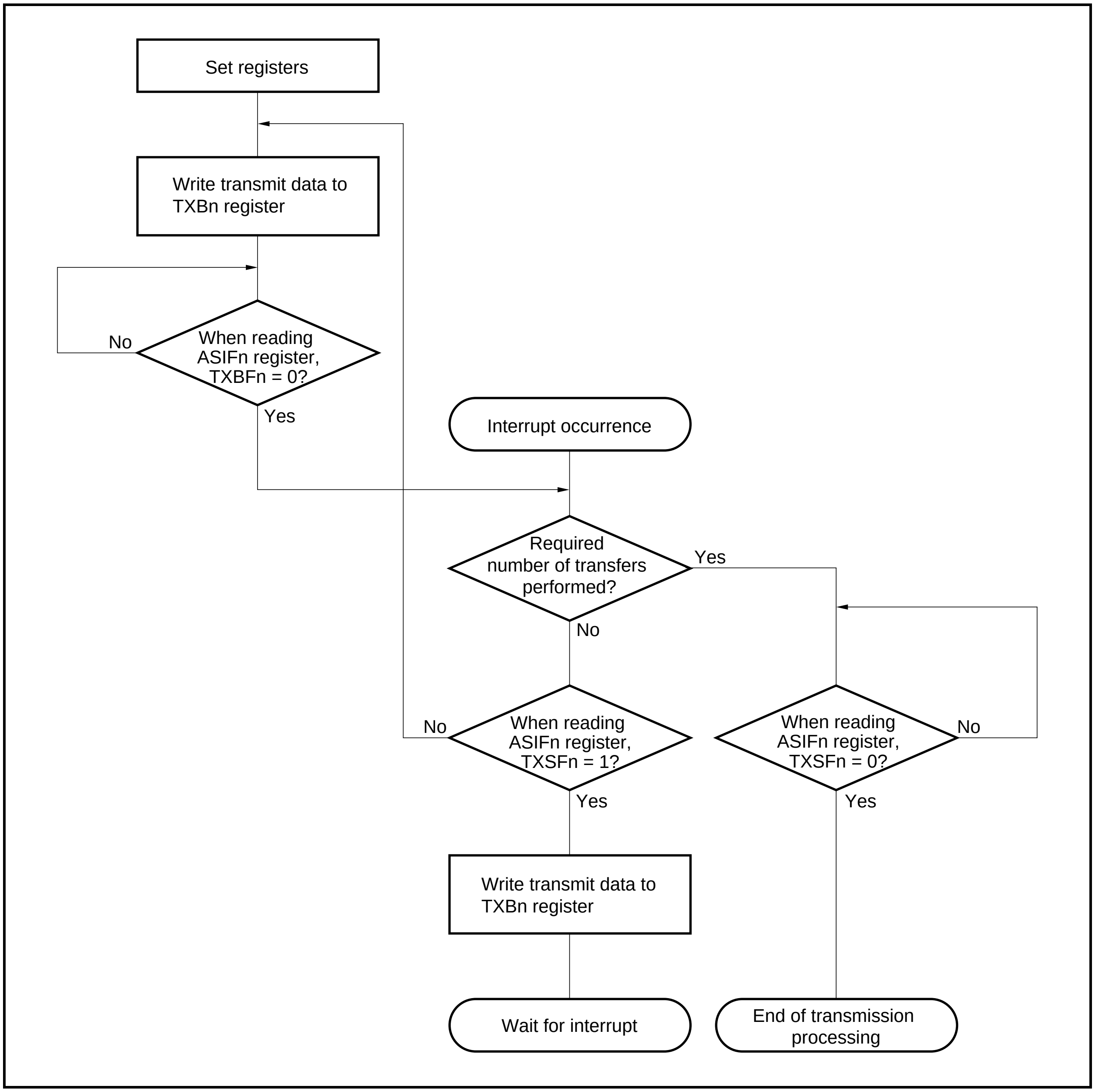
The communication status can be confirmed by referring to the TXSFn bit.

TXSFn	Transmission Status
0	Transmission is completed.
1	Under transmission.

Cautions 1. When initializing the transmission unit when continuous transmission is completed, confirm that the TXBFn bit is 0 after the occurrence of the transmission completion interrupt, and then execute initialization. If initialization is performed when the TXSFn bit is 1, transmit data cannot be guaranteed.

2. While transmission is being performed continuously, an overrun error may occur if the next transmission is completed before the INTSTn interrupt servicing following the transmission of 1 data frame is executed. An overrun error can be detected by embedding a program that can count the number of transmit data and referencing the TXSFn bit.

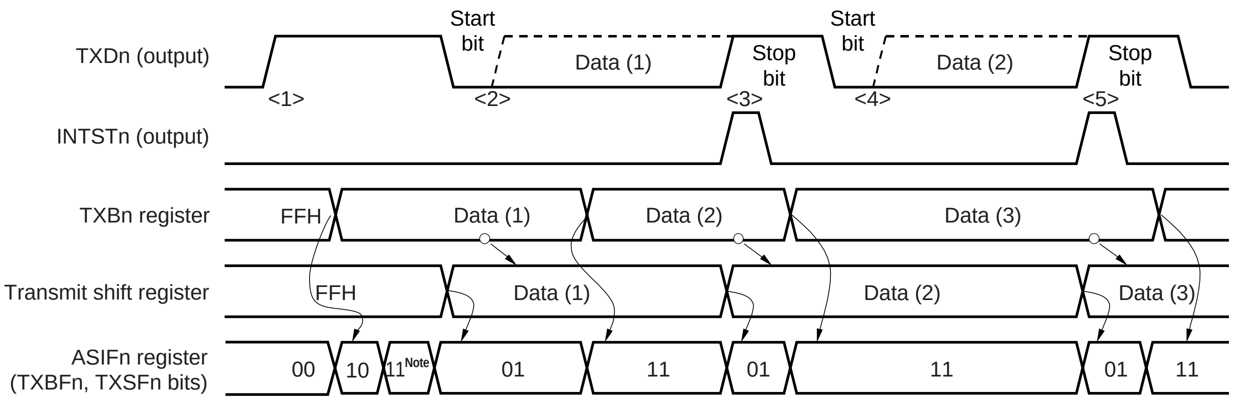
Figure 14-4. Continuous Transmission Processing Flow


Set registers
Write transmit data to TXBn register
When reading ASIFn register, TXBFn = 0?
No
Yes
Interrupt occurrence
Required number of transfers performed?
Yes
No
When reading ASIFn register, TXSFn = 1?
No
Yes
When reading ASIFn register, TXSFn = 0?
No
Yes
Write transmit data to TXBn register
Wait for interrupt
End of transmission processing

(a) Starting procedure

The procedure to start continuous transmission is shown below.

Figure 14-5. Continuous Transmission Starting Procedure (When Stop Bit Length Is 1)



Note Refer to **14.7 Cautions (2)**.

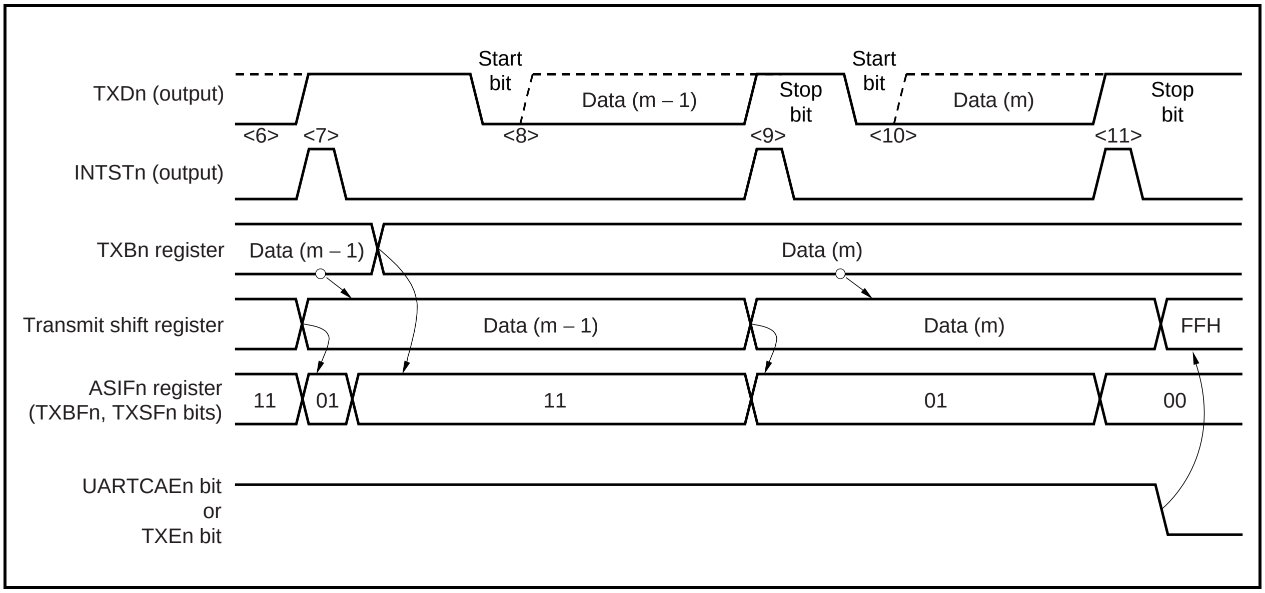
Transmission Starting Procedure	Internal Operation	ASIFn Register	
		TXBFn	TXSFn
• Set transmission mode	<1> Start transmission unit	0	0
• Write data (1)		1	0
	<2> Generate start bit	1	1[Note]
		0	1[Note]
• Read ASIFn register (confirm that TXBFn bit = 0)	Start data (1) transmission	0	1
		0	1
• Write data (2)		1	1
	<<Transmission in progress>>		
	<3> INTSTn interrupt occurs	0	1
• Read ASIFn register (confirm that TXBFn bit = 0)		0	1
• Write data (3)		1	1
	<4> Generate start bit		
	Start data (2) transmission		
	<<Transmission in progress>>		
	<5> INTSTn interrupt occurs	0	1
• Read ASIFn register (confirm that TXBFn bit = 0)		0	1
• Write data (4)		1	1

Note Refer to **14.7 Cautions (2)**.

(b) Ending procedure

The procedure for ending continuous transmission is shown below.

Figure 14-6. Continuous Transmission End Procedure (When Stop Bit Length Is 1)



Transmission End Procedure	Internal Operation	ASIFn Register	
		TXBFn	TXSFn
	<6> Transmission of data (m – 2) is in progress	1	1
	<7> INTSTn interrupt occurs →	0	1
• Read ASIFn register (confirm that TXBFn bit = 0) ←		0	1
• Write data (m) →		1	1
	<8> Generate start bit Start data (m – 1) transmission <<Transmission in progress>>		
	<9> INTSTn interrupt occurs →	0	1
• Read ASIFn register (confirm that TXSFn bit = 1) ← There is no write data		0	1
	<10> Generate start bit Start data (m) transmission <<Transmission in progress>>		
	<11> Generate INTSTn interrupt →	0	0
• Read ASIFn register (confirm that TXSFn bit = 0) ←		0	0
• Clear (0) the UARTCAEn bit or TXEn bit	Initialize internal circuits		

(4) Receive operation

The awaiting reception state is set by setting the ASIMn.UARTCAEn bit to 1 and then setting the RXEn bit to 1 in the ASIMn register. To start the receive operation, start sampling at the falling edge when the falling of the RXDn pin is detected. If the RXDn pin is low level at a start bit sampling point, the start bit is recognized. When the receive operation begins, serial data is stored sequentially in the receive shift register according to the baud rate that was set. A reception completion interrupt (INTSRn) is generated each time the reception of one frame of data is completed. Normally, the receive data is transferred from the RXBn register to memory by this interrupt servicing.

(a) Reception enabled state

The receive operation is set to the reception enabled state by setting the RXEn bit in the ASIMn register to 1.

- RXEn bit = 1: Reception enabled state
- RXEn bit = 0: Reception disabled state

In reception disabled state, the reception hardware stands by in the initial state. At this time, the contents of the RXBn register are retained, and no reception completion interrupt or reception error interrupt is generated.

(b) Starting a receive operation

A receive operation is started by the detection of a start bit.

The RXDn pin is sampled using the serial clock from baud rate generator n (BRGn).

(c) Reception completion interrupt

When the RXEn bit = 1 and the reception of one frame of data is completed (the stop bit is detected), a reception completion interrupt (INTSRn) is generated and the receive data within the receive shift register is transferred to the RXBn register at the same time.

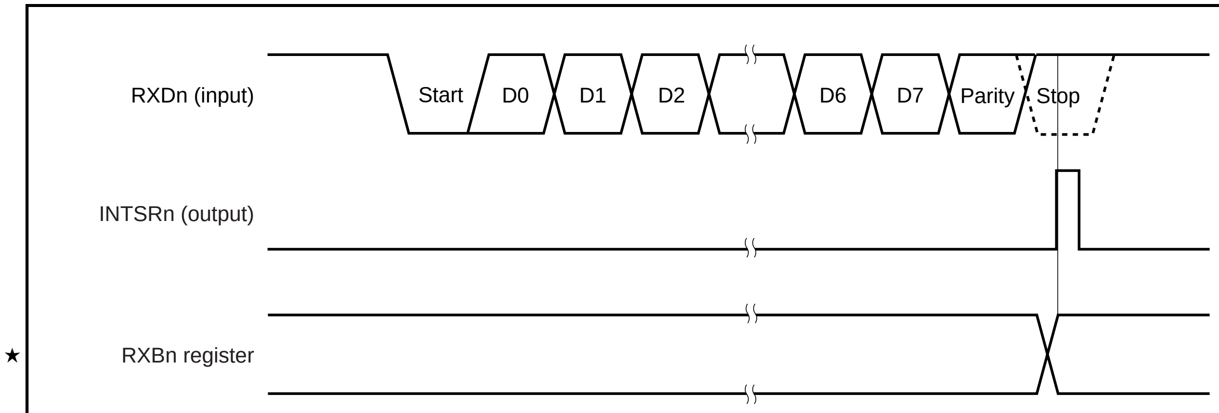
Also, if an overrun error (OVEn flag) occurs, the receive data at that time is not transferred to the RXBn register, and either a reception completion interrupt (INTSRn) or a reception error interrupt (INTSREn) is generated (the receive data within the receive shift register is transferred to RXBn) according to the ASIMn.ISRMn bit setting.

Even if a parity error (PEn flag) or framing error (FEn flag) occurs during a reception operation, the receive operation continues until stop bit is received, and after reception is completed, either a reception completion interrupt (INTSRn) or a reception error interrupt (INTSREn) is generated according to the ISRMn bit setting (the receive data within the receive shift register is transferred to the RXBn register).

If the RXEn bit is cleared (0) during a receive operation, the receive operation is immediately stopped. The contents of the RXBn register and of the ASISn register at this time do not change, and no reception completion interrupt (INTSRn) or reception error interrupt (INTSREn) is generated.

No reception completion interrupt is generated when RXEn = 0 (reception is disabled).

Figure 14-7. Asynchronous Serial Interface Reception Completion Interrupt Timing



Cautions 1. Be sure to read the RXBn register even when a reception error occurs. If the RXBn register is not read, an overrun error will occur at the next data reception and the reception error status will continue infinitely.

**2. Reception is always performed assuming a stop bit length of 1.
A second stop bit is ignored.**

(5) Reception error

The three types of errors that can occur during a receive operation are a parity error, framing error, and overrun error. As a result of data reception, the various flags of the ASISn register are set (1), and a reception error interrupt (INTSREn) or a reception completion interrupt (INTSRn) is generated at the same time. The ASIMn.ISRMn bit specifies whether INTSREn or INTSRn is generated.

The type of error that occurred during reception can be detected by reading the contents of the ASISn register during the INTSREn or INTSRn interrupt servicing.

The contents of the ASISn register are reset (0) by reading the ASISn register.

Table 14-3. Reception Error Causes

Error Flag	Reception Error	Cause
PEn	Parity error	The parity specification during transmission did not match the parity of the reception data
FEn	Framing error	No stop bit was detected
OVEn	Overrun error	The reception of the next data was completed before data was read from the RXBn register

(a) Separation of reception error interrupt

A reception error interrupt can be separated from the INTSRn interrupt and generated as the INTSREn interrupt by clearing the ASIMn.ISRMn bit to 0.

Figure 14-8. When Reception Error Interrupt Is Separated from Reception Completion Interrupt (INTSRn) (ISRMn Bit = 0)

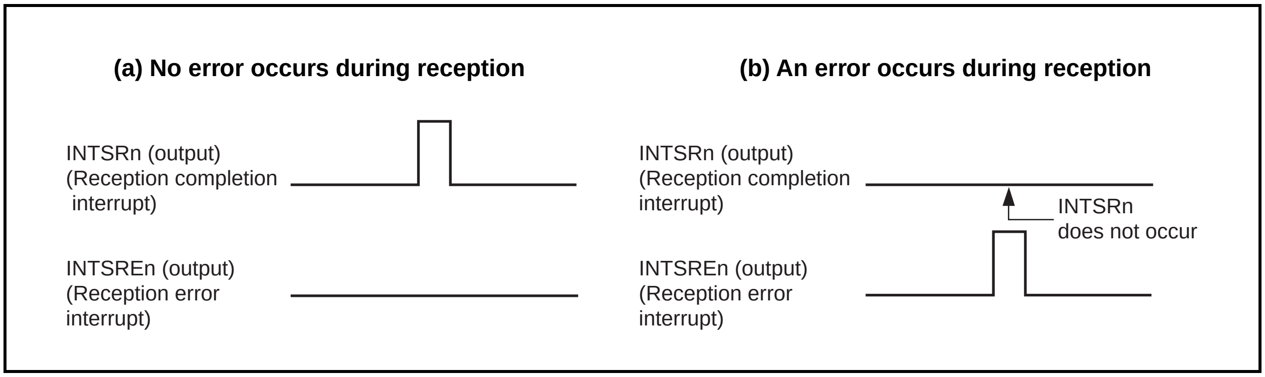
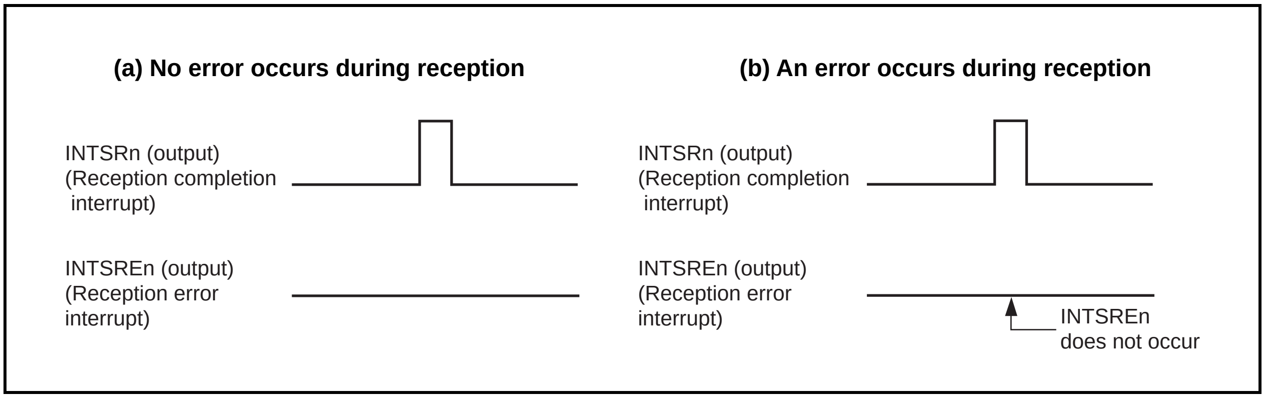

(a) No error occurs during reception
INTSRn (output)
(Reception completion interrupt)
INTSREn (output)
(Reception error interrupt)
(b) An error occurs during reception
INTSRn (output)
(Reception completion interrupt)
INTSRn does not occur
INTSREn (output)
(Reception error interrupt)

Figure 14-9. When Reception Error Interrupt Is Included in Reception Completion Interrupt (INTSRn) (ISRMn Bit = 1)


(a) No error occurs during reception
INTSRn (output)
(Reception completion interrupt)
INTSREn (output)
(Reception error interrupt)
(b) An error occurs during reception
INTSRn (output)
(Reception completion interrupt)
INTSREn (output)
(Reception error interrupt)
INTSREn does not occur

(6) Parity types and corresponding operation

A parity bit is used to detect a bit error in communication data. Normally, the same type of parity bit is used on the transmission and reception sides.

(a) Even parity

(i) During transmission

The parity bit is controlled so that the number of bits with the value “1” within the transmit data including the parity bit is even. The parity bit value is as follows.

- If the number of bits with the value “1” within the transmit data is odd: 1
- If the number of bits with the value “1” within the transmit data is even: 0

(ii) During reception

The number of bits with the value “1” within the receive data including the parity bit is counted, and a parity error is generated if this number is odd.

(b) Odd parity

(i) During transmission

In contrast to even parity, the parity bit is controlled so that the number of bits with the value “1” within the transmit data including the parity bit is odd. The parity bit value is as follows.

- If the number of bits with the value “1” within the transmit data is odd: 0
- If the number of bits with the value “1” within the transmit data is even: 1

(ii) During reception

The number of bits with the value “1” within the receive data including the parity bit is counted, and a parity error is generated if this number is even.

(c) 0 parity

During transmission the parity bit is set to “0” regardless of the transmit data.

During reception, no parity bit check is performed. Therefore, no parity error is generated regardless of whether the parity bit is “0” or “1”.

(d) No parity

No parity bit is added to the transmit data.

During reception, the receive operation is performed as if there were no parity bit. Since there is no parity bit, no parity error is generated.

(7) Receive data noise filter

The RXDn signal is sampled at the rising edge of the prescaler output base clock (Clock). If the same sampling value is obtained twice, the match detector output changes, and this output is sampled as input data. Therefore, data not exceeding one clock width is judged to be noise and is not delivered to the internal circuit (see **Figure 14-11**). Refer to **14.6 (1) (a) Base clock (Clock)** regarding the base clock.

Also, since the circuit is configured as shown in Figure 14-10, internal processing during a receive operation is delayed by up to 2 clocks according to the external signal status.

Figure 14-10. Noise Filter Circuit

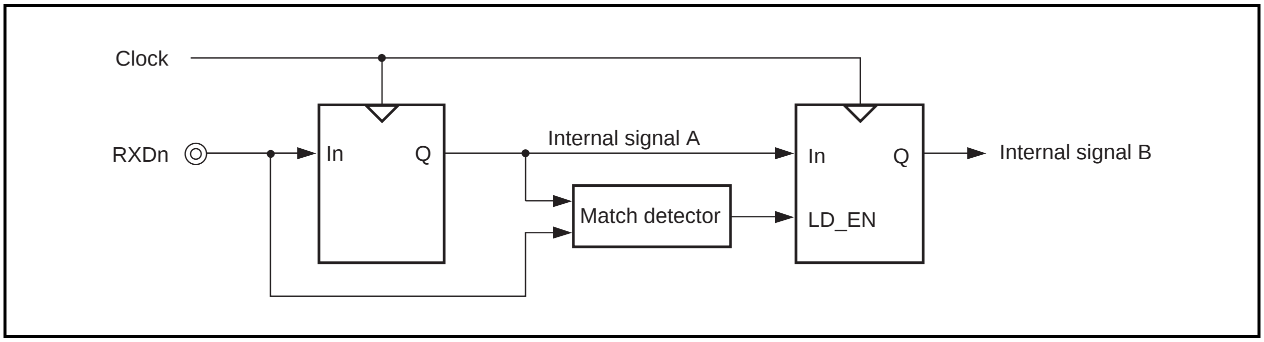
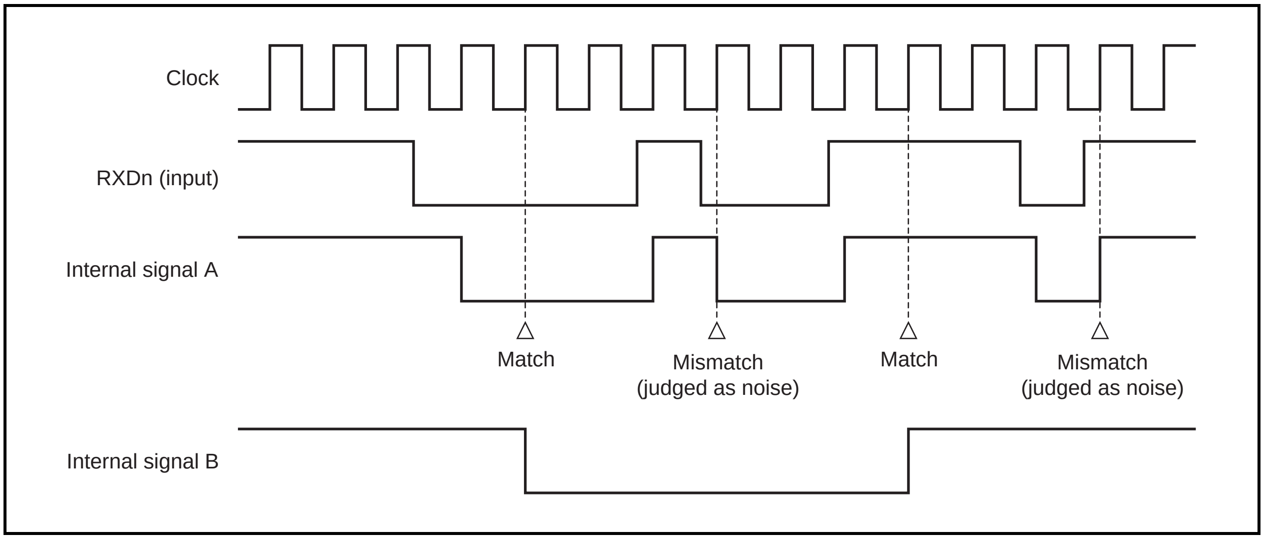


Figure 14-11. Timing of RXDn Signal Judged as Noise



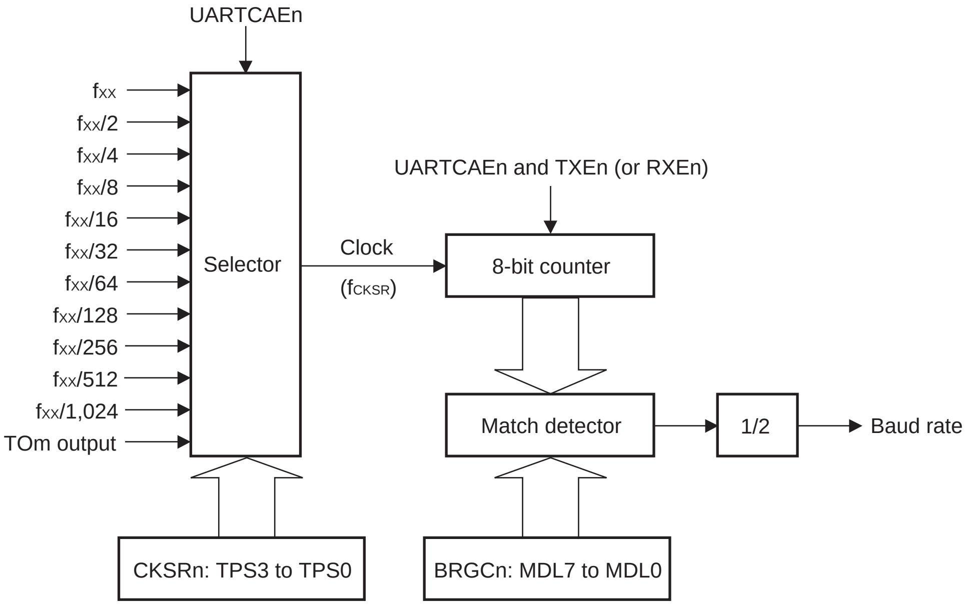
14.6 Dedicated Baud Rate Generator n (BRGn)

A dedicated baud rate generator, which consists of a source clock selector and an 8-bit programmable counter, generates serial clocks during transmission/reception by UARTn. The dedicated baud rate generator output can be selected as the serial clock for each channel.

Separate 8-bit counters exist for transmission and for reception.

(1) Baud rate generator n (BRGn) configuration

Figure 14-12. Configuration of Baud Rate Generator n (BRGn)



Remark f_{XX}: Main clock frequency
m = 20 when n = 0
m = 21 when n = 1

(a) Base clock (clock)

When the ASIMn.UARTCAEn bit = 1 the clock selected according to the CKSRn.TPS3 to CKSRn.TPS0 bits is supplied to the transmission/reception unit. This clock is called the base clock (Clock), and its frequency is referred to as f_{CKSR}. When UARTCAEn = 0, Clock is fixed to low level.

(2) Serial clock generation

A serial clock can be generated according to the settings of the CKSRn and BRGCn registers.
The base clock to the 8-bit counter is selected by the CKSRn.TPS3 to CKSRn.TPS0 bits.
The 8-bit counter divisor value can be set by the BRGCn.MDL7 to BRGCn.MDL0 bits.

(a) Clock select register n (CKSRn)

CKSRn is an 8-bit register for selecting the base clock using the TPS3 to TPS0 bits. The clock selected by the TPS3 to TPS0 bits becomes the base clock (Clock) of the transmission/reception module. Its frequency is referred to as f_{CKSR}.
This register can be read or written in 8-bit units.
After reset, this register is set to 00H.

Caution Set the ASIMn.UARTCAEn bit to 0 before rewriting the TPS3 to TPS0 bits.

After reset: 00H R/W Address: CKSR0 FFFFFA06H, CKSR1 FFFFFA16H

	7	6	5	4	3	2	1	0
CKSRn	0	0	0	0	TPSn3	TPSn2	TPSn1	TPSn0

TPSn3	TPSn2	TPSn1	TPSn0	Receive operation
0	0	0	0	f_{XX}[Note]
0	0	0	1	$f_{XX}/2$
0	0	1	0	$f_{XX}/4$
0	0	1	1	$f_{XX}/8$
0	1	0	0	$f_{XX}/16$
0	1	0	1	$f_{XX}/32$
0	1	1	0	$f_{XX}/64$
0	1	1	1	$f_{XX}/128$
1	0	0	0	$f_{XX}/256$
1	0	0	1	$f_{XX}/512$
1	0	1	0	$f_{XX}/1,024$
1	0	1	1	TOm output
Other than above				Setting prohibited

Remark m = 20 when n = 0
m = 21 when n = 1

Note Setting the TPSn3 to TPSn0 bits to 0000B is prohibited when $V_{DD} < 3.0$ V and $f_{XX} > 10$ MHz.

(b) Baud rate generator control register n (BRGCn)

BRGCn is an 8-bit register that controls the baud rate (serial transfer speed) of UARTn.

This register can be read or written in 8-bit units.

After reset, this register is set to FFH.

Caution If the MDLn7 to MDLn0 bits are to be overwritten, the ASIMn.TXEn and ASIMn.RXEn bits should be set to 0 first.

After reset: FFH R/W Address: BRGC0 FFFFFA07H, BRGC1 FFFFFA17H

	7	6	5	4	3	2	1	0
BRGCn	MDLn7	MDLn6	MDLn5	MDLn4	MDLn3	MDLn2	MDLn1	MDLn0

MD Ln7	MD Ln6	MD Ln5	MD Ln4	MD Ln3	MD Ln2	MD Ln1	MD Ln0	Setting value (k)	Serial clock
0	0	0	0	0	×	×	×	–	Setting prohibited
0	0	0	0	1	0	0	0	8	f_{CKSR}/8
0	0	0	0	1	0	0	1	9	f_{CKSR}/9
0	0	0	0	1	0	1	0	10	f_{CKSR}/10
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	0	1	0	250	f_{CKSR}/250
1	1	1	1	1	0	1	1	251	f_{CKSR}/251
1	1	1	1	1	1	0	0	252	f_{CKSR}/252
1	1	1	1	1	1	0	1	253	f_{CKSR}/253
1	1	1	1	1	1	1	0	254	f_{CKSR}/254
1	1	1	1	1	1	1	1	255	f_{CKSR}/255

Remarks 1. f_{CKSR}: Frequency [Hz] of base clock (Clock) selected by CKSRn.TPSn3 to CKSRn.TPSn0 bits.

2. k: Value set by MDLn7 to MDLn0 bits (k = 8, 9, 10, ..., 255)

3. The baud rate is the output clock for the 8-bit counter divided by 2

4. ×: Don't care

(c) Baud rate

The baud rate is the value obtained by the following formula.

$$\text{Baud rate} = \frac{f_{CKSR}}{2 \times k}\ [\text{bps}]$$

f_{CKSR} = Frequency [Hz] of base clock (Clock) selected by CKSRn.TPSn3 to CKSRn.TPSn0 bits.
k = Value set by BRGCn.MDLn7 to BRGCn.MDLn0 bits. (k = 8, 9, 10, ..., 255)

(d) Baud rate error

The baud rate error is obtained by the following formula.

$$\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Target baud rate (normal baud rate)}} - 1 \right) \times 100[\%]$$

Cautions 1. Make sure that the baud rate error during transmission does not exceed the allowable error of the reception destination.

2. Make sure that the baud rate error during reception is within the allowable baud rate range described in (4) Allowable baud rate range during reception.

Example: Base clock (Clock) frequency = 20 MHz = 20,000,000 Hz
Setting of BRGCn.MDLn7 to BRGCn.MDLn0 bits = 01000001B (k = 65)
Target baud rate = 153,600 bps

Baud rate = 20M/(2 × 65)
= 20,000,000/(2 × 65) = 153,846 [bps]

Error = (153,846/153,600 − 1) × 100
= 0.160 [%]

(3) Baud rate setting example

Table 14-3. Baud Rate Generator Setting Data

Baud Rate [bps]	f_{XX} = 20 MHz			f_{XX} = 10 MHz		
	f_{CKSR}	k	ERR	f_{CKSR}	k	ERR
300	$f_{XX}/512$	65	0.16	$f_{XX}/256$	65	0.16
600	$f_{XX}/256$	65	0.16	$f_{XX}/128$	65	0.16
1,200	$f_{XX}/128$	65	0.16	$f_{XX}/64$	65	0.16
2,400	$f_{XX}/64$	65	0.16	$f_{XX}/32$	65	0.16
4,800	$f_{XX}/32$	65	0.16	$f_{XX}/16$	65	0.16
9,600	$f_{XX}/16$	65	0.16	$f_{XX}/8$	65	0.16
19,200	$f_{XX}/8$	65	0.16	$f_{XX}/4$	65	0.16
31,250	$f_{XX}/32$	10	0.00	$f_{XX}/16$	10	0.00
38,400	$f_{XX}/4$	65	0.16	$f_{XX}/2$	65	0.16
76,800	$f_{XX}/2$	65	0.16	f_{XX}	65	0.16
153,600	f_{XX}	65	0.16	f_{XX}	33	–1.36
312,500	$f_{XX}/4$	8	0.00	$f_{XX}/2$	8	0.00

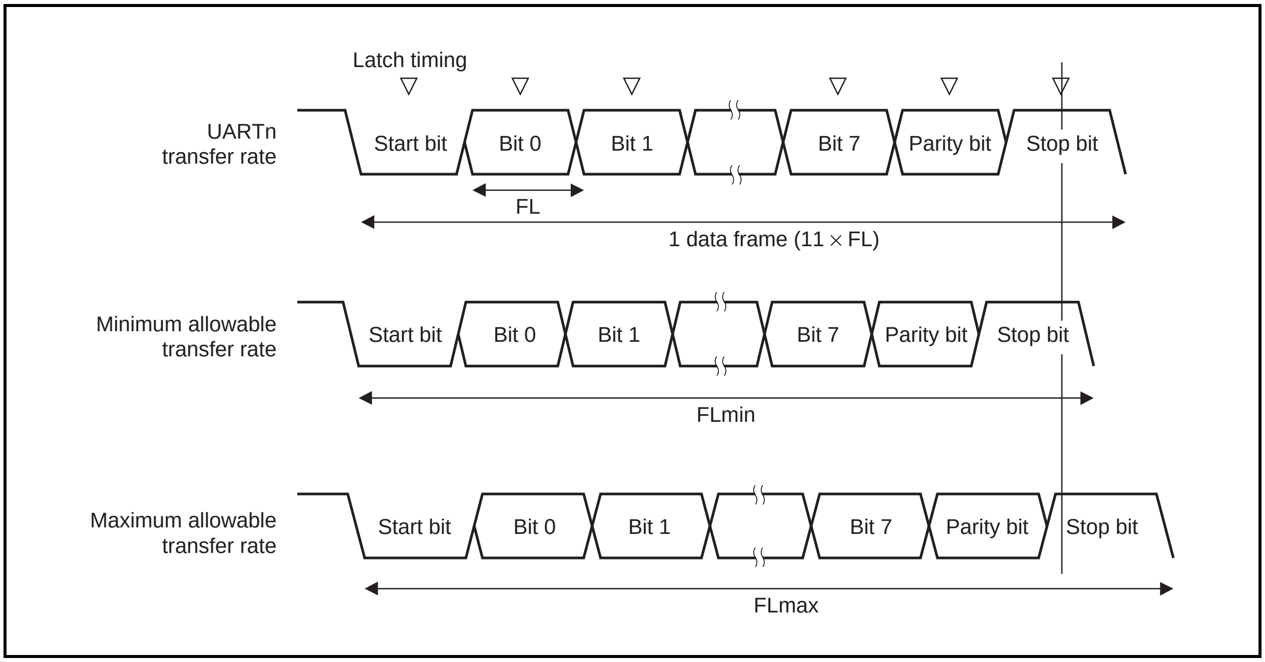
Remark f_{XX}: Main clock frequency
f_{CKSR}: Base clock frequency
k: Setting values of BRGCn.MDLn7 to BRGCn.MDLn0 bits
ERR: Baud rate error [%]

(4) Allowable baud rate range during reception

The degree to which a discrepancy from the transmission destination’s baud rate is allowed during reception is shown below.

Caution The equations described below should be used to set the baud rate error during reception so that it always is within the allowable error range.

Figure 14-13. Allowable Baud Rate Range During Reception



As shown in Figure 14-13, after the start bit is detected, the receive data latch timing is determined according to the counter that was set by the BRGCn register. If all data up to the final data (stop bit) is in time for this latch timing, the data can be received normally.

If this is applied to 11-bit reception, the following is theoretically true.

$$FL = (Brate)^{-1}$$

Brate:	UARTn baud rate
k:	BRGCn register setting value
FL:	1-bit data length

When the latch timing margin is 2 base clocks (Clock), the minimum allowable transfer rate (FLmin) is as follows.

$$FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

Therefore, the transfer destination’s maximum receivable baud rate (BRmax) is as follows.

$$\text{BRmax} = (\text{FLmin}/11)^{-1} = \frac{22k}{21k + 2}\ \text{Brate}$$

Similarly, the maximum allowable transfer rate (FLmax) can be obtained as follows.

$$\frac{10}{11} \times \text{FLmax} = 11 \times \text{FL} - \frac{k + 2}{2 \times k} \times \text{FL} = \frac{21k - 2}{2 \times k}\ \text{FL}$$

$$\text{FLmax} = \frac{21k - 2}{20k}\ \text{FL} \times 11$$

Therefore, the transfer destination’s minimum receivable baud rate (BRmin) is as follows.

$$\text{BRmin} = (\text{FLmax}/11)^{-1} = \frac{20k}{21k - 2}\ \text{Brate}$$

The allowable baud rate error of UARTn and the transfer destination can be obtained as follows from the expressions described above for computing the minimum and maximum baud rate values.

Table 14-4. Maximum and Minimum Allowable Baud Rate Error

Division Ratio (k)	Maximum Allowable Baud Rate Error	Minimum Allowable 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%

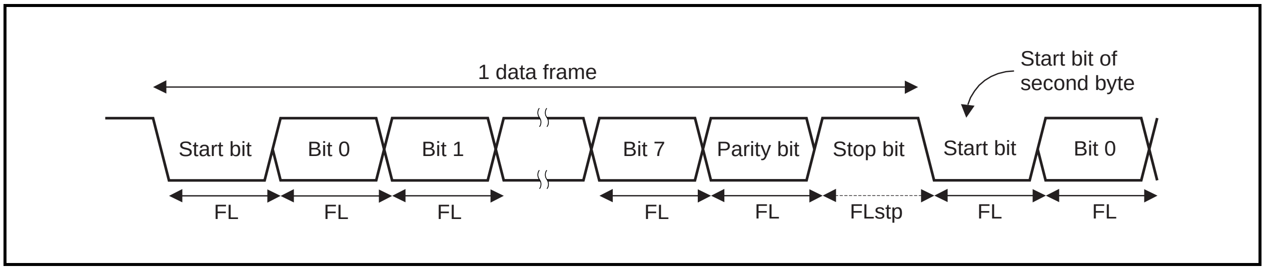
Remarks **1.** The reception precision depends on the number of bits in one frame, the base clock frequency, and the division ratio (k). The higher the base clock frequency and the larger the division ratio (k), the higher the precision.

2. k: BRGCn register setting value

(5) Transfer rate during continuous transmission

During continuous transmission, the transfer rate from a stop bit to the next start bit is extended two clocks of the base clock (Clock) longer than normal. However, on the reception side, the transfer result is not affected since the timing is initialized by the detection of the start bit.

Figure 14-14. Transfer Rate During Continuous Transmission



Representing the 1-bit data length by FL, the stop bit length by FLstp, and the base clock frequency by f_{CKSR} yields the following equation.

$FLstp = FL + 2/f_{CKSR}$

Therefore, the transfer rate during continuous transmission is as follows (when stop bit length = 1).

$\text{Transfer rate} = 11 \times FL = 2/f_{CKSR}$

14.7 Cautions

Cautions to be observed when using UARTn are shown below.

(1) When the supply of clocks to UARTn is stopped (for example, in IDLE or STOP mode), operation stops with each register retaining the value it had immediately before the supply of clocks was stopped. The TXDn pin output also holds and outputs the value it had immediately before the supply of clocks was stopped. However, operation is not guaranteed after the supply of clocks is restarted. Therefore, after the supply of clocks is restarted, the circuits should be initialized by setting the ASIMn.UARTCAEn, ASIMn.RXEn, and ASIMn.TXEn bits = 000 in register.

(2) UARTn has a 2-stage buffer configuration consisting of the TXBn register and the transmit shift register, and has status flags (the ASIFn.TXBFn and ASIFn.TXSFn bits) that indicate the status of each buffer. If the TXBFn and TXSFn bits are read in continuous transmission, the value changes from 10 → 11 → 01. Read only the TXBFn bit during continuous transmission.

CHAPTER 15 CLOCKED SERIAL INTERFACE n (CSIn)

15.1 Features

- Transfer rate: Master mode: Maximum 5 Mbps (when internal system clock operates at 20 MHz)
  Slave mode: Maximum 5 Mbps
- Half-duplex communications
- Master mode and slave mode can be selected
- Transmission data length: 8 bits
- Transfer data direction can be switched between MSB first and LSB first
- Seven clock signals can be selected (6 master clocks and 1 slave clock)
- 3-wire method
  - SOn: Serial data output
  - SIn: Serial data input
  - $\overline{\text{SCKn}}$: Serial clock input/output
- Interrupt sources: 1 type
  - Transmission/reception completion interrupt (INTCSIn)
- Transmission/reception mode or reception-only mode can be specified
- On-chip transmit buffer (SOTBn)

Remark n = 0, 1

15.2 Configuration

CSIn is controlled by the clocked serial interface mode register (CSIMn). Transmit/receive data can be written to or read from the SIOn register.

(1) Clocked serial interface mode register n (CSIMn)
CSIMn is an 8-bit register for specifying the operation of CSIn.

(2) Clocked serial interface clock select register n (CSICn)
CSICn is an 8-bit register for controlling the transmit operation of CSIn.

(3) Serial I/O shift register n (SIOn)
SIOn is an 8-bit register for converting between serial data and parallel data. The SIOn register is used for both transmission and reception.
Data is shifted in (reception) or shifted out (transmission) beginning at either the MSB side or the LSB side.
Actual transmit/receive operations are controlled by reading or writing the SIOn register.

(4) Clocked serial interface transmit buffer register n (SOTBn)
SOTBn is an 8-bit buffer register for storing transmit data.

(5) Selector
The selector selects the serial clock to be used.

(6) Serial clock controller

The serial clock controller controls the supply of serial clocks to the shift register. When an internal clock is used, it also controls the clocks that are output to the $\overline{\text{SCKn}}$ pin.

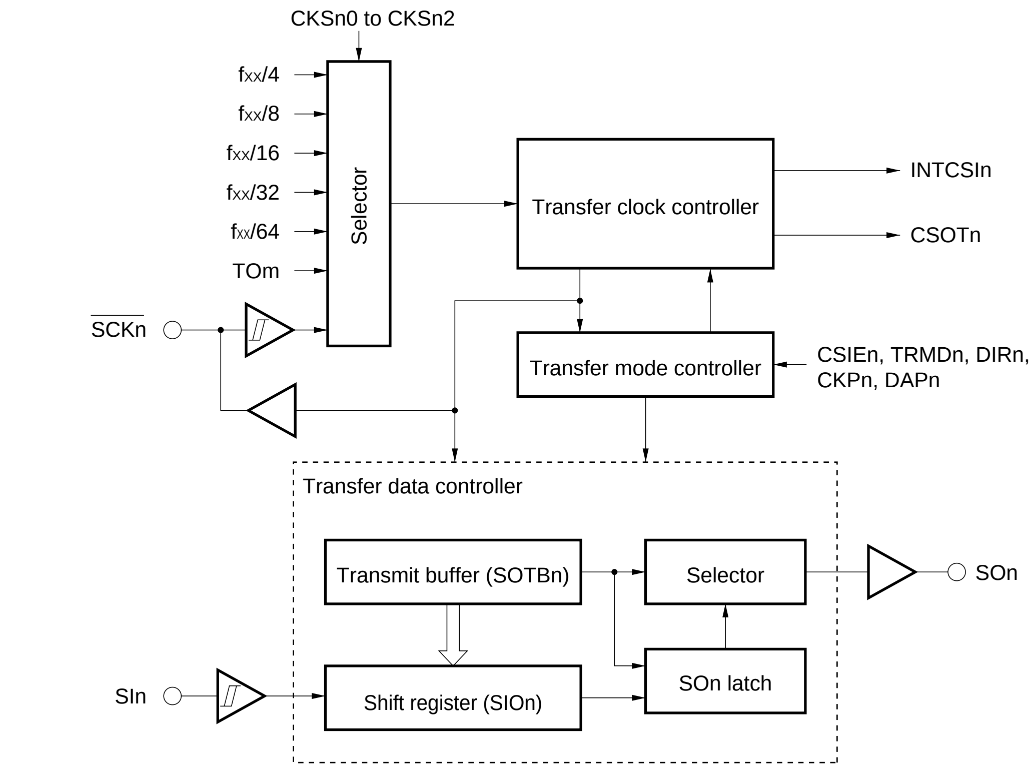
(7) Serial clock counter

The serial clock counter counts serial clocks that are output or input during transmit and receive operations and checks that 8-bit data has been transmitted or received.

(8) Interrupt controller

The interrupt controller controls whether or not an interrupt request is generated when the serial clock counter has counted eight serial clocks.

Figure 15-1. Clocked Serial Interface Block Diagram



Remarks 1. m = 20 when n = 0

m = 21 when n = 1

2. fXX: Main clock frequency

15.3 Registers

(1) Clocked serial interface mode register n (CSIMn)

CSIMn is a register that controls the operation of CSIn.

This register can be read or written in 8-bit or 1-bit units.

After reset, this register is set to 00H.

Caution To use CSIn, be sure to set the external pins related to the CSIn function to control mode and set the CSICn register. Then set the CSIEn bit to 1 before setting the other bits.

Remark n = 0, 1

After reset: 00H R/W Address: CSIM0 FFFFFD00H, CSIM1 FFFFFD10H

	<7>	<6>	5	<4>	3	2	1	<0>
CSIMn	CSIEn	TRMDn	0	DIRn	0	0	0	CSOTn

CSIEn	CSIn operation enable/disable specification
0	CSIn operation is disabled (SOn = low level, $\overline{\text{SCKn}}$ = high level)
1	CSIn operation is enabled

- If CSIEn is set to 0, the CSIn unit can be reset[Note] asynchronously.
- If CSIEn = 0, the CSIn unit is in a reset state. Therefore, to operate CSIn, CSIEn must be set to 1.
- If the CSIEn bit is changed from 1 to 0, all registers of the CSIn unit are initialized. To set CSIEn to 1 again, the registers of the CSIn unit must be set again.

TRMDn	Transmission mode specification
0	Reception-only mode
1	Transmission/reception mode

- If TRMDn = 0, reception mode is selected. In addition, SOn outputs a low level. Data reception is started by reading the SIOn register.
  If TRMDn = 1, transmission/reception is started by writing data to the SOTBn register.
- The TRMDn bit can be overwritten only when CSOTn = 0.

DIRn	Transfer direction mode (MSB/LSB specification)
0	MSB first
1	LSB first

- The DIRn bit can be overwritten only when CSOTn = 0.

CSOTn	Communication status flag
0	Communication stopped
1	Communication in progress

- This flag is used to judge whether writing to the SIOn register is enabled or not when starting serial data transmission in transmission/reception mode (TRMDn = 1)
- The CSOTn bit is cleared when the CSIE bit is cleared (0).

Note The following registers and bit can be reset.
SIOn and SIDEn registers
CSIMn.CSOTn bit

Caution Be sure to clear bits 5 and 3 to 1 to 0.

Remark n = 0, 1

(2) Clocked serial interface clock select register n (CSICn)

CSICn is an 8-bit register that controls the transmit operation of CSIn.

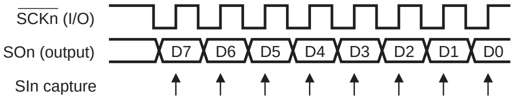
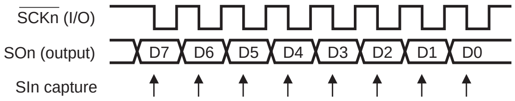
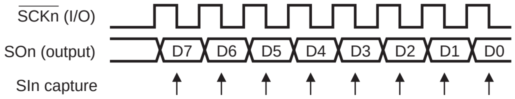
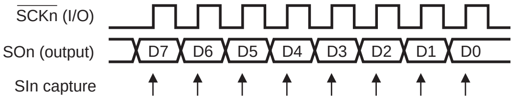
This register can be read or written in 8-bit units.

After reset, this register is set to 00H.

Caution The CSICn register can only be overwritten after CSIMn.CSIEn is cleared to 0.

After reset: 00H R/W Address: CSIC0 FFFFFD01H, CSIC1 FFFFFD11H

	7	6	5	4	3	2	1	0
CSICn	0	0	0	CKPn	DAPn	CKSn2	CKSn1	CKSn0

CKPn	DAPn	Specification of data transmission/reception timing for $\overline{\text{SCKn}}$
0	0	 SCKn (I/O) SOn (output) SIn capture D7 D6 D5 D4 D3 D2 D1 D0
0	1	 SCKn (I/O) SOn (output) SIn capture D7 D6 D5 D4 D3 D2 D1 D0
1	0	 SCKn (I/O) SOn (output) SIn capture D7 D6 D5 D4 D3 D2 D1 D0
1	1	 SCKn (I/O) SOn (output) SIn capture D7 D6 D5 D4 D3 D2 D1 D0

CKSn2	CKSn1	CKSn0	Serial clock	Mode
0	0	0	Setting prohibited	
0	0	1	f_{XX}/4[Note 1]	Master mode
0	1	0	f_{XX}/8	Master mode
0	1	1	f_{XX}/16	Master mode
1	0	0	f_{XX}/32	Master mode
1	0	1	f_{XX}/64	Master mode
1	1	0	TOm output[Note 2]	Master mode
1	1	1	External clock ($\overline{\text{SCKn}}$)	Slave mode

Notes 1. Setting is prohibited when V_{DD} < 3.0 V and f_{XX} > 10 MHz.

2. m = 20 when n = 0

m = 21 when n = 1

(a) Transfer rate selection example

CKSn2	CKSn1	CKSn0	Transfer Rate (bps)	
			20 MHz Operation	10 MHz Operation
0	0	0	Setting prohibited	Setting prohibited
0	0	1	5,000,000[Note]	2,500,000
0	1	0	2,500,000	1,250,000
0	1	1	1,250,000	625,000
1	0	0	625,000	312,500
1	0	1	312,500	156,250

Note Setting is prohibited when $2.7\ \text{V} \le V_{DD} < 3.0\ \text{V}$.

(3) Serial I/O shift register n (SIOn)

SIOn is an 8-bit shift register that converts parallel data to serial data. If CSIMn.TRMDn = 0, the receive operation is started by reading the SIOn register.

Except after reset, the SIOn register becomes 00H even when the CSIMn.CSIEn bit is cleared (0).

SIOn shifts data in (reception) or shifts data out (transmission) beginning at the MSB or the LSB side.

This register is read-only, in 8-bit units.

Caution The SIOn register can be accessed only when the system is in an idle state (CSIMn.CSOTn bit = 0).

After reset: 00H R Address: SIO0 FFFFFD02H, SIO1 FFFFFD12H

	7	6	5	4	3	2	1	0
SIOn	SIOn7	SIOn6	SIOn5	SIOn4	SIOn3	SIOn2	SIOn1	SIOn0

Remark n = 0, 1

(4) Receive-only serial I/O shift register n (SIOEn)

SIOEn is an 8-bit shift register that converts parallel data into serial data. A receive operation does not start even if the SIOEn register is read while the CSIMn.TRMDn bit is 0. Therefore this register is used to read the value of the SIOn register (receive data) without starting a receive operation.

SIOEn shifts data in (reception) beginning at the MSB or the LSB side.

Except after reset, the SIOEn register becomes 00H even when the CSIMn.CSIEn bit is cleared (0).

This register is read-only, in 8-bit units.

Caution The SIOEn register can be accessed only when the system is in an idle state (CSIMn.CSOTn bit = 0).

After reset: 00H R Address: SIOE0 FFFFFD03H, SIOE1 FFFFFD13H

	7	6	5	4	3	2	1	0
SIOEn	SIOEn7	SIOEn6	SIOEn5	SIOEn4	SIOEn3	SIOEn2	SIOEn1	SIOEn0

Remark n = 0, 1

(5) Clocked serial interface transmit buffer register n (SOTBn)

SOTBn is an 8-bit buffer register for storing transmit data.

If transmission/reception mode is set (CSIMn.TRMDn = 1), a transmit operation is started by writing data to the SOTBn register.

This register can be read or written in 8-bit units.

After reset, this register is set to 00H.

Caution The SOTBn register can be accessed only when the system is in an idle state (CSIMn.CSOTn bit = 0).

After reset: 00H R/W Address: SOTB0 FFFFFD04H, SOTB1 FFFFFD14H

	7	6	5	4	3	2	1	0
SOTBn	SOTBn7	SOTBn6	SOTBn5	SOTBn4	SOTBn3	SOTBn2	SOTBn1	SOTBn0

Remark n = 0, 1

15.4 Operation

(1) Transfer mode

CSIn transmits and receives data using three lines: 1 clock line and 2 data lines.

In reception-only mode (CSIMn.TRMDn = 0), the communication is started by reading the SIOn register. To read the value of the SIOn register without starting reception, read the SIOEn register.

In transmission/reception mode (CSIMn.TRMDn = 1), the communication is started by writing data to the SOTBn register.

When an 8-bit transfer of CSIn ends, the CSIMn.CSOTn bit becomes 0, and transfer stops automatically. Also, when the transfer ends, a transmission/reception completion interrupt (INTCSIn) is generated.

Cautions 1. When CSIMn.CSOTn bit = 1, the control registers and data registers should not be accessed.

2. If transmit data is written to the SOTBn register and the CSIMn.TRMDn bit is changed from 0 to 1, serial transfer is not performed.

Remark n = 0, 1

(2) Serial clock

(a) When internal clock is selected as the serial clock

If reception or transmission is started, a serial clock is output from the $\overline{\text{SCKn}}$ pin, and the data of the SIn pin is taken into the SIOn register sequentially or data is output to the SOn pin sequentially from the SIOn register when the data is synchronized with the serial clock in accordance with the setting of the CSICn.CKPn and CSICn.DAPn bits.

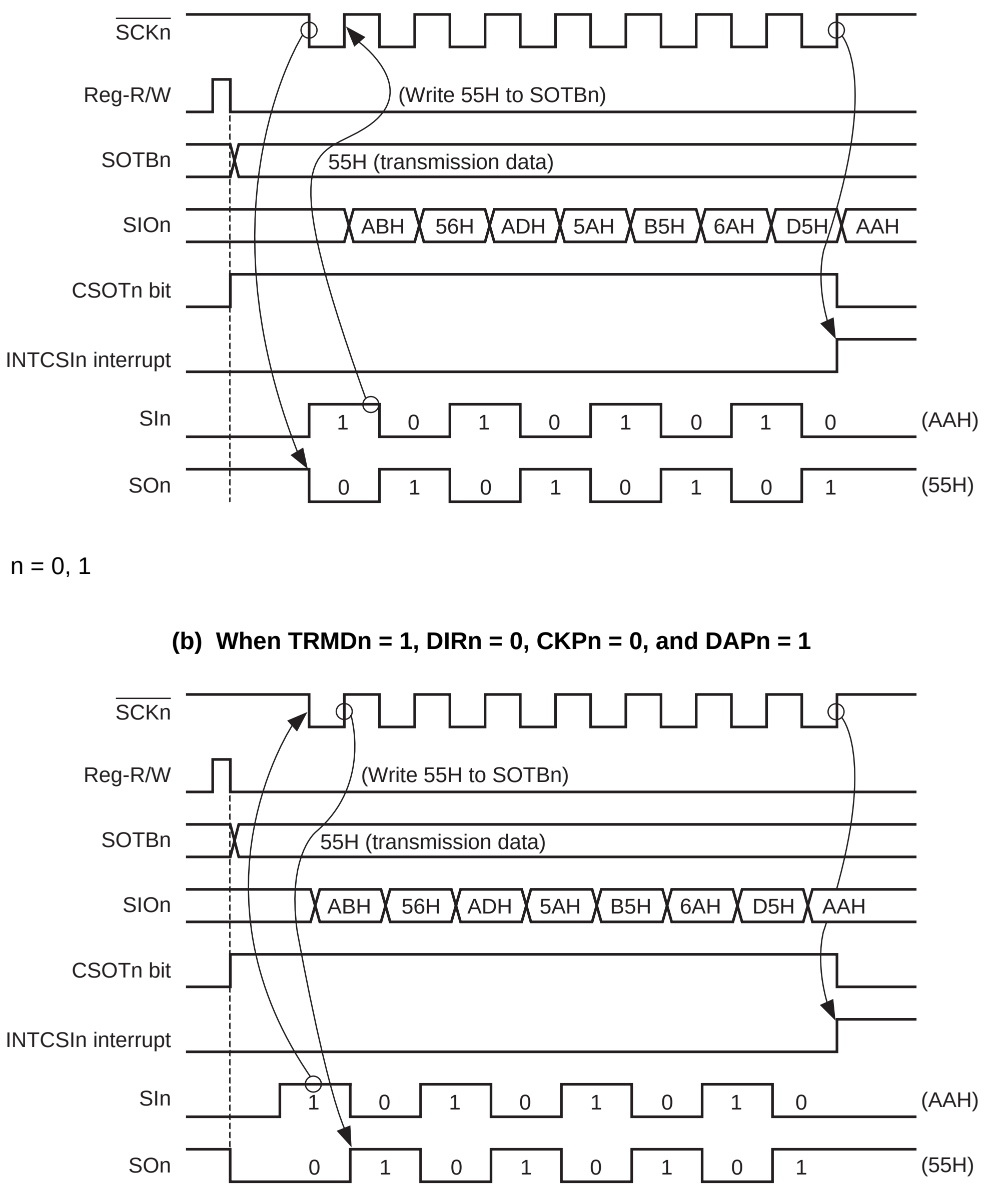
(b) When external clock is selected as the serial clock

If reception or transmission is started, the data of the SIn pin is taken into the SIOn register sequentially or output to the SOn pin sequentially in synchronization with the serial clock that has been input to the $\overline{\text{SCKn}}$ pin following transmission/reception startup in accordance with the setting of the CSICn.CKPn and CSICn.DAPn bits.

If serial clock is input to the $\overline{\text{SCKn}}$ pin when neither reception nor transmission is started, a shift operation will not be executed.

Remark n = 0, 1

Figure 15-2. Transfer Timing



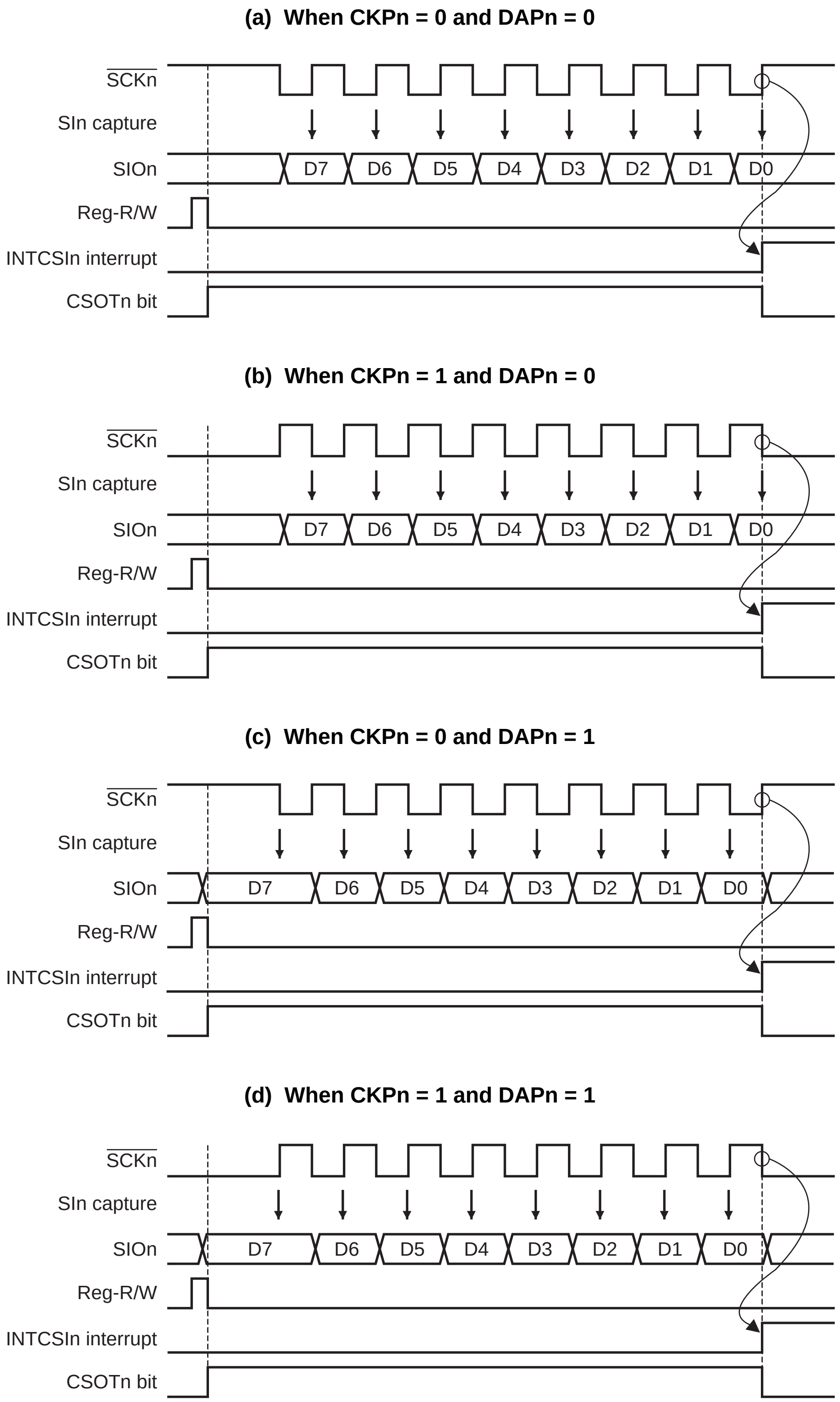
(a) When TRMDn = 1, DIRn = 0, CKPn = 0, and DAPn = 0

Remark n = 0, 1

(b) When TRMDn = 1, DIRn = 0, CKPn = 0, and DAPn = 1

Remark n = 0, 1

Figure 15-3. Clock Timing



(a) When CKPn = 0 and DAPn = 0

(b) When CKPn = 1 and DAPn = 0

(c) When CKPn = 0 and DAPn = 1

(d) When CKPn = 1 and DAPn = 1

Remark n = 0, 1

15.5 Output Pins

The output pins are described below. For the setting of each pin, refer to **Table 4-15 Settings When Port Pins Are Used for Alternate Functions**.

(1) $\overline{\text{SCKn}}$ pin

When CSIn operation is disabled (CSIMn.CSIEn bit = 0), the $\overline{\text{SCKn}}$ pin output state is as follows.

CKPn	$\overline{\text{SCKn}}$ Pin Output
0	Fixed to high level
1	Fixed to low level

Remark n = 0, 1

(2) SOn pin

When CSIn operation is disabled (CSIEn bit = 0), the SOn pin output state is as follows.

TRMDn	DAPn	DIRn	SOn Pin Output
0	×	×	Fixed to low level
1	0	×	SOn latch value (low level)
	1	0	SOTBn7 value
		1	SOTBn0 value

Remarks 1. n = 0, 1
2. ×: Don’t care

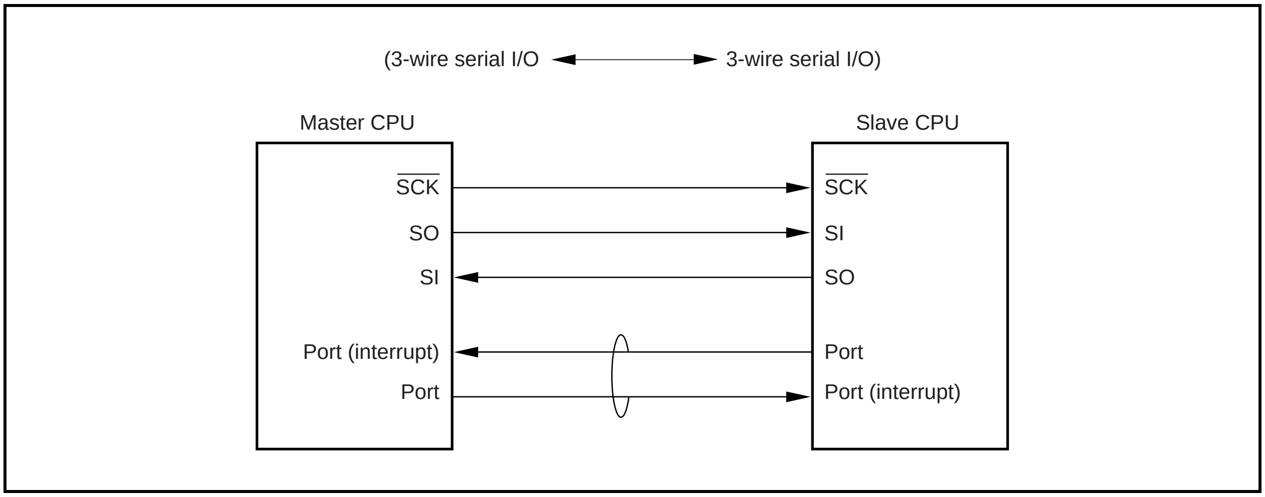
15.6 System Configuration Example

CSIn performs 8-bit length data transfer using three signal lines: a serial clock ($\overline{\text{SCKn}}$), serial input (SIn), and serial output (SOn). This is effective when connecting peripheral I/O that incorporate a conventional clocked serial interface, or a display controller to the V850ES/PM1 (n = 0, 1).

When connecting the V850ES/PM1 to several devices, lines for handshake are required.

Since the first communication bit can be selected as MSB or LSB, communication with various devices can be achieved.

Figure 15-4. System Configuration Example of CSI



CHAPTER 16 INTERRUPT/EXCEPTION PROCESSING FUNCTION

The V850ES/PM1 is provided with a dedicated interrupt controller (INTC) for interrupt servicing and can process a total of 32 interrupt requests.

An interrupt is an event that occurs independently of program execution, and an exception is an event whose occurrence is dependent on program execution.

The V850ES/PM1 can process interrupt requests from the on-chip peripheral hardware and external sources. Moreover, exception processing can be started by the TRAP instruction (software exception) or by generation of an exception event (i.e. fetching of an illegal opcode) (exception trap).

16.1 Features

○ Interrupts

- Non-maskable interrupts: 1 source
- Maskable interrupts: External: 3, Internal: 28 sources
- 8 levels of programmable priorities (maskable interrupts)
- Multiple interrupt control according to priority
- Masks can be specified for each maskable interrupt request.
- Noise elimination, edge detection, and valid edge specification for external interrupt request signals.

○ Exceptions

• Software exceptions:	32 sources
• Exception trap:	2 sources (illegal opcode exception, debug trap)

Interrupt/exception sources are listed in Table 16-1.

Table 16-1. Interrupt/Exception Source List (1/2)

Type	Classification	Default Priority	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Reset	Interrupt	–	RESET	$\overline{\text{RESET}}$ pin input	Pin	0000H	00000000H	Undefined	–
				WDT overflow (WDTRES)	WDT				
Non-maskable	Interrupt	–	NMI	NMI pin valid edge input	Pin	0010H	00000010H	nextPC	–
Software exception	Exception	–	TRAP0n[Note]	TRAP instruction	–	004nH[Note]	00000040H	nextPC	–
		–	TRAP1n[Note]	TRAP instruction	–	005nH[Note]	00000050H	nextPC	–
Exception trap	Exception	–	ILGOP/ DBTRAP	Illegal opcode/ DBTRAP instruction	–	0060H	00000060H	nextPC	–
Maskable	Interrupt	0	INTWDTM	Interval timer overflow	WDT	0080H	00000080H	nextPC	WDTIC
		1	INTP0	INTP0 pin valid edge input	Pin	0090H	00000090H	nextPC	PIC0
		2	INTP1	INTP1 pin valid edge input	Pin	00A0H	000000A0H	nextPC	PIC1
		3	INTP2	INTP2 pin valid edge input	Pin	00B0H	000000B0H	nextPC	PIC2
		4	INTAD	AD conversion completion	ADC	00C0H	000000C0H	nextPC	ADIC
		5	INTRTC	RTC interrupt	RTC	00D0H	000000D0H	nextPC	RTCIC
		6	INTTM000	TM00-CR000 match/ TI001 pin input	TM00	00E0H	000000E0H	nextPC	TMIC000
		7	INTTM001	TM00-CR001 match/ TI000 pin input	TM00	00F0H	000000F0H	nextPC	TMIC001
		8	INTTM010	TM01-CR010 match/ TI011 pin input	TM01	0100H	00000100H	nextPC	TMIC010
		9	INTTM011	TM01-CR011 match/ TI010 pin input	TM01	0110H	00000110H	nextPC	TMIC011
		10	INTTM020	TM02-CR020 match/ TI021 pin input	TM02	0120H	00000120H	nextPC	TMIC020
		11	INTTM021	TM02-CR021 match/ TI020 pin input	TM02	0130H	00000130H	nextPC	TMIC021
		12	INTTM030	TM03-CR030 match/ TI031 pin input	TM03	0140H	00000140H	nextPC	TMIC030
		13	INTTM031	TM03-CR031 match/ TI030 pin input	TM03	0150H	00000150H	nextPC	TMIC031
		14	INTCC100	CC100 capture trigger input/ TM10-CC100 match	TM10	0160H	00000160H	nextPC	CCIC100
		15	INTCC101	CC101 capture trigger input/ TM10-CC101 match	TM10	0170H	00000170H	nextPC	CCIC101
		16	INTOVF10	TM10 overflow	TM10	0180H	00000180H	nextPC	OVFIC10

Note n = 0 to FH

Table 16-1. Interrupt/Exception Source List (2/2)

Type	Classification	Default Priority	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	17	INTCC110	CC110 capture trigger input/ TM11-CC110 match	TM11	0190H	00000190H	nextPC	CCIC110
		18	INTCC111	CC111 capture trigger input/ TM11-CC111 match	TM11	01A0H	000001AH	nextPC	CCIC111
		19	INTOVF11	TM11 overflow	TM11	01B0H	000001B0H	nextPC	OVFIC11
		20	INTTM20	TM20-CR20 match/ TM20 overflow	TM20	01C0H	000001C0H	nextPC	TMIC20
		21	INTTM21	TM21-CR21 match/ TM21 overflow	TM21	01D0H	000001D0H	nextPC	TMIC21
		22	INTCSI0	CSI0 transfer completion	CSI0	01E0H	000001E0H	nextPC	CSIIC0
		23	INTCSI1	CSI1 transfer completion	CSI1	01F0H	000001F0H	nextPC	CSIIC1
		24	INTSRE0	UART0 reception error	UART0	0200H	00000200H	nextPC	SREIC0
		25	INTSR0	UART0 reception completion	UART0	0210H	00000210H	nextPC	SRIC0
		26	INTST0	UART0 transmission completion	UART0	0220H	00000220H	nextPC	STIC0
		27	INTSRE1	UART1 reception error	UART1	0230H	00000230H	nextPC	SREIC1
		28	INTSR1	UART1 reception completion	UART1	0240H	00000240H	nextPC	SRIC1
		29	INTST1	UART1 transmission completion	UART1	0250H	00000250H	nextPC	STIC1
		30	INTROV	RTC overflow	RTC	0260H	00000260H	nextPC	ROVIC

Remarks 1. Default Priority: The priority order when two or more maskable interrupt requests occur at the same time. The highest priority is 0.

Restored PC: The value of the program counter (PC) saved to EIPC, FEPC, or DBPC when interrupt servicing is started. Note, however, that the restored PC when a non-maskable or maskable interrupt is acknowledged while one of the following instructions is being executed does not become the nextPC (if an interrupt is acknowledged during instruction execution, execution stops, and then resumes after the interrupt servicing has finished).

- Load instructions (SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W)
- Division instructions (DIV, DIVH, DIVU, DIVHU)
- PREPARE, DISPOSE instructions (only if an interrupt is generated before the stack pointer is updated)

nextPC: The PC value that starts the processing following interrupt/exception processing.

2. The execution address of the illegal instruction when an illegal opcode exception occurs is calculated by (Restored PC − 4).

16.2 Non-Maskable Interrupts

A non-maskable interrupt request is acknowledged unconditionally, even when interrupts are in the interrupt disabled (DI) status. An NMI is not subject to priority control and takes precedence over all the other interrupts.

The non-maskable interrupt request is input from the NMI pin.

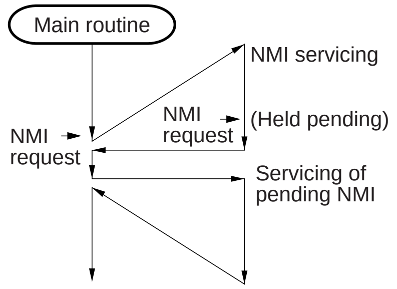
The valid edge of the NMI pin can be selected from four types: “rising edge”, “falling edge”, “both edges”, and “no edge detection”.

If a new NMI request is issued while a NMI is being serviced, the new NMI request is held pending, regardless of the value of the NP bit of the program status word (PSW) in the CPU. The pending NMI interrupt is acknowledged after the NMI currently under execution has been serviced (after the RETI instruction has been executed).

Caution If a non-maskable interrupt request is generated, the values of the PC and PSW are saved to the NMI status save registers (FEPC and FEPSW). Execution can be returned from the NMI servicing by the RETI instruction.

Figure 16-1. Non-Maskable Interrupt Request Acknowledgment Operation

Case in which new non-maskable interrupt (NMI) occurs while a non-maskable interrupt (NMI) is being serviced



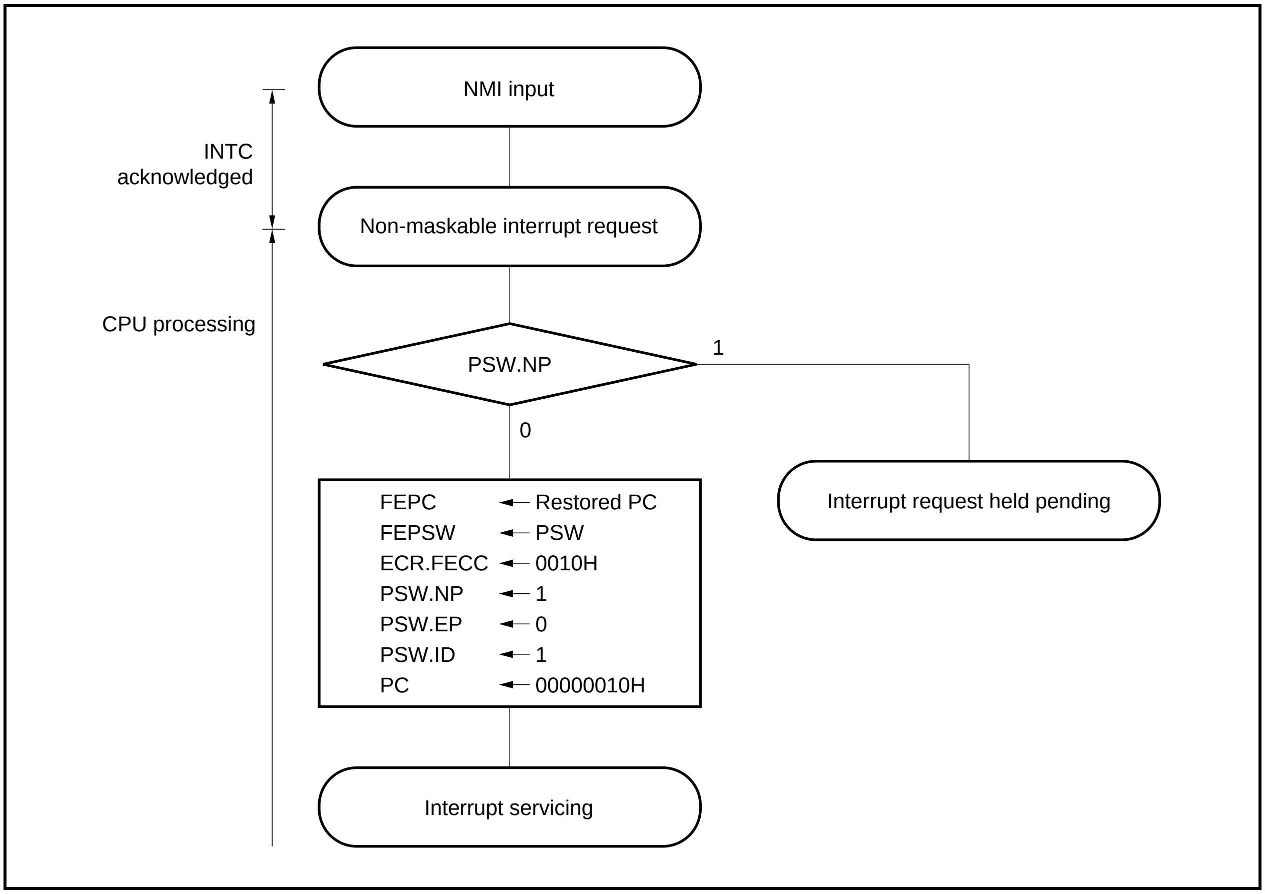
16.2.1 Operation

If a non-maskable interrupt is generated by NMI input, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the restored PC to FEPC.
<2> Saves the current PSW to FEPSW.
<3> Writes exception code 0010H to the higher halfword (FECC) of ECR.
<4> Sets the PSW.NP and PSW.ID bits (1) and clears the PSW.EP bit (0).
<5> Sets the handler address (00000010H) corresponding to the non-maskable interrupt to the PC, and transfers control.

The servicing configuration of a non-maskable interrupt is shown in Figure 16-2.

Figure 16-2. Servicing Configuration of Non-Maskable Interrupt



16.2.2 Restore

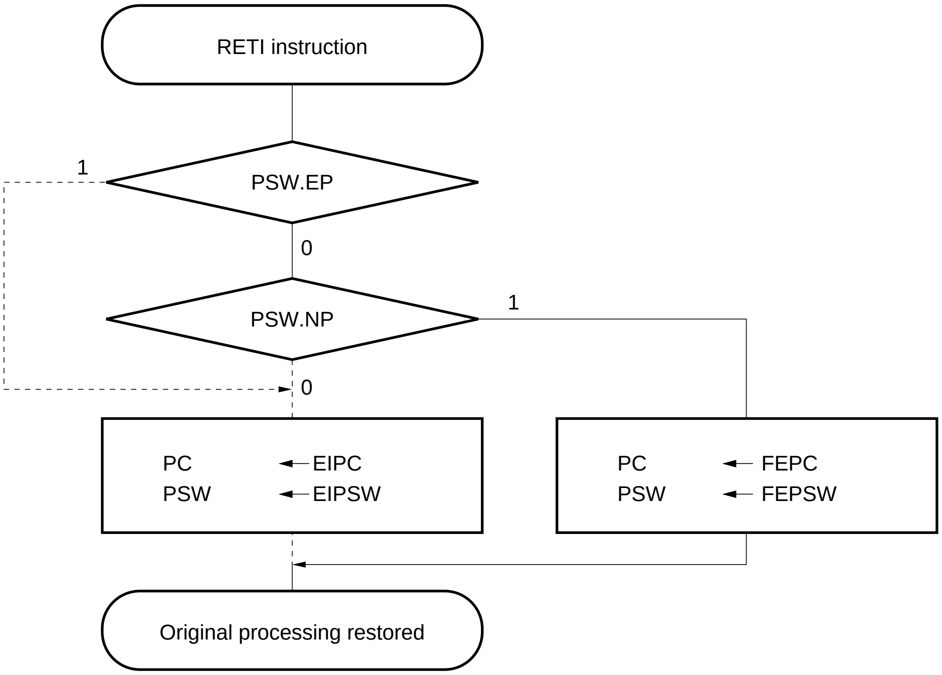
Execution is restored from the NMI by the RETI instruction.

When the RETI instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

<1> Loads the restored PC and PSW from FEPC and FEPSW, respectively, because the PSW.EP bit is 0 and the PSW.NP bit is 1.
<2> Transfers control back to the address of the restored PC and PSW.

Figure 16-3 illustrates how the RETI instruction is processed.

Figure 16-3. RETI Instruction Processing



Caution **When the PSW.EP bit and PSW.NP bit are changed by the LDSR instruction during non-maskable interrupt servicing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 0 and PSW.NP back to 1 using the LDSR instruction immediately before the RETI instruction.**

Remark The solid line shows the CPU processing flow.

16.2.3 NP flag

The NP flag is a status flag that indicates that non-maskable interrupt servicing is under execution.

This flag is set when a non-maskable interrupt request has been acknowledged, and masks non-maskable interrupt requests to prohibit multiple interrupts from being acknowledged.

After reset: 00000020H

	31 8	7	6	5	4	3	2	1	0
PSW	0	NP	EP	ID	SAT	CY	OV	S	Z

NP	Non-maskable interrupt servicing status
0	No non-maskable interrupt servicing
1	Non-maskable interrupt currently being serviced

16.3 Maskable Interrupts

Maskable interrupt requests can be masked by interrupt control registers. The V850ES/PM1 has 31 maskable interrupt sources.

If two or more maskable interrupt requests are generated at the same time, they are acknowledged according to the default priority. In addition to the default priority, eight levels of priorities can be specified by using the interrupt control registers (programmable priority control).

When an interrupt request has been acknowledged, the acknowledgment of other maskable interrupt requests is disabled and the interrupt disabled (DI) status is set.

When the EI instruction is executed in an interrupt service routine, the interrupt enabled (EI) status is set, which enables servicing of interrupts having a higher priority than the interrupt request in progress (specified by the interrupt control register). Note that only interrupts with a higher priority will have this capability; interrupts with the same priority level cannot be nested.

To enable multiple interrupts, however, save EIPC and EIPSW to memory or registers before executing the EI instruction, and execute the DI instruction before the RETI instruction to restore the original values of EIPC and EIPSW.

16.3.1 Operation

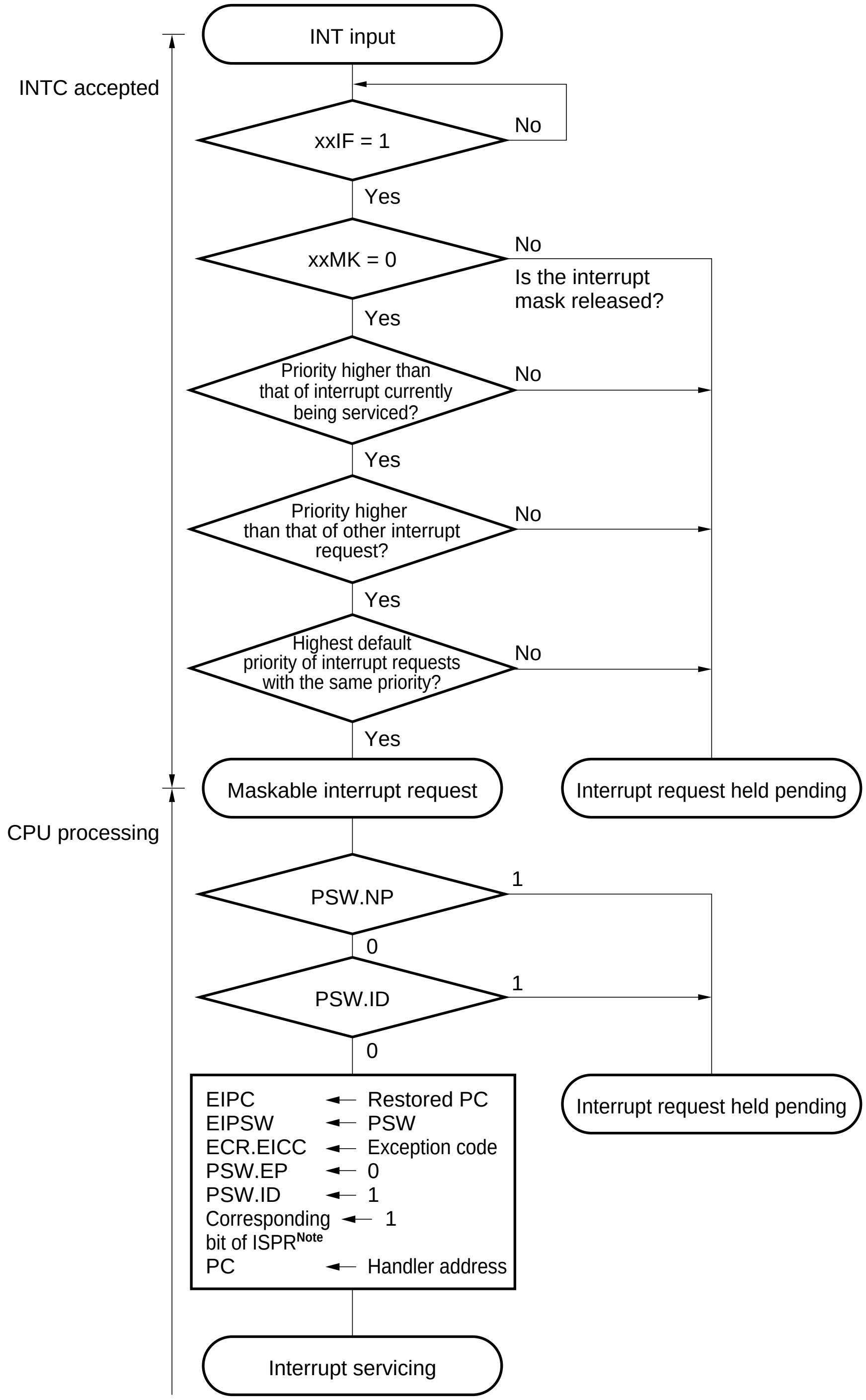
If a maskable interrupt occurs by INT input, the CPU performs the following processing, and transfers control to a handler routine.

<1> Saves the restored PC to EIPC.
<2> Saves the current PSW to EIPSW.
<3> Writes an exception code to the lower halfword of ECR (EICC).
<4> Sets the PSW.ID bit (1) and clears the PSW.EP bit (0).
<5> Sets the handler address corresponding to each interrupt to the PC, and transfers control.

The maskable interrupt request masked by INTC and the maskable interrupt request generated while another interrupt is being serviced (while PSW.NP = 1 or PSW.ID = 1) are held pending inside INTC. In this case, servicing a new maskable interrupt is started in accordance with the priority of the pending maskable interrupt request if either the maskable interrupt is unmasked or PSW.NP and PSW.ID are cleared to 0 by using the RETI or LDSR instruction.

How maskable interrupts are serviced is illustrated below.

Figure 16-4. Maskable Interrupt Servicing


INT input
INTC accepted
xxIF = 1
No
Yes
xxMK = 0
No
Is the interrupt mask released?
Yes
Priority higher than that of interrupt currently being serviced?
No
Yes
Priority higher than that of other interrupt request?
No
Yes
Highest default priority of interrupt requests with the same priority?
No
Yes
Maskable interrupt request
Interrupt request held pending
CPU processing
PSW.NP
1
0
PSW.ID
1
0
EIPC ← Restored PC
EIPSW ← PSW
ECR.EICC ← Exception code
PSW.EP ← 0
PSW.ID ← 1
Corresponding bit of ISPR[Note] ← 1
PC ← Handler address
Interrupt request held pending
Interrupt servicing

Note For the ISPR register, see **16.3.6 In-service priority register (ISPR)**.

The INT input masked by the interrupt controllers and the INT input that occurs while another interrupt is being serviced (when PSW.NP = 1 or PSW.ID = 1) are held pending internally by the interrupt controller. In such case, if the interrupts are unmasked, or when PSW.NP = 0 and PSW.ID = 0 as set by the RETI and LDSR instructions, input of the pending INT starts the new maskable interrupt servicing.

16.3.2 Restore

Recovery from maskable interrupt servicing is carried out by the RETI instruction.

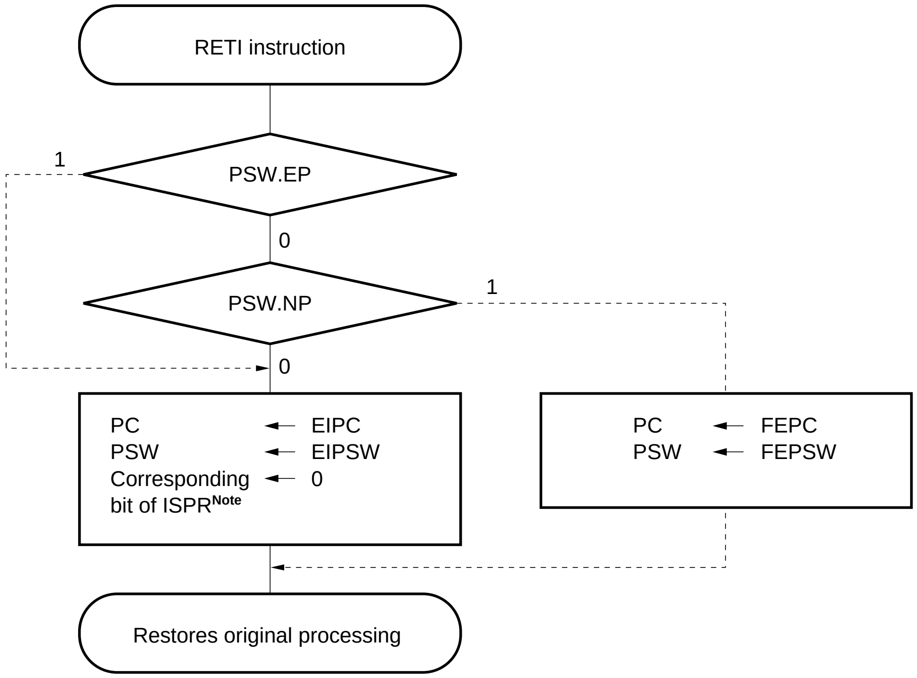
When the RETI instruction is executed, the CPU performs the following steps, and transfers control to the address of the restored PC.

<1> Loads the restored PC and PSW from EIPC and EIPSW because the PSW.EP bit is 0 and the PSW.NP bit is 0.

<2> Transfers control to the address of the restored PC and PSW.

Figure 16-5 illustrates the processing of the RETI instruction.

Figure 16-5. RETI Instruction Processing



Note For the ISPR register, see **16.3.6 In-service priority register (ISPR)**.

Caution When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during maskable interrupt servicing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 0 and PSW.NP back to 0 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

16.3.3 Priorities of maskable interrupts

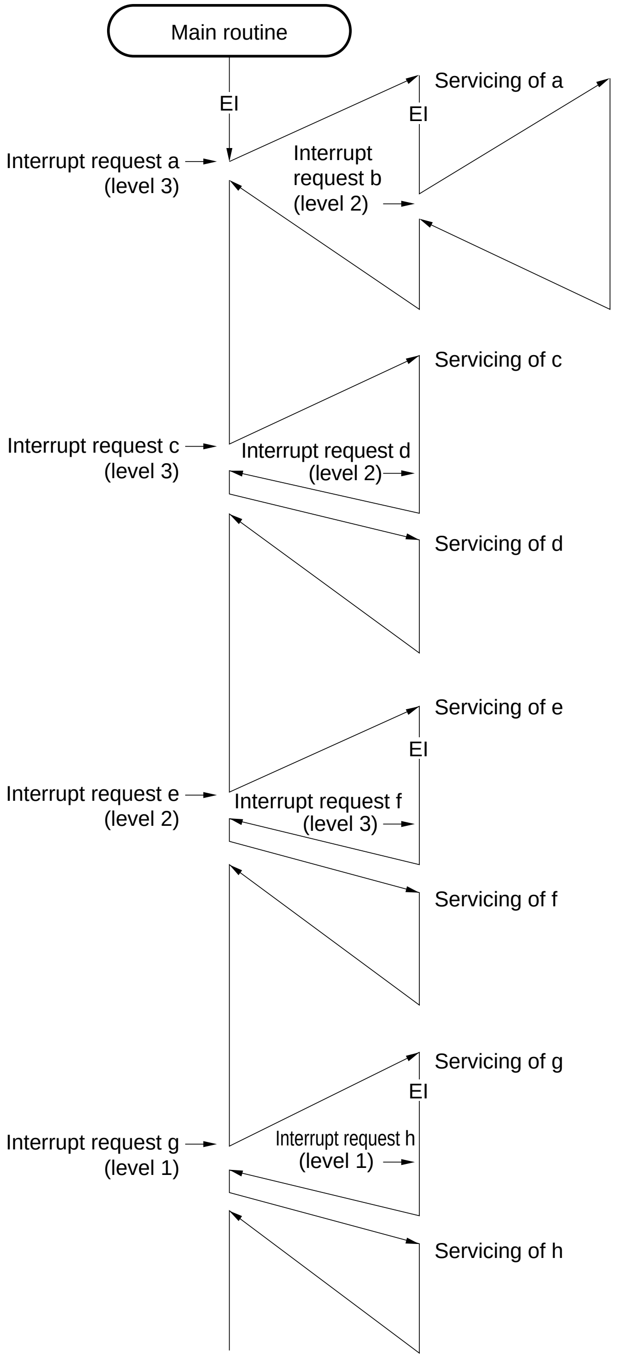
The V850ES/PM1 provides multiple interrupt servicing in which an interrupt is acknowledged while another interrupt is being serviced. Multiple interrupts can be controlled by priority levels.

There are two types of priority level control: control based on the default priority levels, and control based on the programmable priority levels that are specified by the interrupt priority level specification bit (xxPRn) of the interrupt control register (xxICn). When two or more interrupts having the same priority level specified by the xxPRn bit are generated at the same time, interrupts are serviced in order depending on the priority level allocated to each interrupt request type (default priority level) beforehand. For more information, refer to **Table 16-1 Interrupt Source List**. The programmable priority control customizes interrupt requests into eight levels by setting the priority level specification flag.

Note that when an interrupt request is acknowledged, the ID flag of PSW is automatically set to 1. Therefore, when multiple interrupts are to be used, clear the ID flag to 0 beforehand (for example, by placing the EI instruction in the interrupt service program) to set the interrupt enable mode.

Remark xx: Identification name of each peripheral unit (refer to **Table 16-2**)
n: Peripheral unit number (refer to **Table 16-2**).

Figure 16-6. Example of Processing in Which Another Interrupt Request Is Issued While an Interrupt Is Being Serviced (1/2)


Main routine
EI
Servicing of a
EI
Servicing of b
Interrupt request a (level 3)
Interrupt request b (level 2)
Servicing of c
Interrupt request c (level 3)
Interrupt request d (level 2)
Servicing of d
Servicing of e
EI
Interrupt request e (level 2)
Interrupt request f (level 3)
Servicing of f
Servicing of g
EI
Interrupt request g (level 1)
Interrupt request h (level 1)
Servicing of h

Interrupt request b is acknowledged because the priority of b is higher than that of a and interrupts are enabled.

Although the priority of interrupt request d is higher than that of c, d is held pending because interrupts are disabled.

Interrupt request f is held pending even if interrupts are enabled because its priority is lower than that of e.

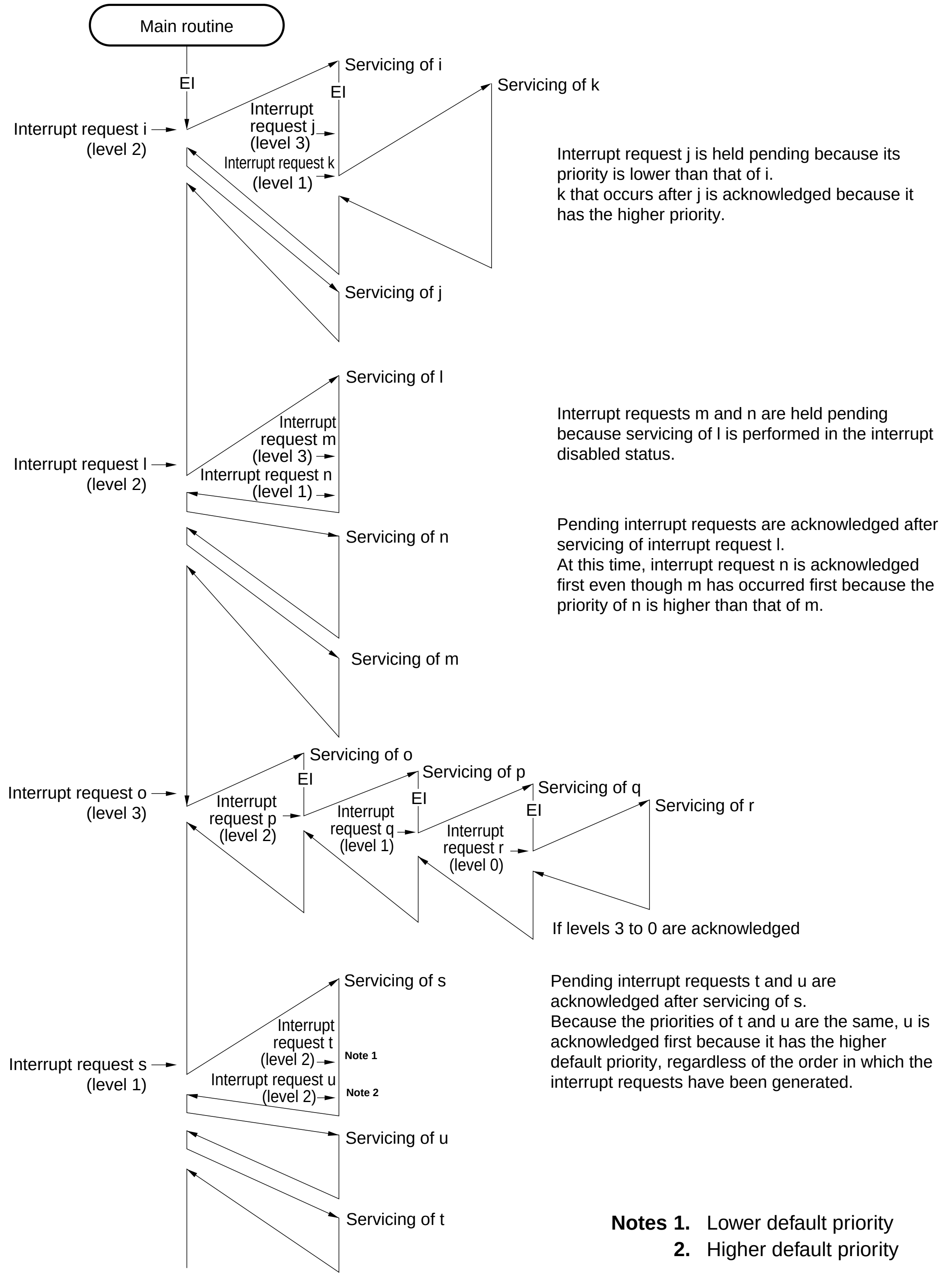
Interrupt request h is held pending even if interrupts are enabled because its priority is the same as that of g.

Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Remarks 1. a to u in the figure are the temporary names of interrupt requests shown for the sake of explanation.

2. The default priority in the figure indicates the relative priority between two interrupt requests.

Figure 16-6. Example of Processing in Which Another Interrupt Request Is Issued While an Interrupt Is Being Serviced (2/2)



Interrupt request j is held pending because its priority is lower than that of i.
k that occurs after j is acknowledged because it has the higher priority.

Interrupt requests m and n are held pending because servicing of l is performed in the interrupt disabled status.

Pending interrupt requests are acknowledged after servicing of interrupt request l.
At this time, interrupt request n is acknowledged first even though m has occurred first because the priority of n is higher than that of m.

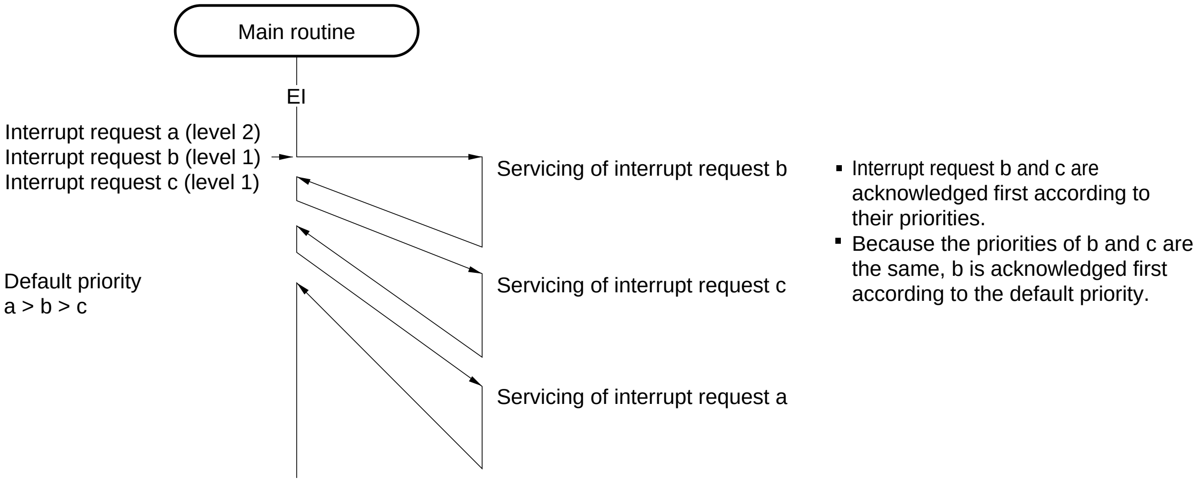
If levels 3 to 0 are acknowledged

Pending interrupt requests t and u are acknowledged after servicing of s.
Because the priorities of t and u are the same, u is acknowledged first because it has the higher default priority, regardless of the order in which the interrupt requests have been generated.

Notes 1. Lower default priority
2. Higher default priority

Caution **To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.**

Figure 16-7. Example of Servicing Interrupt Requests Simultaneously Generated


Main routine
EI
Interrupt request a (level 2)
Interrupt request b (level 1)
Interrupt request c (level 1)
Default priority
a > b > c
Servicing of interrupt request b
Servicing of interrupt request c
Servicing of interrupt request a
▪ Interrupt request b and c are acknowledged first according to their priorities.
▪ Because the priorities of b and c are the same, b is acknowledged first according to the default priority.

Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Remarks 1. a to c in the figure are the temporary names of interrupt requests shown for the sake of explanation.

2. The default priority in the figure indicates the relative priority between two interrupt requests.

16.3.4 Interrupt control register (xxICn)

An interrupt control register is assigned to each interrupt request (maskable interrupt) and sets the control conditions for each maskable interrupt request.

This register can be read or written in 8-bit or 1-bit units.

This register is set to 47H after reset.

Caution Read the xxICn.xxIFn bit while interrupts are disabled (DI status). If the xxIFn bit is read while interrupts are enabled (EI status), and if acknowledging an interrupt conflicts with reading the bit, the correct value of the bit may not be read.

After reset: 47H R/W Address: FFFFF110H to FFFFF14CH

	<7>	<6>	5	4	3	<2>	<1>	<0>
xxICn	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0

xxIFn	Interrupt request flag[Note]
0	Interrupt request not issued
1	Interrupt request issued

xxMKn	Interrupt mask flag
0	Interrupt servicing enabled
1	Interrupt servicing disabled (pending)

xxPRn2	xxPRn1	xxPRn0	Interrupt priority specification bit
0	0	0	Specifies level 0 (highest).
0	0	1	Specifies level 1.
0	1	0	Specifies level 2.
0	1	1	Specifies level 3.
1	0	0	Specifies level 4.
1	0	1	Specifies level 5.
1	1	0	Specifies level 6.
1	1	1	Specifies level 7 (lowest).

Note The flag xxlFn is reset automatically by the hardware if an interrupt request is acknowledged.

Remark xx: Identification name of each peripheral unit (refer to **Table 16-2**)
n: Peripheral unit number (refer to **Table 16-2**).

The addresses and bits of the interrupt control registers are as follows.

Table 16-2. Interrupt Control Register (xxICn)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF110H	WDTIC	WDTIF	WDTMK	0	0	0	WDTPR2	WDTPR1	WDTPR0
FFFFF112H	PIC0	PIF0	PMK0	0	0	0	PPR02	PPR01	PPR00
FFFFF114H	PIC1	PIF1	PMK1	0	0	0	PPR12	PPR11	PPR10
FFFFF116H	PIC2	PIF2	PMK2	0	0	0	PPR22	PPR21	PPR20
FFFFF118H	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0
FFFFF11AH	RTCIC	RTCIF	RTCMK	0	0	0	RTCPR2	RTCPR1	RTCPR0
FFFFF11CH	TMIC000	TMIF000	TMMK000	0	0	0	TMPR0002	TMPR0001	TMPR0000
FFFFF11EH	TMIC001	TMIF001	TMMK001	0	0	0	TMPR0012	TMPR0011	TMPR0010
FFFFF120H	TMIC010	TMIF010	TMMK010	0	0	0	TMPR0102	TMPR0101	TMPR0100
FFFFF122H	TMIC011	TMIF011	TMMK011	0	0	0	TMPR0112	TMPR0111	TMPR0110
FFFFF124H	TMIC020	TMIF020	TMMK020	0	0	0	TMPR0202	TMPR0201	TMPR0200
FFFFF126H	TMIC021	TMIF021	TMMK021	0	0	0	TMPR0212	TMPR0211	TMPR0210
FFFFF128H	TMIC030	TMIF030	TMMK030	0	0	0	TMPR0302	TMPR0301	TMPR0300
FFFFF12AH	TMIC031	TMIF031	TMMK031	0	0	0	TMPR0312	TMPR0311	TMPR0310
FFFFF12CH	CCIC100	CCIF100	CCMK100	0	0	0	CCPR1002	CCPR1001	CCPR1000
FFFFF12EH	CCIC101	CCIF101	CCMK101	0	0	0	CCPR1012	CCPR1011	CCPR1010
FFFFF130H	OVFIC10	OVFIF10	OVFMK10	0	0	0	OVFPR102	OVFPR101	OVFPR100
FFFFF132H	CCIC110	CCIF110	CCMK110	0	0	0	CCPR1102	CCPR1101	CCPR1100
FFFFF134H	CCIC111	CCIF111	CCMK111	0	0	0	CCPR1112	CCPR1111	CCPR1110
FFFFF136H	OVFIC11	OVFIF11	OVFMK11	0	0	0	OVFPR112	OVFPR111	OVFPR110
FFFFF138H	TMIC20	TMIF20	TMMK20	0	0	0	TMPR202	TMPR201	TMPR200
FFFFF13AH	TMIC21	TMIF21	TMMK21	0	0	0	TMPR212	TMPR211	TMPR210
FFFFF13CH	CSIIC0	CSIIF0	CSIMK0	0	0	0	CSIPR02	CSIPR01	CSIPR00
FFFFF13EH	CSIIC1	CSIIF1	CSIMK1	0	0	0	CSIPR12	CSIPR11	CSIPR10
FFFFF140H	SREIC0	SREIF0	SREMK0	0	0	0	SREPR02	SREPR01	SREPR00
FFFFF142H	SRIC0	SRIF0	SRMK0	0	0	0	SRPR02	SRPR01	SRPR00
FFFFF144H	STIC0	STIF0	STMK0	0	0	0	STPR02	STPR01	STPR00
FFFFF146H	SREIC1	SREIF1	SREMK1	0	0	0	SREPR12	SREPR11	SREPR10
FFFFF148H	SRIC1	SRIF1	SRMK1	0	0	0	SRPR12	SRPR11	SRPR10
FFFFF14AH	STIC1	STIF1	STMK1	0	0	0	STPR12	STPR11	STPR10
FFFFF14CH	ROVIC	ROVIF	ROVMK	0	0	0	ROVPR2	ROVPR1	ROVPR0

16.3.5 Interrupt mask registers 0, 1 (IMR0, IMR1)

These registers set the interrupt mask state for the maskable interrupts. The xxMKn bit of the IMR0 and IMR1 registers is equivalent to the xxICn.xxMKn bit.

The IMRm register can be read or written in 16-bit units (m = 0, 1).

If the higher 8 bits of the IMRm register are used as an IMRmH register and the lower 8 bits as an IMRmL register, these registers can be read or written in 8-bit or 1-bit units (m = 0, 1).

This register is set to FFFFH after reset.

Caution The device file defines the xxICn.xxMKn bit as a reserved word. If a bit is manipulated using the name of xxMKn, the contents of the xxICn register, instead of the IMRm register, are rewritten (as a result, the contents of the IMRm register are also rewritten).

After reset: FFFFH R/W Address: FFFFF102H

	15	14	13	12	11	10	9	8
IMR1	1	ROVMK	STMK1	SRMK1	SREMK1	STMK0	SRMK0	SREMK0
	7	6	5	4	3	2	1	0
	CSIMK1	CSIMK0	TMMK21	TMMK20	OVFMK11	CCMK111	CCMK110	OVFMK10

After reset: FFFFH R/W Address: FFFFF100H

	15	14	13	12	11	10	9	8
IMR0	CCMK101	CCMK100	TMMK031	TMMK030	TMMK021	TMMK020	TMMK011	TMMK010
	7	6	5	4	3	2	1	0
	TMMK001	TMMK000	RTCMK	ADMK	PMK2	PMK1	PMK0	WDTMK

xxMKn	Setting of interrupt mask flag
0	Interrupt servicing enabled.
1	Interrupt servicing disabled.

Note Be sure to set bit 15 of the IMR1 register to 1.

Remark xx: Identification name of each peripheral unit (refer to **Table 16-2**).
n: Peripheral unit number (refer to **Table 16-2**)

16.3.6 In-service priority register (ISPR)

This register holds the priority level of the maskable interrupt currently acknowledged. When an interrupt request is acknowledged, the bit of this register corresponding to the priority level of that interrupt request is set to 1 and remains set while the interrupt is serviced.

When the RETI instruction is executed, the bit corresponding to the interrupt request having the highest priority is automatically reset to 0 by hardware. However, it is not reset to 0 when execution is returned from non-maskable interrupt servicing or exception processing.

This register is read-only in 8-bit or 1-bit units.

This register is set to 00H after reset.

Caution **If an interrupt is acknowledged in the interrupt enabled (EI) status while the ISPR register is being read, the value of the ISPR register after the bit of the register has been set by the interrupt may be read. To accurately read the value of the ISPR register before the interrupt is acknowledged, read the register in the interrupt disabled (DI) status.**

After reset: 00H R Address: FFFFF1FAH

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0

ISPRn	Priority of interrupt currently acknowledged
0	Interrupt request with priority n not acknowledged
1	Interrupt request with priority n acknowledged

Remark n = 0 to 7 (priority level)

16.3.7 ID flag

This flag controls the maskable interrupt’s operating state, and stores control information regarding enabling or disabling of interrupt requests. An interrupt disable flag (ID) is incorporated, which is assigned to the PSW.

This register is set to 00000020H after reset.

After reset: 00000020H

	31	8	7	6	5	4	3	2	1	0
PSW	0		NP	EP	ID	SAT	CY	OV	S	Z

ID	Maskable interrupt servicing specificationNote
0	Maskable interrupt request acknowledgment enabled
1	Maskable interrupt request acknowledgment disabled (pending)

Note Interrupt disable flag (ID) function

This bit is set to 1 by the DI instruction and reset to 0 by the EI instruction. Its value is also modified by the RETI instruction or LDSR instruction when referencing the PSW.

Non-maskable interrupt requests and exceptions are acknowledged regardless of this flag. When a maskable interrupt is acknowledged, the ID flag is automatically set to 1 by hardware.

When interrupt servicing is completed, the flag is cleared by RETI instruction execution.

The interrupt request generated during the acknowledgment disabled period (ID = 1) is acknowledged when the xxIFn bit of xxICn is set to 1, and the ID flag is reset to 0.

16.4 External Interrupt Request Input Pins (NMI, INTP0 to INTP2)

16.4.1 Noise elimination

(1) Noise elimination for NMI pin

The NMI pin includes a noise eliminator that operates using analog delay. Therefore, a signal input to the NMI pin is not detected as an edge unless it maintains its input level for a certain period. The edge is detected only after a certain period has elapsed.

The NMI pin can be used for releasing the STOP mode. In the STOP mode, noise elimination using the system clock is not performed because the internal system clock is stopped.

(2) Noise elimination for INTP0 to INTP2 pins

The INTP0 to INTP2 pins include a noise eliminator that operates using analog delay. Therefore, a signal input to each pin is not detected as an edge unless it maintains its input level for a certain period. The edge is detected only after a certain period has elapsed.

The INTP0 to INTP2 pins can be used for releasing the STOP mode. In the STOP mode, noise elimination using the system clock is not performed because the internal system clock is stopped.

16.4.2 Edge detection

The valid edges of the NMI and INTP0 to INTP2 pins can be selected from the following four types for each pin.

- Rising edge
- Falling edge
- Both edges
- No edge detection

After reset, the edge detection for the NMI and INTP0 to INTP2 pins is set to “no edge detection”. Therefore, interrupt requests cannot be acknowledged (the NMI pin functions as a normal port) unless a valid edge is specified by the INTF0 and INTR0 registers.

When using port 0 as an output port, set the NMI and INTP0 to INTP2 pin valid edge to “no edge detection”.

(1) External interrupt rising edge specification register 0 (INTR0), external interrupt falling edge specification register 0 (INTF0)

INTR0 and INTF0 are 8-bit registers that specify detection of the rising and falling edges of the NMI and INTP0 to INTP2 pins.

These registers can be read or written in 8-bit or 1-bit units.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF0n and INTR0n bits to 0, and then set the port mode.

After reset: 00H R/W Address: INTR0 FFFFFC20H, INTF0 FFFFFC00H

	7	6	5	4	<3>	<2>	<1>	<0>
INTR0	0	0	0	0	INTR03	INTR02	INTR01	INTR00
					INTP2	INTP1	INTP0	NMI

	7	6	5	4	<3>	<2>	<1>	<0>
INTF0	0	0	0	0	INTF03	INTF02	INTF01	INTF00
					INTP2	INTP1	INTP0	NMI

Remark For how to specify a valid edge, refer to **Table 16-3**.

Table 16-3. Valid Edge Specification

INTF0n	INTR0n	Valid Edge Specification (n = 0 to 3)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Remark n = 0: Control of NMI pin
n = 1 to 3: Control of INTP0 to INTP2 pins

16.5 Software Exception

A software exception is generated when the CPU executes the TRAP instruction, and can always be acknowledged.

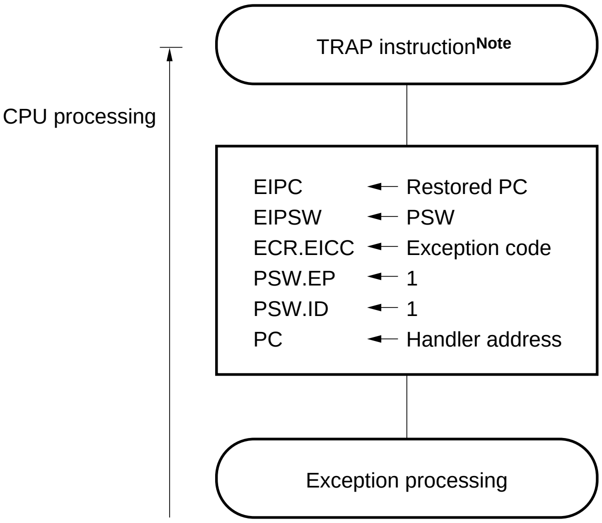
16.5.1 Operation

If a software exception occurs, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the restored PC to EIPC.
<2> Saves the current PSW to EIPSW.
<3> Writes an exception code to the lower 16 bits (EICC) of ECR (interrupt source).
<4> Sets the PSW.EP and PSW.ID bits (1).
<5> Sets the handler address (00000040H or 00000050H) corresponding to the software exception to the PC, and transfers control.

Figure 16-8 illustrates the processing of a software exception.

Figure 16-8. Software Exception Processing



Note TRAP instruction format: TRAP vector (the vector is a value from 00H to 1FH.)

The handler address is determined by the TRAP instruction’s operand (vector). If the vector is 00H to 0FH, it becomes 00000040H, and if the vector is 10H to 1FH, it becomes 00000050H.

16.5.2 Restore

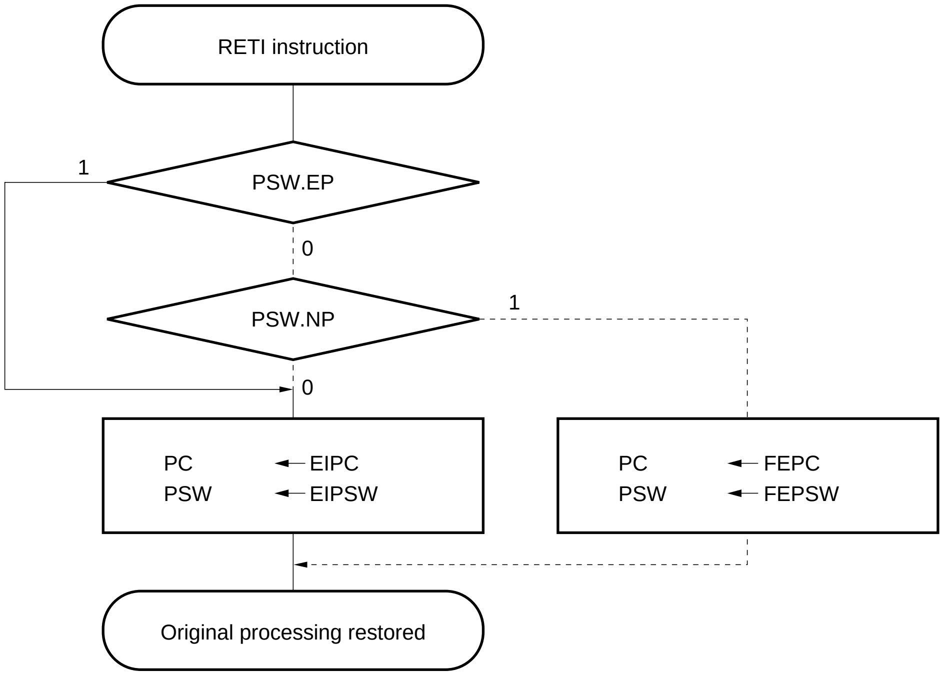
Recovery from software exception processing is carried out by the RETI instruction.

By executing the RETI instruction, the CPU carries out the following processing and shifts control to the restored PC's address.

<1> Loads the restored PC and PSW from EIPC and EIPSW because the PSW.EP bit is 1.
<2> Transfers control to the address of the restored PC and PSW.

Figure 16-9 illustrates the processing of the RETI instruction.

Figure 16-9. RETI Instruction Processing



Caution **When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during the software exception processing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 1 using the LDSR instruction immediately before the RETI instruction.**

Remark The solid line shows the CPU processing flow.

16.5.3 Exception status flag (EP)

The EP flag is bit 6 of the PSW, and is a status flag used to indicate that exception processing is in progress. It is set when an exception occurs.

After reset: 00000020H

	31	8	7	6	5	4	3	2	1	0
PSW	0		NP	**EP**	ID	SAT	CY	OV	S	Z

EP	Exception processing status
0	Exception processing not in progress.
1	Exception processing in progress.

16.6 Exception Trap

An exception trap is an interrupt that is requested when the illegal execution of an instruction takes place. In the V850ES/PM1, an illegal opcode exception (ILGOP: Illegal Opcode Trap) is considered as an exception trap.

16.6.1 Illegal opcode definition

The illegal instruction has an opcode (bits 10 to 5) of 111111B, a sub-opcode (bits 26 to 23) of 0111B to 1111B, and a sub-opcode (bit 16) of 0B. An exception trap is generated when an instruction applicable to this illegal instruction is executed.

15 11	10 5	4 0	31 27	26 23	22	16
X X X X X	1 1 1 1 1 1	X X X X X	X X X X X	0 1 1 1 to 1 1 1 1	X X X X X X	0

X: Arbitrary

Caution Since it is possible to assign this instruction to an illegal opcode in the future, it is recommended that it not be used.

(1) Operation

If an exception trap occurs, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the restored PC to DBPC.

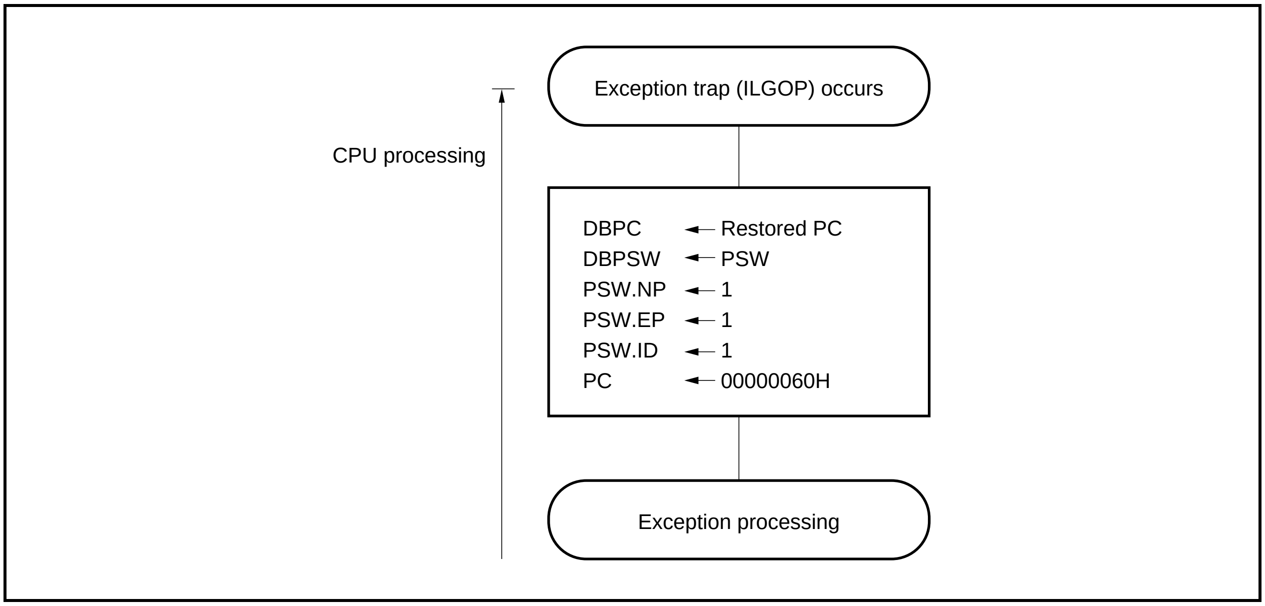
<2> Saves the current PSW to DBPSW.

<3> Sets the PSW.NP, PSW.EP, and PSW.ID bits (1).

<4> Sets the handler address (00000060H) corresponding to the exception trap to the PC, and transfers control.

Figure 16-10 illustrates the processing of the exception trap.

Figure 16-10. Exception Trap Processing



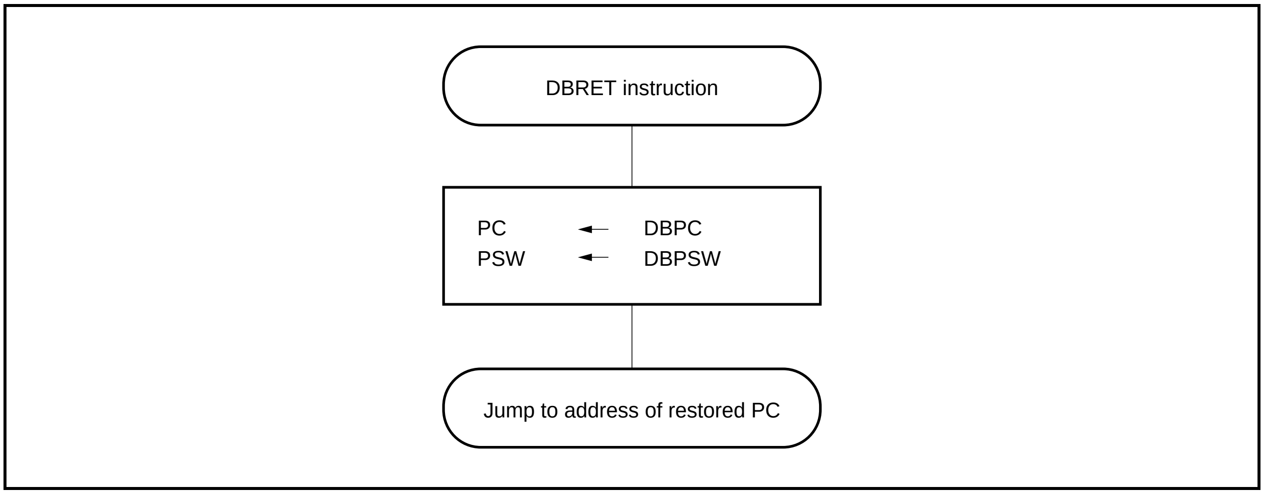
(2) Restore

Recovery from an exception trap is carried out by the DBRET instruction. By executing the DBRET instruction, the CPU carries out the following processing and controls the address of the restored PC.

<1> Loads the restored PC and PSW from DBPC and DBPSW.
<2> Transfers control to the address indicated by the restored PC and PSW.

Figure 16-11 illustrates the restore processing from an exception trap.

Figure 16-11. Restore Processing from Exception Trap



16.6.2 Debug trap

The debug trap is an exception that can be acknowledged every time and is generated by execution of the DBTRAP instruction.

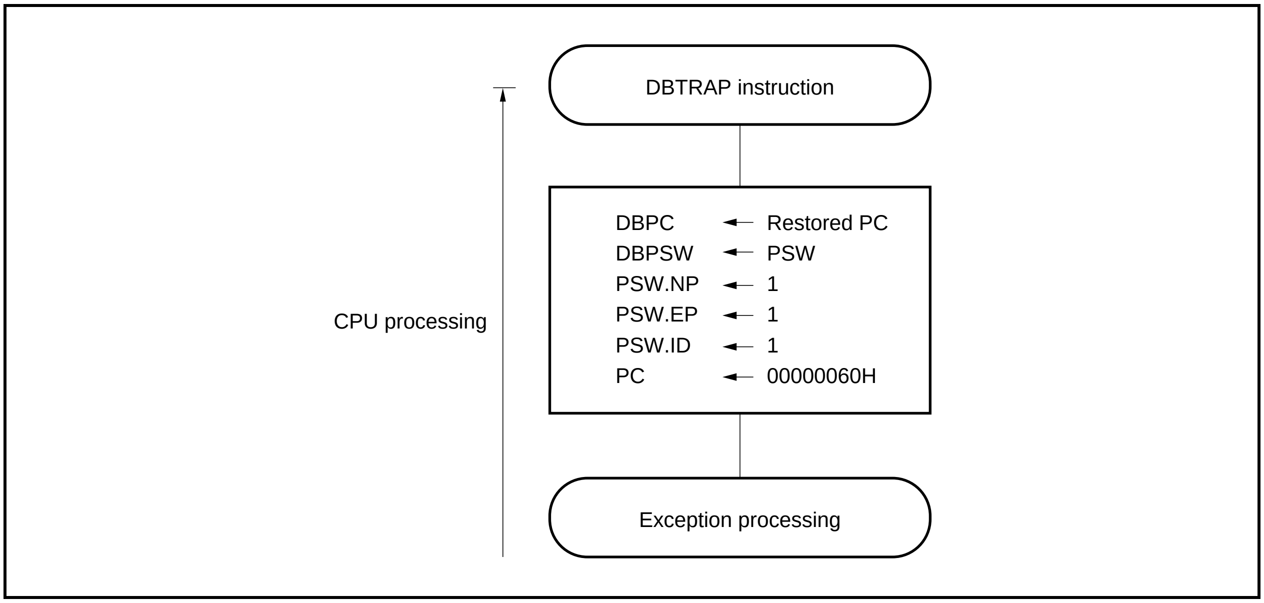
When the debug trap is generated, the CPU performs the following processing.

(1) Operation

<1> Saves the restored PC to DBPC.
<2> Saves the current PSW to DBPSW.
<3> Sets the PSW.NP, PSW.EP and PSW.ID bits of the (1).
<4> Sets the handler address (00000060H) corresponding to the debug trap to the PC and transfers control.

Figure 16-12 illustrates the processing of the debug trap.

Figure 16-12. Debug Trap Processing



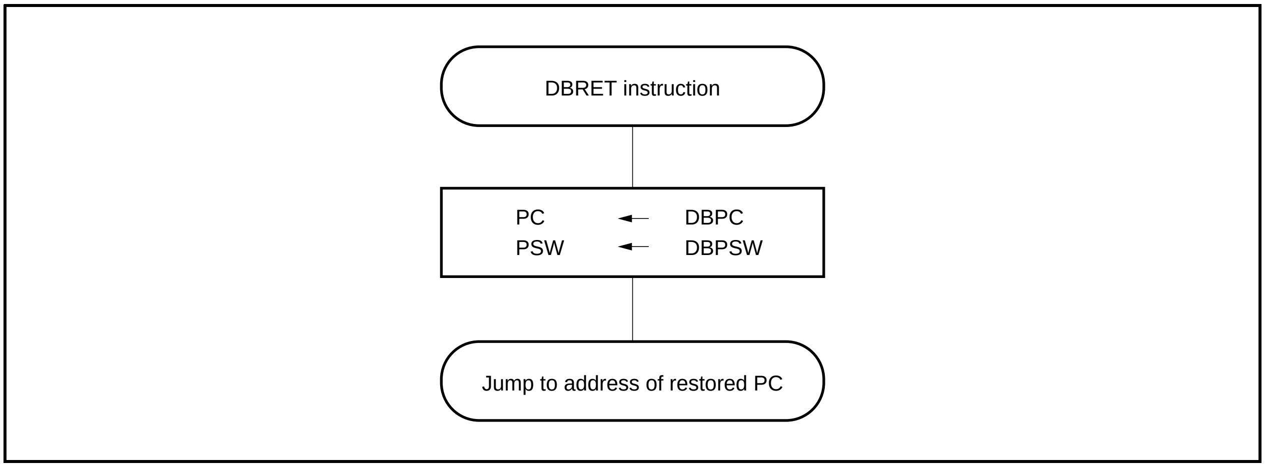
(2) Restore

Recovery from a debug trap is carried out by the DBRET instruction. By executing the DBRET instruction, the CPU carries out the following processing and controls the address of the restored PC.

<1> Loads the restored PC and PSW from DBPC and DBPSW.
<2> Transfers control to the address indicated by the restored PC and PSW.

Figure 16-13 illustrates the restore processing from a debug trap.

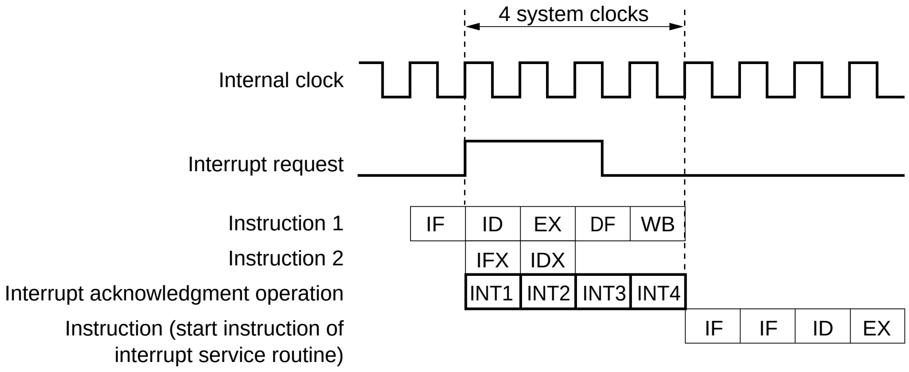
Figure 16-13. Restore Processing from Debug Trap



16.7 Interrupt Acknowledge Time of CPU

The interrupt response time from generation of an interrupt request to the start of interrupt servicing is illustrated below.

Figure 16-14. Pipeline Operation at Interrupt Request Acknowledgment (Outline)



Remark INT1 to INT4: Interrupt acknowledgment processing
IFX: Invalid instruction fetch
IDX: Invalid instruction decode

Interrupt acknowledge time (internal system clock)			Condition
	Internal interrupt	External interrupt	
Minimum	4	4 + Analog delay time	The following cases are exceptions. • In IDLE/software STOP/sub-IDLE/sub-software STOP mode • External bus access • Two or more interrupt request non-sample instructions are executed in succession • Access to peripheral I/O register
Maximum	6	6 + Analog delay time	

16.8 Periods in Which Interrupts Are Not Acknowledged by CPU

An interrupt is acknowledged by the CPU while an instruction is being executed. However, no interrupt will be acknowledged between an interrupt request non-sample instruction and the next instruction (interrupt is held pending).

The interrupt request non-sample instructions are as follows.

- EI instruction
- DI instruction
- LDSR reg2, 0x5 instruction (for PSW)
- The store instruction for the command register (PRCMD)
- The load, store, or bit manipulation instructions for the following interrupt-related registers.
  Interrupt control register (xxICn), interrupt mask registers 0 and 1 (IMR0 and IMR1)

CHAPTER 17 STANDBY FUNCTION

17.1 Overview

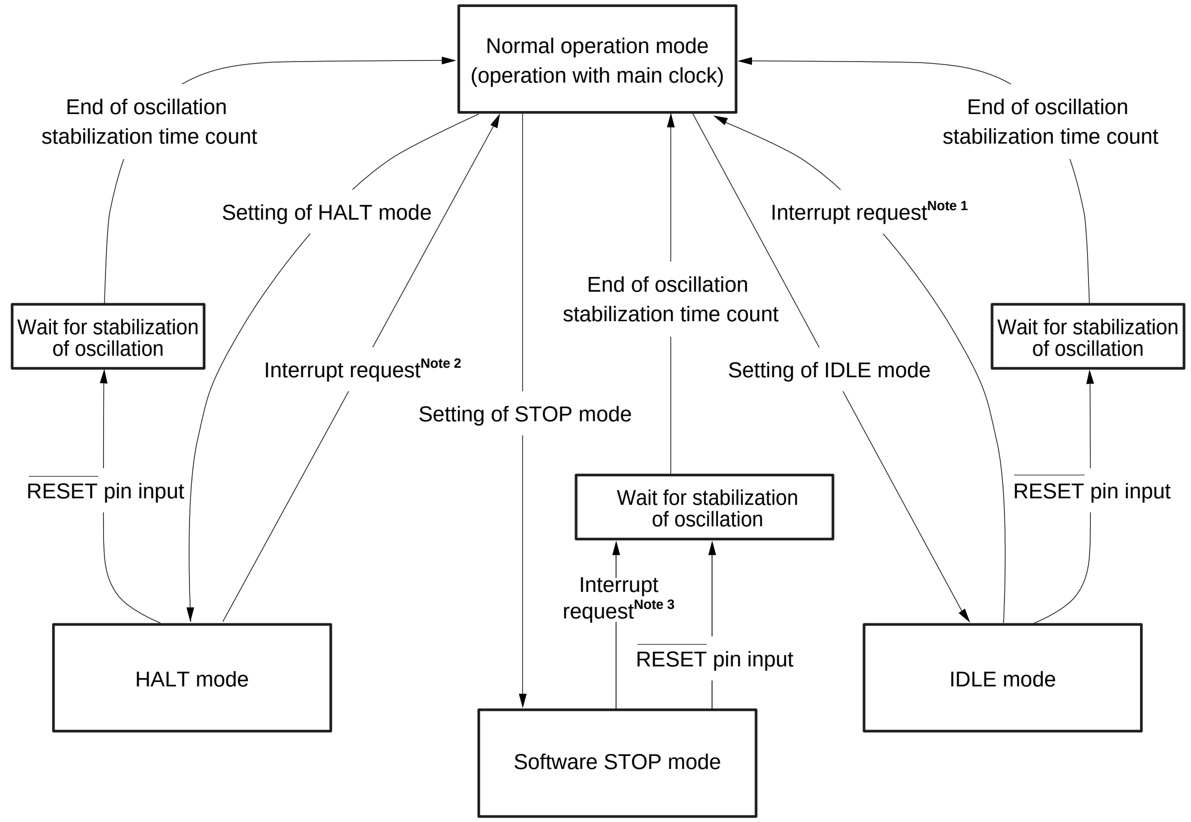
The power consumption of the system can be effectively reduced by using the standby modes in combination and selecting the appropriate mode for the application. The available standby modes are listed in Table 17-1.

Table 17-1. Standby Modes

Mode	Functional Outline
HALT mode	Mode to stop only the operating clock of the CPU
IDLE mode	Mode to stop all the operations of the internal circuit except the oscillator and RTC[Note]
Software STOP mode	Mode to stop all the operations of the internal circuit except the subclock oscillator and RTC[Note]
Subclock operation mode	Mode to use the subclock as the internal system clock
Sub-IDLE mode	Mode to stop all the operations of the internal circuit, except the oscillator and RTC[Note], in the subclock operation mode
Sub-software STOP mode	Mode to stop RTC and all the internal operations of the V850ES/PM1, including the oscillator, in the subclock operation mode

Note When RTC operation is enabled

Figure 17-1. Status Transition

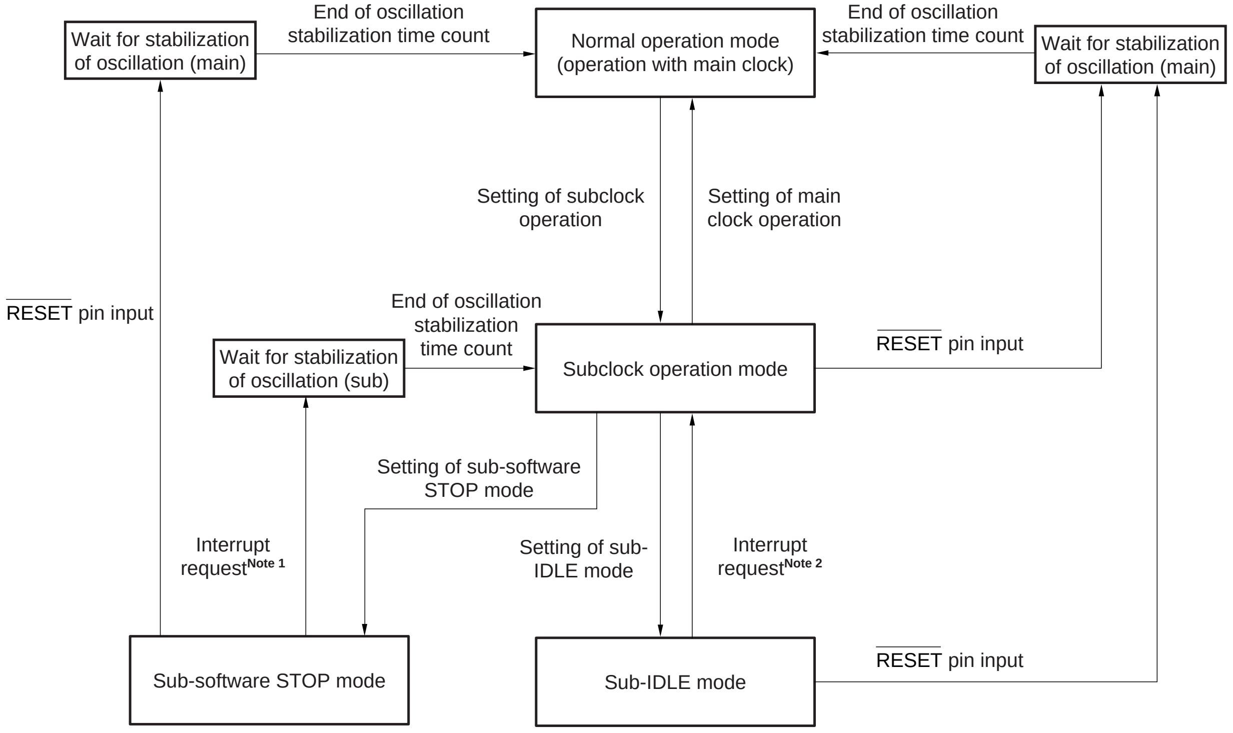

Normal operation mode
(operation with main clock)
End of oscillation stabilization time count
Setting of HALT mode
Interrupt request^Note 2
Wait for stabilization of oscillation
RESET pin input
HALT mode
Setting of STOP mode
End of oscillation stabilization time count
Wait for stabilization of oscillation
Interrupt request^Note 3
RESET pin input
Software STOP mode
Interrupt request^Note 1
Setting of IDLE mode
End of oscillation stabilization time count
Wait for stabilization of oscillation
RESET pin input
IDLE mode

Notes 1. Non-maskable interrupt request (NMI pin input), unmasked external interrupt request (INTP0 to INTP2 pin input), or unmasked internal interrupt request from peripheral functions operable in IDLE mode.

2. Non-maskable interrupt request (NMI pin input) or unmasked maskable interrupt request.

3. Non-maskable interrupt request (NMI pin input), unmasked external interrupt request (INTP0 to INTP2 pin input), or unmasked internal interrupt request from peripheral functions operable in software STOP mode.

Figure 17-2. Status Transition (During Subclock Operation)


Wait for stabilization of oscillation (main)
End of oscillation stabilization time count
Normal operation mode (operation with main clock)
End of oscillation stabilization time count
Wait for stabilization of oscillation (main)
Setting of subclock operation
Setting of main clock operation
RESET pin input
End of oscillation stabilization time count
Wait for stabilization of oscillation (sub)
Subclock operation mode
RESET pin input
Setting of sub-software STOP mode
Interrupt request Note 1
Setting of sub-IDLE mode
Interrupt request Note 2
Sub-software STOP mode
Sub-IDLE mode
RESET pin input

Notes 1. Non-maskable interrupt request (NMI pin input), unmasked external interrupt request (INTP0 to INTP2 pin input), or unmasked internal interrupt request from peripheral functions operable in sub-software STOP mode.

2. Non-maskable interrupt request (NMI pin input), unmasked external interrupt request (INTP0 to INTP2 pin input), or unmasked internal interrupt request from peripheral functions operable in sub-IDLE mode.

17.2 Registers

(1) Power save control register (PSC)

PSC is an 8-bit register that controls the standby function. The STP bit of this register is used to specify the IDLE/software STOP mode. The PSC register is a special register (refer to **3.4.8 Special registers**). Data can be written to this register only in a specific sequence so that its contents are not rewritten by mistake due to a program hang-up.

This register can be read or written in 8-bit or 1-bit units.

After reset, this register is set to 00H.

After reset: 00H R/W Address: FFFFF1FEH

	7	6	<5>	<4>	3	2	<1>	0
PSC	0	0	NMI0M	INTM	0	0	STP[Note]	0

NMI0M	Standby mode release control by NMI pin input
0	Release enabled
1	Release disabled

INTM	Standby mode release control by maskable interrupt request
0	Release enabled
1	Release disabled

STP[Note]	Standby mode setting
0	Normal operation mode
1	Standby mode[Note]

★ **Note** When setting software STOP mode, set the PSMR.PSM bit and then set the STP bit to 1.

(2) Power save mode register (PSMR)

PSMR is an 8-bit register that controls the operation status and clock operation in the power save mode.

This register can be read or written in 8-bit or 1-bit units.

After reset, this register is set to 00H.

After reset: 00H R/W Address: FFFFF820H

	7	6	5	4	3	2	1	<0>
PSMR	0	0	0	0	0	0	0	PSM

PSM	Specifies operation in software standby mode
0	IDLE mode
1	Software STOP mode

Cautions 1. Be sure to clear bits 1 to 7 of the PSMR register to 0.

2. The PSM bit is valid only when the PSC.STP bit is set to 1.

(3) Oscillation stabilization time select register (OSTS)

The OSTS register controls the wait time until the oscillation stabilizes after the software STOP/sub-software STOP mode is released.

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

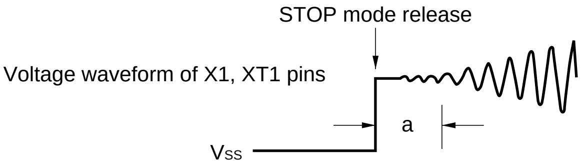
This register is set to 04H after reset.

After reset: 04H R/W Address: FFFFF6C0H

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time			
				f_X		f_{XT}
				20 MHz	10 MHz	32.768 kHz
0	0	0	$2^{14}/f_X$, $2^{14}/f_{XT}$	0.819 ms	1.638 ms	500 ms
0	0	1	$2^{16}/f_X$, $2^{16}/f_{XT}$	3.277 ms	6.554 ms	2.0 s
0	1	0	$2^{17}/f_X$, $2^{17}/f_{XT}$	6.554 ms	13.11 ms	4.0 s
0	1	1	$2^{18}/f_X$, $2^{18}/f_{XT}$	13.11 ms	26.21 ms	8.0 s
1	0	0	$2^{19}/f_X$, $2^{19}/f_{XT}$	26.21 ms	52.43 ms	16 s
1	0	1	$2^{20}/f_X$, $2^{20}/f_{XT}$	52.43 ms	104.9 ms	32 s
1	1	0	$2^{21}/f_X$, $2^{21}/f_{XT}$	104.9 ms	209.7 ms	64 s
1	1	1	$2^{22}/f_X$, $2^{22}/f_{XT}$	209.7 ms	419.4 ms	128 s

Cautions 1. The wait time following release of the software STOP mode does not include the time until the clock oscillation starts (“a” in the figure below) following release of the software STOP mode, regardless of whether the STOP mode is released through $\overline{\text{RESET}}$ input or the occurrence of an interrupt request signal.



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

3. The oscillation stabilization time following reset release is $2^{19}/f_X$ (because the initial value of the OSTS register = 04H).

Remark f_X = Main clock oscillation frequency

17.3 HALT Mode

17.3.1 Setting and operation status

The HALT mode is set when a dedicated instruction (HALT) is executed in the normal operation mode.

In the HALT mode, the clock oscillator continues operating. Only clock supply to the CPU is stopped; clock supply to the other on-chip peripheral functions continues.

As a result, program execution is stopped, and the internal RAM retains the contents before the HALT mode was set. The on-chip peripheral functions that are independent of instruction processing by the CPU continue operating.

Table 17-3 shows the operation status in the HALT mode.

The average power consumption of the system can be reduced by using the HALT mode in combination with the normal operation mode for intermittent operation.

Cautions 1. Insert five or more NOP instructions after the HALT instruction.

★ **2. If the HALT instruction is executed with an interrupt request signal held pending, the system shifts to the HALT mode, but the HALT mode is immediately released by the pending interrupt request.**

17.3.2 Releasing HALT mode

The HALT mode is released by a non-maskable interrupt request, an unmasked maskable interrupt request, and $\overline{\text{RESET}}$ pin input.

After the HALT mode has been released, the normal operation mode is restored.

(1) Releasing HALT mode by non-maskable interrupt request or unmasked maskable interrupt request

The HALT mode is released by a non-maskable interrupt request or an unmasked maskable interrupt request, regardless of the priority of the interrupt request. If the HALT mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

(a) If an interrupt request with a priority lower than that of the interrupt request currently being serviced is issued, only the HALT mode is released, and that interrupt request is not acknowledged. The interrupt request itself is retained.

(b) If an interrupt request with a priority higher than that of the interrupt request currently being serviced is issued (including a non-maskable interrupt request), the HALT mode is released and that interrupt request is acknowledged.

Table 17-2. Operation After Releasing HALT Mode by Interrupt Request

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request	Execution branches to the handler address	
Maskable interrupt request	Execution branches to the handler address or the next instruction is executed	The next instruction is executed

(2) Releasing HALT mode by $\overline{\text{RESET}}$ pin input

The same operation as the normal reset operation is performed.

Table 17-3. Operation Status in HALT Mode

Setting of HALT Mode		Operation Status	
Item		When Subclock Is Not Used	When Subclock Is Used
Main clock oscillator		Oscillation enabled	
Subclock oscillator		–	Oscillation enabled
CPU		Stops operation	
Interrupt controller		Operable	
ROM correction		Stops operation	
16-bit timer/event counters (TM00 to TM03, TM10, TM11)		Operable	
8-bit timer/event counters (TM20, TM21)		Operable	
Real-time counter		–	Operable
Watchdog timer		Operable[Note]	
Serial interface	CSI0, CSI1	Operable	
	UART0, UART1	Operable	
A/D converter		Operable	
PWM (PWM0 to PWM3)		Operable	
External bus interface		Refer to **2.2 Pin Status**.	
Port function		Retains status before HALT mode was set.	
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the HALT mode was set.	

★ **Note** Take care to prevent an overflow from occurring because the watchdog timer operates in the HALT mode.

17.4 IDLE Mode

17.4.1 Setting and operation status

The IDLE mode is set by clearing the PSMR.PSM bit to 0 and setting the PSC.STP bit to 1 in the normal operation mode.

In the IDLE mode, the clock oscillator continues operation but clock supply to the CPU and other on-chip peripheral functions stops.

As a result, program execution stops and the contents of the internal RAM before the IDLE mode was set are retained. The CPU and other on-chip peripheral functions stop operating. However, the on-chip peripheral functions that can operate with the subclock or an external clock continue operating.

Table 17-5 shows the operation status in the IDLE mode.

The IDLE mode can reduce the power consumption more than the HALT mode because it stops the operation of the on-chip peripheral functions. The main clock oscillator does not stop, so the normal operation mode can be restored without waiting for the oscillation stabilization time after the IDLE mode has been released, in the same manner as when the HALT mode is released.

Caution Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the IDLE mode.

17.4.2 Releasing IDLE mode

The IDLE mode is released by a non-maskable interrupt request (NMI pin input), unmasked external interrupt request (INTP0 to INTP2 pin input), unmasked internal interrupt request from the peripheral functions operable in the IDLE mode, or $\overline{\text{RESET}}$ input.

After the IDLE mode has been released, the normal operation mode is restored.

(1) Releasing IDLE mode by non-maskable interrupt request or unmasked maskable interrupt request

The IDLE mode is released by a non-maskable interrupt request or an unmasked maskable interrupt request, regardless of the priority of the interrupt request. If the IDLE mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is processed as follows.

(a) If an interrupt request with a priority lower than that of the interrupt request currently being serviced is issued, only the IDLE mode is released, and that interrupt request is not acknowledged. The interrupt request itself is retained.

(b) If an interrupt request with a priority higher than that of the interrupt request currently being serviced is issued (including a non-maskable interrupt request), the IDLE mode is released and that interrupt request is acknowledged.

Table 17-4. Operation After Releasing IDLE Mode by Interrupt Request

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request	Execution branches to the handler address	
Maskable interrupt request	Execution branches to the handler address or the next instruction is executed	The next instruction is executed

(2) Releasing IDLE mode by $\overline{\text{RESET}}$ pin input

The same operation as the normal reset operation is performed.

Table 17-5. Operation Status in IDLE Mode

Setting of IDLE Mode		Operation Status	
Item		When Subclock Is Not Used	When Subclock Is Used
Main clock oscillator		Oscillation enabled	
Subclock oscillator		–	Oscillation enabled
CPU		Stops operation	
Interrupt controller		Stops operation (mode releasing request can be acknowledged)	
ROM correction		Stops operation	
16-bit timer/event counters (TM00 to TM03, TM10, TM11)		Stops operation	
8-bit timer/event counters (TM20, TM21)		Stops operation	
Real-time counter		–	Operable
Watchdog timer		Stops operation	
Serial interface	CSI0, CSI1	Operable when $\overline{\text{SCKn}}$ input clock is selected as operation clock (n = 0, 1)	
	UART0, UART1	Stops operation	
A/D converter		Stops operation	
PWM (PWM0 to PWM3)		Stops operation	
External bus interface		Refer to **2.2 Pin Status**.	
Port function		Retains status before IDLE mode was set.	
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the IDLE mode was set.	

17.5 Software STOP Mode

17.5.1 Setting and operation status

The software STOP mode is set when the PSMR.PSM bit is set to 1 and the STP bit of the PSC register is set to 1 in the normal operation mode.

In the software STOP mode, the subclock oscillator continues operating but the main clock oscillator stops. Clock supply to the CPU and the on-chip peripheral functions is stopped.

As a result, program execution is stopped, and the contents of the internal RAM before the software STOP mode was set are retained. The on-chip peripheral functions that operate with the clock oscillated by the subclock oscillator or an external clock continue operating.

Table 17-7 shows the operation status in the software STOP mode.

Because the software STOP stops operation of the main clock oscillator, it reduces the current consumption to a level lower than the IDLE mode. If the subclock oscillator and external clock are not used, the power consumption can be minimized with only leakage current flowing.

Caution Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the software STOP mode.

17.5.2 Releasing software STOP mode

The software STOP mode is released by a non-maskable interrupt request (NMI pin input), unmasked external interrupt request (INTP0 to INTP2 pin input), unmasked internal interrupt request from the peripheral functions operable in the software STOP mode, or $\overline{\text{RESET}}$ pin input.

After the software STOP mode has been released, the normal operation mode is restored after the oscillation stabilization time has been secured.

(1) Releasing software STOP mode by non-maskable interrupt request or unmasked maskable interrupt request

The software STOP mode is released by a non-maskable interrupt request or an unmasked maskable interrupt request, regardless of the priority of the interrupt request. If the software STOP mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

(a) If an interrupt request with a priority lower than that of the interrupt request currently being serviced is issued, only the software STOP mode is released, and that interrupt request is not acknowledged. The interrupt request itself is retained.

(b) If an interrupt request with a priority higher than that of the interrupt request currently being serviced is issued (including a non-maskable interrupt request), the software STOP mode is released and that interrupt request is acknowledged.

Table 17-6. Operation After Releasing Software STOP Mode by Interrupt Request

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request	Execution branches to the handler address	
Maskable interrupt request	Execution branches to the handler address or the next instruction is executed	The next instruction is executed

(2) Releasing software STOP mode by $\overline{\text{RESET}}$ pin input

The same operation as the normal reset operation is performed.

Table 17-7. Operation Status in Software STOP Mode

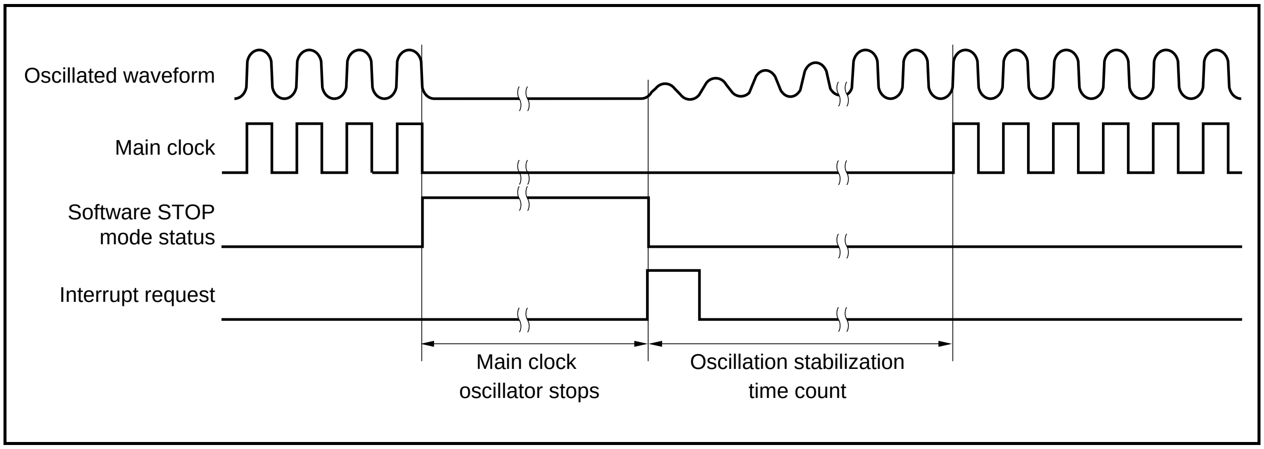
Setting of Software STOP Mode Item		Operation Status	
		When Subclock Is Not Used	When Subclock Is Used
Main clock oscillator		Stops operation	
Subclock oscillator		–	Oscillation enabled
CPU		Stops operation	
Interrupt controller		Stops operation (mode releasing request can be acknowledged)	
ROM correction		Stops operation	
16-bit timer/event counters (TM00 to TM03, TM10, TM11)		Stops operation	
8-bit timer/event counters (TM20, TM21)		Stops operation	
Real-time counter		–	Operable
Watchdog timer		Stops operation	
Serial interface	CSI0, CSI1	Operable when $\overline{\text{SCKn}}$ input clock is selected as operation clock (n = 0, 1)	
	UART0, UART1	Stops operation	
A/D converter		Stops operation	
PWM (PWM0 to PWM3)		Stops operation	
External bus interface		Refer to **2.2 Pin Status**.	
Port function		Retains status before software STOP mode was set.	
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the software STOP mode was set.	

17.6 Securing Oscillation Stabilization Time

When the software STOP mode or sub-software STOP mode is released, only the oscillation stabilization time set by the OSTS register elapses. If the software STOP mode has been released by $\overline{\text{RESET}}$ pin input, however, the reset value of the OSTS register, $2^{19}/f_X$ elapses.

Figure 17-3 shows the operation performed when the software STOP mode is released by an interrupt request.

Figure 17-3. Oscillation Stabilization Time



Caution For details of the OSTS register, refer to 17.2 (3) Oscillation stabilization time select register (OSTS).

17.7 Subclock Operation Mode

17.7.1 Setting and operation status

The subclock operation mode is set when the PCC.CK3 bit is set to 1 in the normal operation mode.

When the subclock operation mode is set, the internal system clock is changed from the main clock to the subclock.

When the PCC.MCK bit is set to 1, the operation of the main clock oscillator is stopped. As a result, the system operates only with the subclock. However, watchdog timer stops counting when subclock operation is started (PCC.CLS bit = 1). (Watchdog timer retains the value before the subclock operation mode was set.)

In the subclock operation mode, the power consumption can be reduced to a level lower than in the normal operation mode because the subclock is used as the internal system clock. In addition, the power consumption can be further reduced to the level of the software STOP mode by stopping the operation of the main system clock oscillator.

Table 17-8 shows the operation status in subclock operation mode.

Cautions 1. When manipulating the CK3 bit, do not change the set values of the PCC.CK1 and PCC.CK0 bits (using a bit manipulation instruction to manipulate the bit is recommended). For details of the PCC register, refer to 6.3 (1) Processor clock control register (PCC).

2. To select the subclock operation and stop the main clock, stop the operations of TM00 to TM03, TM10, TM11, TM20, TM21, UART0, UART1, and PWM0 to PWM3.

3. The watchdog timer stops during the subclock operation. Do not write anything to the WDTM register.

17.7.2 Releasing subclock operation mode

The subclock operation mode is released by $\overline{\text{RESET}}$ pin input when the CK3 bit is cleared to 0. If the main clock is stopped (MCK bit = 1), set the MCK bit to 0, secure the oscillation stabilization time of the main clock by software, and clear the CK3 bit to 0.

The normal operation mode is restored when the subclock operation mode is released.

Caution When manipulating the CK3 bit, do not change the set values of the CK1 and CK0 bits (using a bit manipulation instruction to manipulate the bit is recommended).
For details of the PCC register, refer to 6.3 (1) Processor clock control register (PCC).

★ 17.7.3 Registers to which access is disabled in subclock operation mode

While the CPU is operating on the subclock and a clock is not input to X1 or when the main oscillator is stopped, do not access the following registers in which a wait is generated using an access method that causes a wait. If a wait is generated, only a reset can release the wait.

For details, refer to **3.4.8 (2)**.

Peripheral Function	Register Name	Access Method
Watchdog timer (WDT)	WDTM	Write
16-bit timer/event counters (TM10, TM11)	TM10, TM11	Read
	CC100, CC101, CC110, CC111	Read (in capture mode)
		Write (in compare mode)
	TMC100, TMC110	Write
		Read modify write
PWM	PWMB0 to PWMB3	Write

Table 17-8. Operation Status in Subclock Operation Mode

Setting of Subclock Operation Mode Item		Operation Status	
		When Main Clock Is Oscillating	When Main Clock Is Stopped
Subclock oscillator		Oscillation enabled	
CPU		Operable	
Interrupt controller		Operable	
ROM correction		Operable	
16-bit timer/event counters (TM00 to TM03, TM10, TM11)		Operable	Stops operation
8-bit timer/event counters (TM20, TM21)		Operable	Stops operation
Real-time counter		Operable	
Watchdog timer		Stops operation	
Serial interface	CSI0, CSI1	Operable	Operable when $\overline{\text{SCKn}}$ input clock is selected as operation clock (n = 0, 1)[Note]
	UART0, UART1	Operable	Stops operation
A/D converter		Operable	Stops operation
PWM (PWM0 to PWM3)		Operable	Stops operation
External bus interface		Operable[Note]	
Port function		Settable	
Internal data		Settable	

★ **Note** Operation when $V_{DD} \leq 2.7$ V is not guaranteed.

17.8 Sub-IDLE Mode

17.8.1 Setting and operation status

The sub-IDLE mode is set when the PSMR.PSM bit is cleared to 0 and the PSC.STP bit is set to 1 in the subclock operation mode.

In this mode, the clock oscillator continues operation but clock supply to the CPU and the other on-chip peripheral functions is stopped.

As a result, program execution is stopped and the contents of the internal RAM before the sub-IDLE mode was set are retained. The CPU and the other on-chip peripheral functions are stopped. However, the on-chip peripheral functions that can operate with the subclock or an external clock continue operating.

Because the sub-IDLE mode stops operation of the CPU and other on-chip peripheral functions, it can reduce the power consumption more than the subclock operation mode. If the sub-IDLE mode is set after the main clock has been stopped, the current consumption can be reduced to a level as low as that in the software STOP mode.

Table 17-10 shows the operation status in the sub-IDLE mode.

Caution Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the sub-IDLE mode.

17.8.2 Releasing sub-IDLE mode

The sub-IDLE mode is released by a non-maskable interrupt request (NMI pin input), unmasked external interrupt request (INTP0 to INTP2 pin input), unmasked internal interrupt request from the peripheral functions operable in the sub-IDLE mode, or $\overline{\text{RESET}}$ pin input.

When the sub-IDLE mode is released by an interrupt request, the subclock operation mode is set. If it is released by $\overline{\text{RESET}}$ pin input, the normal operation mode is restored.

(1) Releasing sub-IDLE mode by non-maskable interrupt request or unmasked maskable interrupt request

The sub-IDLE mode is released by a non-maskable interrupt request or an unmasked maskable interrupt request, regardless of the priority of the interrupt request. If the sub-IDLE mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

(a) If an interrupt request with a priority lower than that of the interrupt request currently being serviced is issued, only the sub-IDLE mode is released, and that interrupt request is not acknowledged. The interrupt request itself is retained.

(b) If an interrupt request with a priority higher than that of the interrupt request currently being serviced is issued (including a non-maskable interrupt request), the sub-IDLE mode is released and that interrupt request is acknowledged.

Table 17-9. Operation After Releasing Sub-IDLE Mode by Interrupt Request

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request	Execution branches to the handler address	
Maskable interrupt request	Execution branches to the handler address or the next instruction is executed	The next instruction is executed

(2) Releasing sub-IDLE mode by $\overline{\text{RESET}}$ pin input

The same operation as the normal reset operation is performed.

Table 17-10. Operation Status in Sub-IDLE Mode

Setting of Sub-IDLE Mode Item		Operation Status	
		When Main Clock Is Oscillating	When Main Clock Is Stopped
Subclock oscillator		Oscillation enabled	
CPU		Stops operation	
Interrupt controller		Stops operation (mode releasing request can be acknowledged)	
ROM correction		Stops operation	
16-bit timer/event counters (TM00 to TM03, TM10, TM11)		Stops operation	
8-bit timer/event counters (TM20, TM21)		Stops operation	
Real-time counter		Operable	
Watchdog timer		Stops operation	
Serial interface	CSI0, CSI1	Operable when $\overline{\text{SCKn}}$ input clock is selected as operation clock (n = 0, 1)	
	UART0, UART1	Stops operation	
A/D converter		Stops operation	
PWM (PWM0 to PWM3)		Stops operation	
External bus interface		Refer to **2.2 Pin Status**.	
Port function		Retains status before sub-IDLE mode was set.	
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the sub-IDLE mode was set.	

17.9 Sub-Software STOP Mode

17.9.1 Setting and operation status

The sub-software STOP mode is set when the PSMR.PSM bit is set to 1 and the PSC.STP bit is set to 1 in the subclock operation mode.

In the sub-software STOP mode, the subclock oscillator and main clock oscillator are stopped. Therefore, clock supply to the CPU and other on-chip peripheral functions is stopped.

Table 17-12 shows the operation status in the sub-software STOP mode.

Caution Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the sub-software STOP mode.

17.9.2 Releasing sub-software STOP mode

The sub-software STOP mode is released by a non-maskable interrupt request (NMI pin input), unmasked external interrupt request (INTP0 to INTP2 pin input), unmasked internal interrupt request from the peripheral functions operable in the sub-software STOP mode, or $\overline{\text{RESET}}$ pin input.

When the sub-software STOP mode is released by an interrupt request, the subclock operation mode is set. If it is released by $\overline{\text{RESET}}$ pin input, the normal operation mode is restored.

(1) Releasing sub-software STOP mode by non-maskable interrupt request or unmasked maskable interrupt request

The sub-software STOP mode is released by a non-maskable interrupt request or an unmasked maskable interrupt request, regardless of the priority of the interrupt request. If the sub-software STOP mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

(a) If an interrupt request with a priority lower than that of the interrupt request currently being serviced is issued, only the sub-software STOP mode is released, and that interrupt request is not acknowledged. The interrupt request itself is retained.

(b) If an interrupt request with a priority higher than that of the interrupt request currently being serviced is issued (including a non-maskable interrupt request), the sub-software STOP mode is released and that interrupt request is acknowledged.

Table 17-11. Operation After Releasing Sub-Software STOP Mode by Interrupt Request

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request	Execution branches to the handler address	
Maskable interrupt request	Execution branches to the handler address or the next instruction is executed	The next instruction is executed

(2) Releasing sub-IDLE mode by $\overline{\text{RESET}}$ pin input

The same operation as the normal reset operation is performed.

Table 17-12. Operation Status in Sub-Software STOP Mode

Item		Operation Status
Main clock oscillator		Stops operation
Subclock oscillator		Stops operation
CPU		Stops operation
Interrupt controller		Stops operation (mode releasing request can be acknowledged)
ROM correction		Stops operation
16-bit timer/event counters (TM00 to TM03, TM10, TM11)		Stops operation
8-bit timer/event counters (TM20, TM21)		Stops operation
Real-time counter		Stops operation
Watchdog timer		Stops operation
Serial interface	CSI0, CSI1	Operable when $\overline{\text{SCKn}}$ input clock is selected as operation clock (n = 0, 1)
	UART0, UART1	Stops operation
A/D converter		Stops operation
PWM (PWM0 to PWM3)		Stops operation
External bus interface		Power supply stopped
Port function		Retains status before sub-software STOP mode was set.
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the sub-software STOP mode was set.

CHAPTER 18 RESET FUNCTION

18.1 Overview

The following reset functions are available.

- Reset function by $\overline{\text{RESET}}$ pin input
- Reset function by WDT overflow (WDTRES)

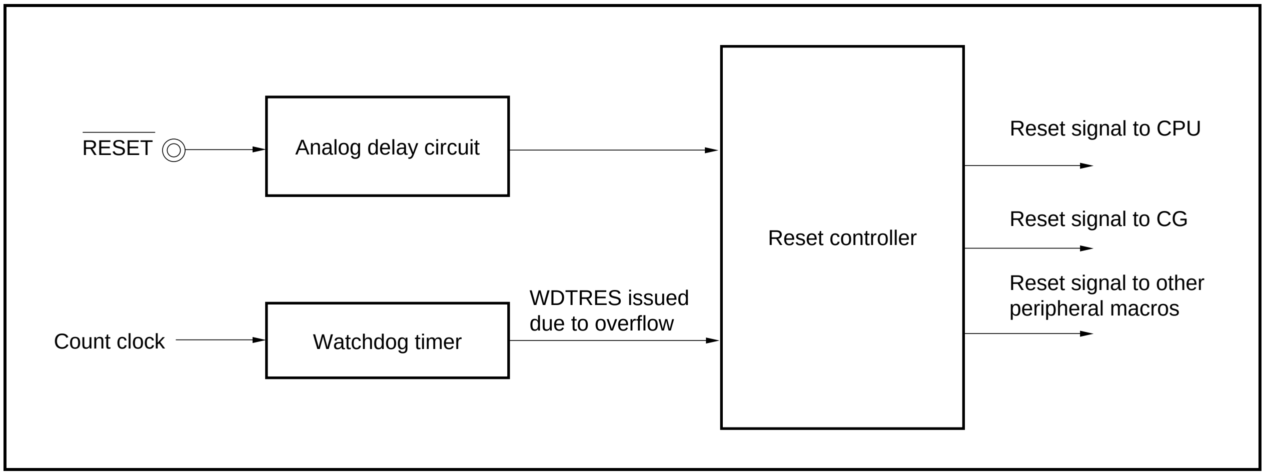
If the $\overline{\text{RESET}}$ pin goes high, the reset status is released, and the CPU starts executing the program. Initialize the contents of each register in the program as necessary.

The $\overline{\text{RESET}}$ pin has a noise eliminator that operates by analog delay to prevent malfunction caused by noise.

A flag (WRESF) that detects occurrence of reset because of overflow of the WDT is also provided. This flag identifies whether reset is effected by $\overline{\text{RESET}}$ pin input or overflow of the WDT during processing after the reset has been released.

18.2 Configuration

Figure 18-1. Reset Block Diagram



18.3 Register

(1) WDT reset status register (WDRES)

WDRES is an 8-bit register that indicates the status of WDTRES, and can be read or written by using an 8-bit or 1-bit manipulation instruction.

To write the WDRES register, a specific sequence using PRCDM as a command register is required. If the register is written in an illegal sequence, writing is invalid and the protect error flag (the SYS.PRERR bit) is set to 1, and nothing is written to the register.

This register is set to 00H by $\overline{\text{RESET}}$ pin input, and is set to 01H when the WDTRES signal is generated.

After reset: Undefined R/W Address: FFFFF82AH

	7	6	5	4	3	2	1	<0>
WDRES	0	0	0	0	0	0	0	WRESF

WRESF	WDTRES detection flag
0	WDTRES did not occur
1	WDTRES occurred

Setting (1) condition: Reset by overflow of watchdog timer (WDT)
Clearing (0) condition: Writing “0” by instruction or $\overline{\text{RESET}}$ pin input.
Only “0” can be written to the WRESF bit.

Caution Write “0” to the WRESF bit after confirming (reading) that the WRESF bit is 1 to avoid a conflict with setting the flag.

Remark The WRESF bit can be read or written, but it can only be cleared by writing “0”. “1” cannot be written to it.

18.4 Operation

The system is reset, initializing each hardware unit, when a low level is input to the $\overline{\text{RESET}}$ pin or by WDT overflow (WDTRES signal)[Note].

If the $\overline{\text{RESET}}$ pin goes high or if the WDTRES signal is received, the reset status is released.

If the reset status is released by $\overline{\text{RESET}}$ pin input, the oscillation stabilization time elapses (reset value of OSTS register: $2^{19}/f_X$) and then the CPU starts program execution.

If the reset status is released by the WDTRES signal, the oscillation stabilization time is not inserted because the main system clock oscillator does not stop.

Note Reset by WDT overflow (WDTRES signal) is valid only when the WDTM.WDTM4 and WDTM.WDTM3 bits are set to “11” (refer to **11.3 (2)**).

Table 18-1. Hardware Status on $\overline{\text{RESET}}$ Pin Input

Item	During Reset	After Reset
Main clock oscillator (f_X)	Oscillation stops (f_X = 0 level).	Oscillation starts
Subclock oscillator (f_{XT})	Oscillation can continue without effect from reset[Note 1].	
Peripheral clock (f_X to f_X/1,024), internal system clock (f_{CLK}), CPU clock (f_{CPU})	Operation stops	Operation starts. However, operation stops during oscillation stabilization time count.
WDT clock (f_{XW})	Operation stops	Operation starts[Note 2]
Internal RAM	Undefined if power-on reset occurs or writing data to RAM and reset conflict (data loss). Otherwise, retains values immediately before reset input.	
I/O lines (ports)	High impedance	
On-chip peripheral I/O registers	Initialized to specified status	
Real-time counter	Operation can be started[Note 3]	
Other on-chip peripheral functions	Operation stops	Operation can be started

Notes 1. The on-chip feedback resistor is “connected” by default (refer to **6.3 (1) Processor clock control register (PCC)**).

★ **2.** The WDT clock is in the initial status (interval timer mode).

★ **3.** If the subclock is supplied, therefore, the real-time counter performs a count operation on the subclock after reset.

Table 18-2. Hardware Status on Occurrence of WDTRES

Item	During Reset	After Reset
Main clock oscillator (f_X)	Oscillation continues[Note 1]	
Subclock oscillator (f_{XT})	Oscillation can continue without effect from reset[Note 1].	
Peripheral clock (f_X to f_X/1,024), internal system clock (f_{CLK}), CPU clock (f_{CPU})	Operation stops	Operation starts
WDT clock (f_{XW})	Operation continues	
Internal RAM	Undefined if writing data to RAM and reset conflict (data loss). Otherwise, retains values immediately before reset input.	
I/O lines (ports)	High impedance	
On-chip peripheral I/O registers	Initialized to specified status	
Real-time counter	Operation continues[Note 2]	
Other on-chip peripheral functions	Operation stops	Operation can be started

Notes 1. The on-chip feedback resistor is "connected" by default (refer to **6.3 (1) Processor clock control register (PCC)**).

2. Reset sets the internal peripheral I/O register of the real-time counter so that its count operation by subclock (f_{XT}) is enabled. If the subclock is supplied, therefore, the real-time counter performs a count operation on the subclock after reset.

Figure 18-2. Hardware Status on $\overline{\text{RESET}}$ Input

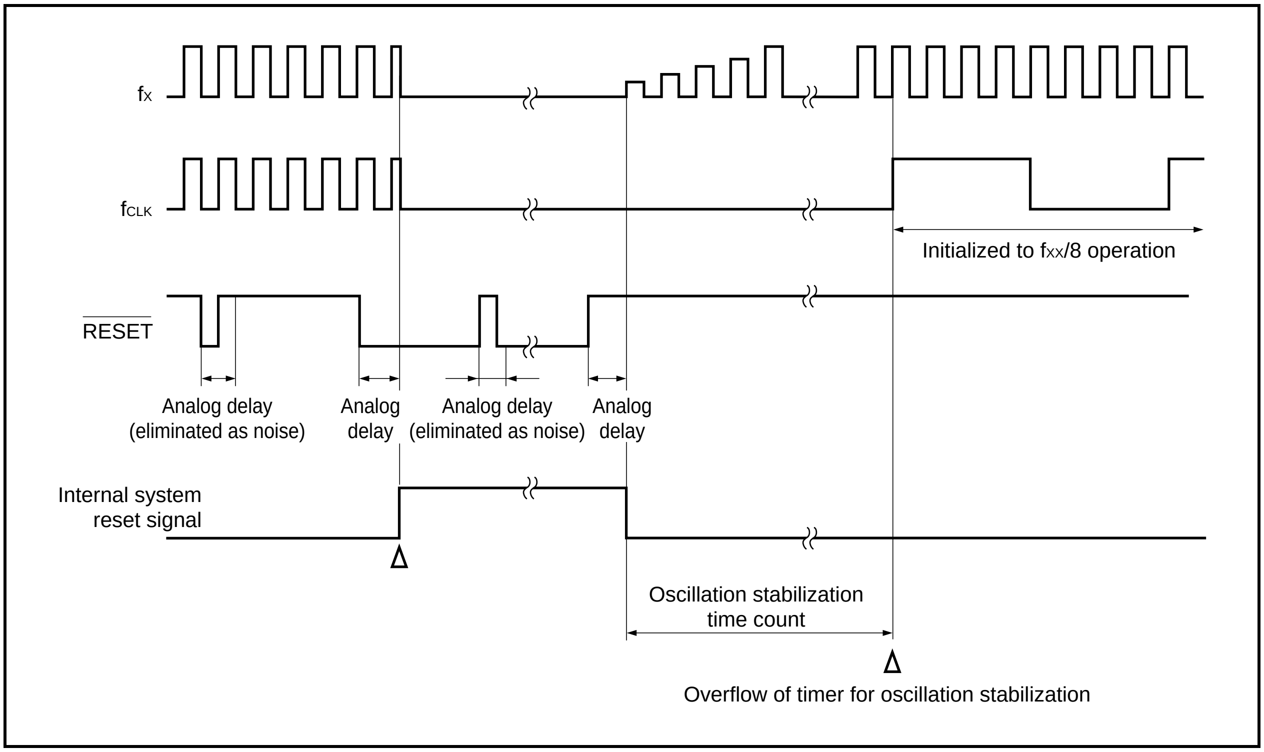
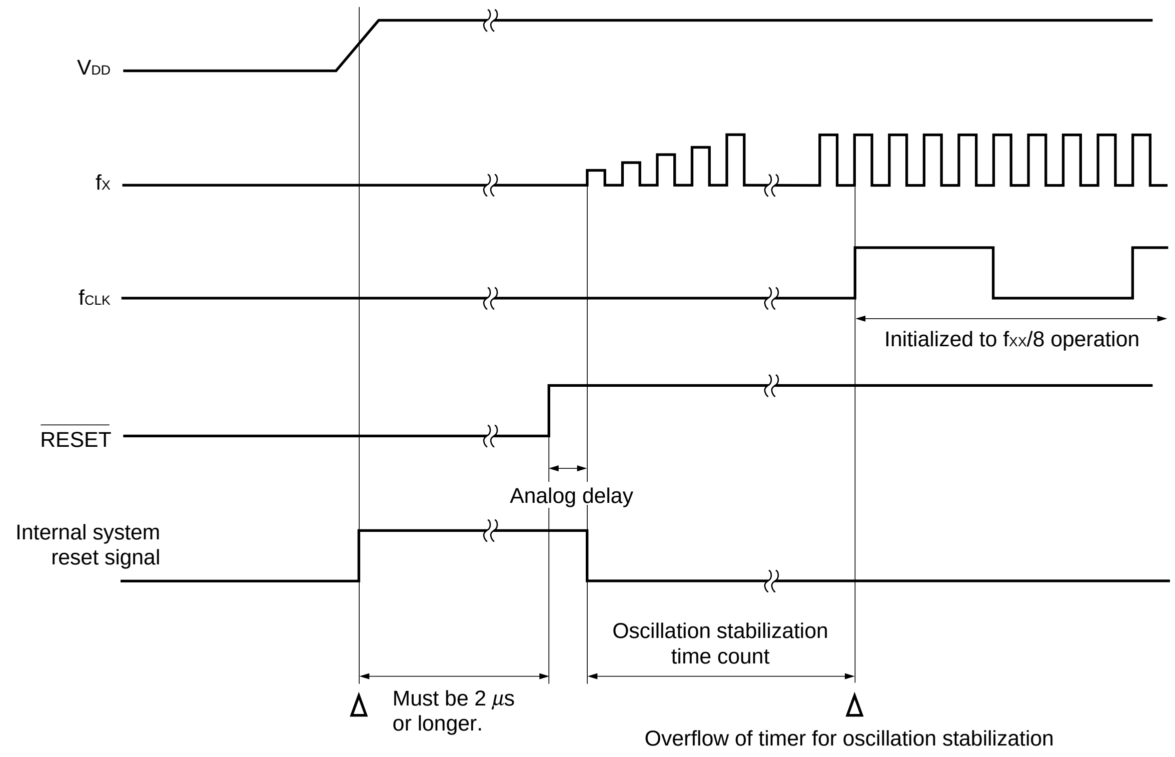

fX
fCLK
Initialized to fXX/8 operation
RESET
Analog delay (eliminated as noise)
Analog delay
Analog delay (eliminated as noise)
Analog delay
Internal system reset signal
Oscillation stabilization time count
Overflow of timer for oscillation stabilization

Figure 18-3. Operation on Power Application


V_{DD}
f_X
f_{CLK}
Initialized to $f_{XX}/8$ operation
$\overline{RESET}$
Analog delay
Internal system reset signal
Oscillation stabilization time count
Must be 2 μs or longer.
Overflow of timer for oscillation stabilization

Caution Turn on V_{DD} (power supply for internal circuits) and EV_{DD} (power supply for external circuits) in that order. If EV_{DD} is turned on first, the status of each pin is undefined until the reset function operates. Consequently, input and output of a port may conflict.

CHAPTER 19 ROM CORRECTION FUNCTION

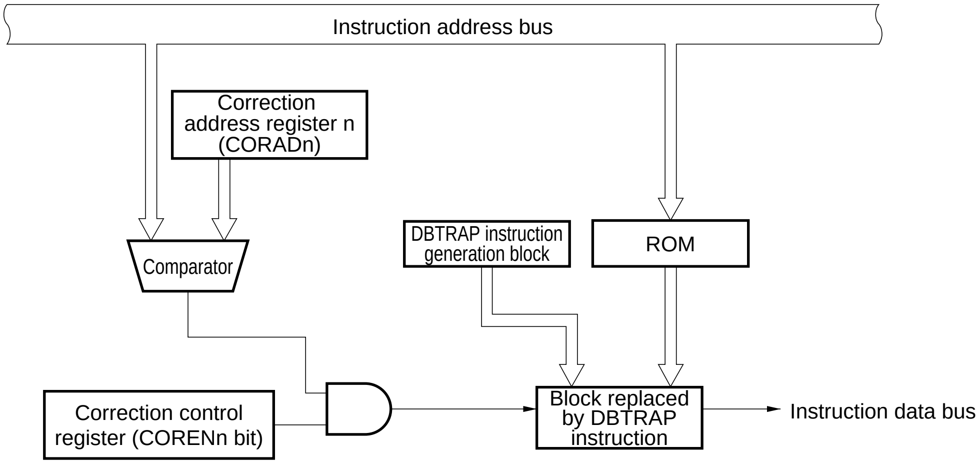
19.1 Overview

The ROM correction function is used to replace part of the program in the mask ROM with the program of an external memory or the internal RAM.

By using this function, instruction bugs found in the mask ROM can be corrected.

Up to four addresses can be specified for correction.

Figure 19-1. Block Diagram of ROM Correction


Instruction address bus
Correction address register n (CORADn)
Comparator
DBTRAP instruction generation block
ROM
Correction control register (CORENn bit)
Block replaced by DBTRAP instruction
Instruction data bus

Remark n = 0 to 3

19.2 Registers

19.2.1 Correction address registers 0 to 3 (CORAD0 to CORAD3)

CORAD0 to CORAD3 are used to set the first address of the program to be corrected.

The program can be corrected at up to four places because four CORADn registers are provided (n = 0 to 3).

The CORADn register can only be read or written in 32-bit units.

If the higher 16 bits of the CORADn register are used as the CORADnH register, and the lower 16 bits as the CORADnL register, these registers can be read or written in 16-bit units.

Set correction addresses within the range of 0000000H to 003FFFEH.

Fix bits 0 and 18 to 31 to 0.

These registers are set to 00000000H after reset.

After reset: 00000000H R/W Address: Refer to **Table 19-1**.

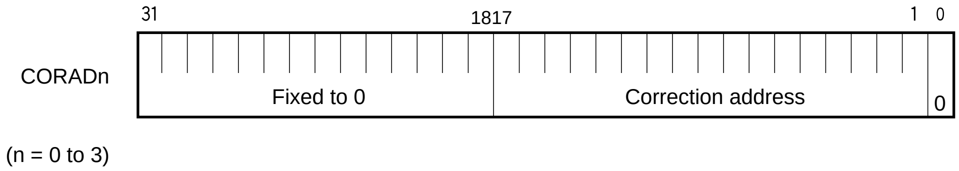


Table 19-1. Address of CORADn

FFFFF840H		CORAD0	FFFFF848H		CORAD2
	FFFFF840H	CORAD0L		FFFFF848H	CORAD2L
	FFFFF842H	CORAD0H		FFFFF84AH	CORAD2H
FFFFF844H		CORAD1	FFFFF84CH		CORAD3
	FFFFF844H	CORAD1L		FFFFF84CH	CORAD3L
	FFFFF846H	CORAD1H		FFFFF84EH	CORAD3H

19.2.2 Correction control register (CORCN)

CORCN is a register that disables or enables the correction operation at the address specified by the CORADn register (n = 0 to 3).

Each channel can be enabled or disabled by this register.

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

This register is set to 00H after reset.

After reset: 00H R/W Address: FFFFF880H

	7	6	5	4	<3>	<2>	<1>	<0>
CORCN	0	0	0	0	COREN3	COREN2	COREN1	COREN0

CORENn	Enables/disables correction operation
0	Disabled
1	Enabled

Remark n = 0 to 3

Table 19-2. Correspondence Between CORCN Register Bits and CORADn Registers

CORCN Register Bit	Corresponding CORADn Register
COREN3	CORAD3
COREN2	CORAD2
COREN1	CORAD1
COREN0	CORAD0

19.3 ROM Correction Operation and Program Flow

<1> If the address to be corrected and the fetch address of the internal ROM match, the fetch code is replaced by the DBTRAP instruction.

<2> When the DBTRAP instruction is executed, execution branches to address 00000060H.

<3> Software processing after branching causes the result of ROM correction to be judged (the fetch address and ROM correction operation are confirmed) and execution to branch to the correction software.

<4> After the correction software has been executed, the return address is set, and return processing is started by the DBRET instruction.

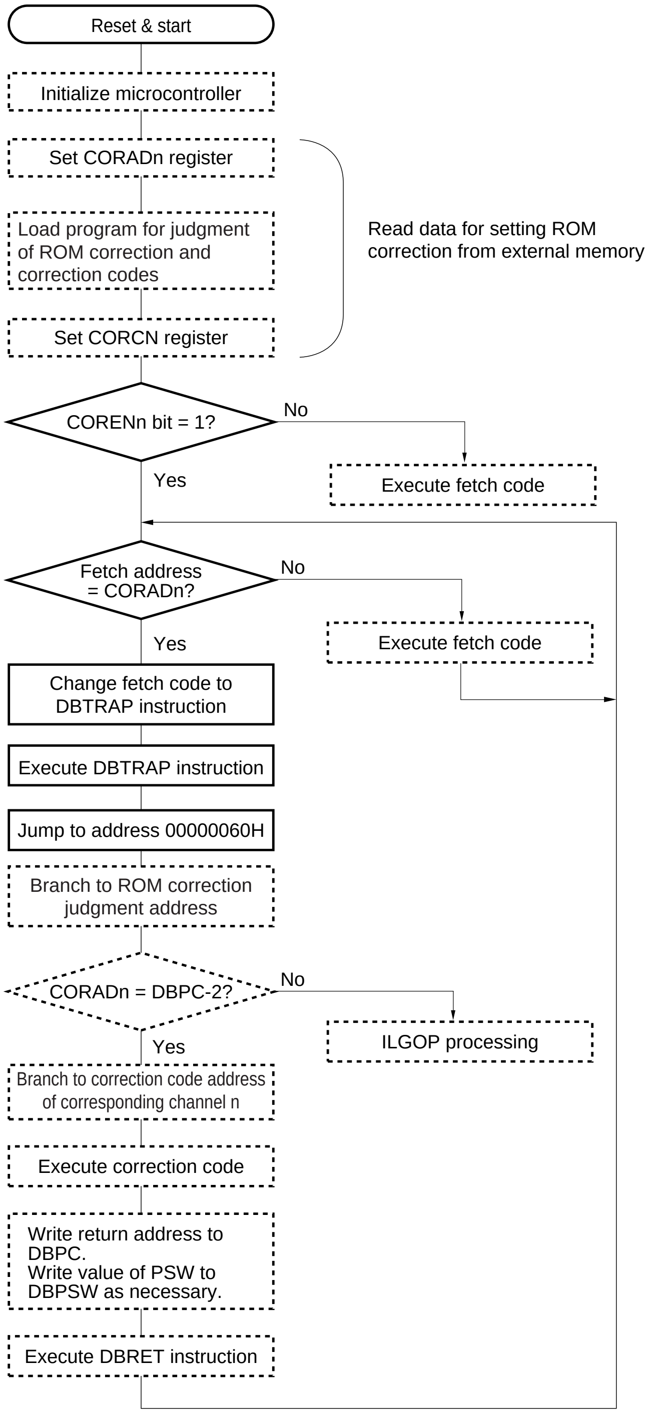
Cautions 1. The software that performs <3> and <4> must be executed in the internal ROM/RAM.

2. When setting an address to be corrected to the CORADn register, clear the higher bits to 0 in accordance with the capacity of the internal ROM.

3. The ROM correction function cannot be used to correct the data of the internal ROM. It can only be used to correct instruction codes. If ROM correction is used to correct data, that data is replaced with the DBTRAP instruction code.

★

Figure 19-2. ROM Correction Operation and Program Flow


Reset & start
Initialize microcontroller
Set CORADn register
Load program for judgment of ROM correction and correction codes
Set CORCN register
Read data for setting ROM correction from external memory
CORENn bit = 1?
No
Execute fetch code
Yes
Fetch address = CORADn?
No
Execute fetch code
Yes
Change fetch code to DBTRAP instruction
Execute DBTRAP instruction
Jump to address 00000060H
Branch to ROM correction judgment address
CORADn = DBPC-2?
No
ILGOP processing
Yes
Branch to correction code address of corresponding channel n
Execute correction code
Write return address to DBPC.
Write value of PSW to DBPSW as necessary.
Execute DBRET instruction

Remarks 1. ⬚ : Processing by user program (software)

□ : Processing by ROM correction (hardware)

2. n = 0 to 3

CHAPTER 20 ELECTRICAL SPECIFICATIONS

Absolute maximum ratings (T_A = 25°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.5 to +4.6	V
	AV_{DD}		–0.5 to +4.6	V
	EV_{DD}		–0.5 to +4.6	V
	AV_{SS}		–0.5 to +0.5	V
	EV_{SS}		–0.5 to +0.5	V
Input voltage	V_I	Pins other than X1, X2	–0.5 to EV_{DD} + 0.5[Note]	V
Clock input voltage	V_K	X1, X2, V_{DD} = 2.7 to 3.6 V	–0.5 to V_{DD} + 0.5[Note]	V
Analog input voltage	V_{IAN}		–0.5 to AV_{DD} + 0.5[Note]	V
Analog reference voltage	AV_{REF}	AV_{REFIN}	–0.5 to AV_{DD} + 0.5[Note]	V
Output current, low	I_{OL}	Per pin	4	mA
		Total for P0, P4, and PCS	35	mA
		Total for P1 and P3	35	mA
		Total for P2 and P9	35	mA
		Total for PCM, PCT, PDL, and PDH	35	mA
Output current, high	I_{OH}	Per pin	–4	mA
		Total for P0, P4, and PCS	–35	mA
		Total for P1 and P3	–35	mA
		Total for P2 and P9	–35	mA
		Total for PCM, PCT, PDL, and PDH	–35	mA
Output voltage	V_O	V_{DD} = 2.7 to 3.6 V	–0.5 to V_{DD} + 0.5[Note]	V
Operating ambient temperature	T_A		–40 to +85	°C
Storage temperature	T_{stg}		–65 to +150	°C

Note Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

Cautions 1. Do not directly connect the output (or I/O) pins of IC products to each other, or to V_{DD}, V_{CC}, and GND. Open-drain pins or open-collector pins, however, can be directly connected to each other. Direct connection of the output pins between an IC product and an external circuit is possible, if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.

2. 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.

The ratings and conditions indicated for DC characteristics and AC characteristics represent the quality assurance range during normal operation.

Capacitance (T_A = 25°C, V_{DD} = AV_{DD} = EV_{DD} = V_{SS} = AV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	f_X = 1 MHz Unmeasured pins returned to 0 V			15	pF
I/O capacitance	C_{IO}				15	pF
Output capacitance	C_O				15	pF

Operating conditions (T_A = –40 to +85°C, V_{SS} = AV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Internal system clock frequency	f_{CLK}	@V_{DD} = AV_{DD} = EV_{DD} = 3.0 to 3.6 V, operation with main clock	0.25 Note 1		20 Note 2	MHz
		@V_{DD} = AV_{DD} = EV_{DD} = 2.7 to 3.6 V, operation with main clock	0.25 Note 1		10 Note 2	MHz
		@V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 3.6 V, operation with subclock		32.768		kHz

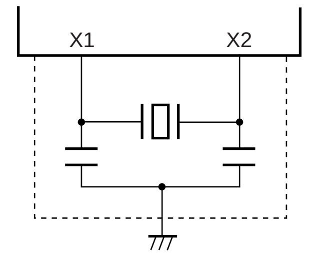
Notes 1. Main clock frequency: MIN. value of f_{XX} condition is value of f_{XX} divided by 8.

2. Main clock frequency: MAX. value of f_{XX} condition is undivided value of f_{XX}.

Recommended oscillator

(1) Main clock oscillator (T_A = –40 to +85°C)

(a) Connection of ceramic resonator or crystal resonator



Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency	f_X (f_{XX})	V_{DD} = 2.7 to 3.6 V	2		20	MHz
Oscillation stabilization time		Upon reset release		$2^{19}/f_X$		s
		Upon STOP mode release		**Note**		s

Note The value differs depending on the setting of the OSTS register.
For details, refer to **17.2 (3) Oscillation stabilization time select register (OSTS)**.

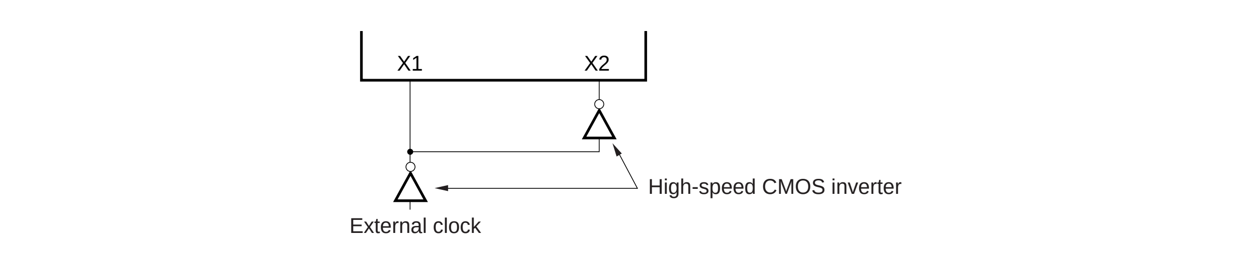
Cautions 1. When using the main clock oscillator, keep the following points in mind in wiring the area enclosed by the broken lines in the above figure to avoid an adverse effect from wiring capacitance.

- **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.**

2. Ensure that the duty of the oscillation waveform is within 5.5:4.5.

3. For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

(2) External clock input (T_A = –40 to +85°C)



★

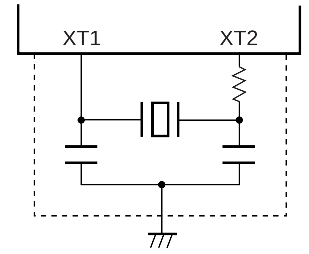
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency	f_X (f_{XX})	V_{DD} = 2.7 to 3.6 V	2		20	MHz

Cautions 1. Thoroughly evaluate matching of the μPD703228 and the high-speed CMOS inverter.

2. Connect the high-speed CMOS inverter as close as possible to the X1 and X2 pins.

(3) Subclock oscillator (T_A = –40 to +85°C)

(a) Connection of crystal resonator



	Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
	Oscillation frequency	f_{XT}	V_{DD} = 2.2 to 3.6 V	32	32.768	35	kHz
★	Oscillation stabilization time		When reset is released		10		ms
			When sub-STOP mode is released		**Note**		s

Note The value differs depending on the setting of the OSTS register.
For details, refer to **17.2 (3) Oscillation stabilization time select register (OSTS)**.

Cautions 1. Inputting an external clock to the subclock oscillator is prohibited.

2. When using the subclock oscillator, keep the following points in mind in wiring the area enclosed by the broken lines in the above figure 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.**

3. Ensure that the duty of the oscillation waveform is within 5.5:4.5.

4. Thoroughly evaluate matching of the μPD703228 and the resonator.

DC characteristics 1

★ **(a) T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	**Note 1**	$0.7EV_{DD}$		EV_{DD}	V
	V_{IH2}	**Note 2**	$0.8EV_{DD}$		EV_{DD}	V
	V_{IH3}	X1, X2	$0.8V_{DD}$		V_{DD}	V
Input voltage, low	V_{IL1}	**Note 1**	EV_{SS}		$0.3EV_{DD}$	V
	V_{IL2}	**Note 2**	EV_{SS}		$0.2EV_{DD}$	V
	V_{IL3}	X1, X2	V_{SS}		$0.2V_{DD}$	V
Output voltage, high	V_{OH1}	I_{OH} = –100 μA	EV_{DD} – 0.5			V
Output voltage, low	V_{OL1}	I_{OL} = 100 μA			0.4	V

Notes 1. P10 to P13, P20, P21, P31, P33, P41, P44, P90 to P97, PCM0, PCM1, PCS0 to PCS2, PCT0, PCT1, PCT4, PDH0 to PDH2, PDL0 to PDL15 (and their alternate-function pins)

2. $\overline{\text{RESET}}$, P00 to P03, P14, P30, P32, P34 to P36, P40, P42, P43, P45, P46, P98 to P915 (and their alternate-function pins)

★ **(b) T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 3.6 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	**Note 1**	$0.75EV_{DD}$		EV_{DD}	V
	V_{IH2}	**Note 2**	$0.85EV_{DD}$		EV_{DD}	V
	V_{IH3}	X1, X2	$0.85V_{DD}$		V_{DD}	V
Input voltage, low	V_{IL1}	**Note 1**	EV_{SS}		$0.25EV_{DD}$	V
	V_{IL2}	**Note 2**	EV_{SS}		$0.15EV_{DD}$	V
	V_{IL3}	X1, X2	V_{SS}		$0.15V_{DD}$	V
Output voltage, high	V_{OH1}	I_{OH} = –100 μA	EV_{DD} – 0.5			V
Output voltage, low	V_{OL1}	I_{OL} = 100 μA			0.4	V

Notes 1. P10 to P13, P20, P21, P31, P33, P41, P44, P90 to P97, PCM0, PCM1, PCS0 to PCS2, PCT0, PCT1, PCT4, PDH0 to PDH2, PDL0 to PDL15 (and their alternate-function pins)

2. $\overline{\text{RESET}}$, P00 to P03, P14, P30, P32, P34 to P36, P40, P42, P43, P45, P46, P98 to P915 (and their alternate-function pins)

DC characteristics 2

★ **(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 3.6 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V)**

Parameter	Symbol	Conditions		MIN.	TYP.[Note 1]	MAX.	Unit
Input leakage current, high	I_{LIH}	Pins other than X1, X2				5	μA
		X1, X2				20	μA
Input leakage current, low	I_{LIL}	Pins other than X1, X2				–5	μA
		X1, X2				–20	μA
Output leakage current, high	I_{LOH}	Pins other than X1, X2				5	μA
		X1, X2				20	μA
Output leakage current, low	I_{LOL}	Pins other than X1, X2				–5	μA
		X1, X2				–20	μA
★ Supply current	I_{DD1}[Note 2]	Normal operation All peripheral functions operating	f_{XX} = f_{CLK} = 20 MHz		20	35	mA
	I_{DD2}[Note 2]	HALT mode All peripheral functions operating	f_{XX} = f_{CLK} = 20 MHz		16	25	mA
★	I_{DD3}[Note 2]	IDLE mode RTC operating	f_{XX} = 20 MHz		1.2	4.5	mA
★	I_{DD4}[Note 3]	STOP mode, sub-STOP mode	V_{DD} = 3.3 V, subclock oscillation, RTC operating		10	57	μA
			V_{DD} = 3.3 V, T_A = 50°C, subclock oscillation, RTC operating			27	μA
			V_{DD} = 3.3 V, subclock oscillation stopped (XT1 = V_{SS})		1	37	μA
			V_{DD} = 3.3 V, T_A = 50°C, subclock oscillation stopped (XT1 = V_{SS})			7	μA
★	I_{DD5}[Note 3]	Subclock operation mode	V_{DD} = 3.3 V, f_{XT} = f_{CLK} = 32.768 kHz, main clock oscillation stopped		42	97	μA
			V_{DD} = 3.3 V, T_A = 50°C, f_{XT} = f_{CLK} = 32.768 kHz, main clock oscillation stopped			57	μA
★	I_{DD6}[Note 3]	Sub-IDLE mode	V_{DD} = 3.3 V, f_{XT} = 32.768 kHz, main clock oscillation stopped, RTC operating		10	57	μA
			V_{DD} = 3.3 V, T_A = 50°C, f_{XT} = 32.768 kHz, main clock oscillation stopped, RTC operating			27	μA
Pull-up resistance	R_L	V_{IN} = 0 V		10	30	100	kΩ

Notes 1. The typical value of V_{DD} is 3.3 V, and T_A is 25°C, excluding the current that flows through the output buffer.

2. Excluding the AV_{DD} supply current and the current that flows through the output buffer. The operating voltage range is V_{DD} = AV_{DD} = EV_{DD} = 2.7 to 3.6 V.

★ **3.** Excluding the current that flows through by the output buffer. The operating voltage range is V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 3.6 V.

Data retention characteristics

In STOP mode (T_A = –40 to +85°C, V_{SS} = AV_{SS} = EV_{SS} = 0 V)

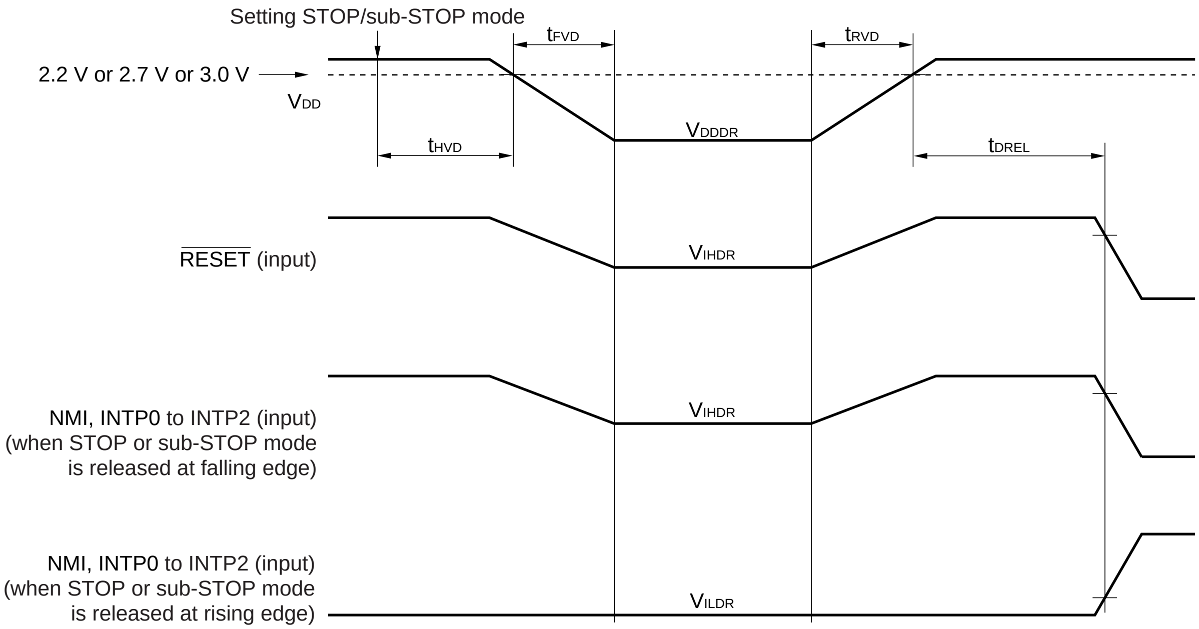
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Data retention voltage	V_{DDDR}	STOP mode, sub-STOP mode		1.8		3.6	V
★ Data retention current	I_{DDDR}[Note 1]	STOP mode, sub-STOP mode	V_{DD} = 3.3 V[Note 2], subclock oscillation, RTC operating		10	57	µA
			V_{DD} = 3.3 V[Note 2], subclock oscillation, RTC operating, T_A = 50°C			27	µA
			V_{DD} = 3.3 V[Note 2], subclock oscillation stopped (XT1 = V_{SS})		1	37	µA
			V_{DD} = 3.3 V[Note 2], subclock oscillation stopped (XT1 = V_{SS}), T_A = 50°C			7	µA
Supply voltage rise time	t_{RVD}			200			µs
Supply voltage fall time	t_{FVD}			200			µs
Supply voltage hold time (from STOP mode setting)	t_{HVD}			0			ms
STOP release signal input time	t_{DREL}			0			ms
Data retention high-level input voltage	V_{IHDR}	All input ports		V_{IHn}		V_{DDDR}	V
Data retention low-level input voltage	V_{ILDR}	All input ports		0		V_{ILn}	V

★ **Notes 1.** Excluding the current that flows through the output buffer.

2. V_{DD} = V_{DDDR}

Remarks 1. n = 1 to 3

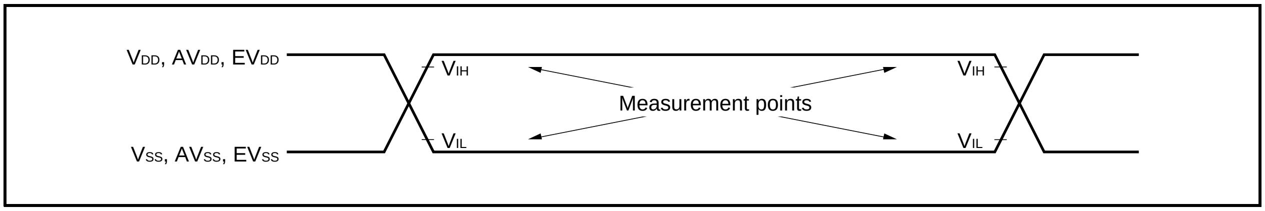
2. V_{IHn}: High-level input voltage, V_{ILn}: Low-level input voltage


Setting STOP/sub-STOP mode
2.2 V or 2.7 V or 3.0 V
VDD
tFVD
tRVD
tHVD
VDDDR
tDREL
RESET (input)
VIHDR
NMI, INTP0 to INTP2 (input)
(when STOP or sub-STOP mode
is released at falling edge)
VIHDR
NMI, INTP0 to INTP2 (input)
(when STOP or sub-STOP mode
is released at rising edge)
VILDR

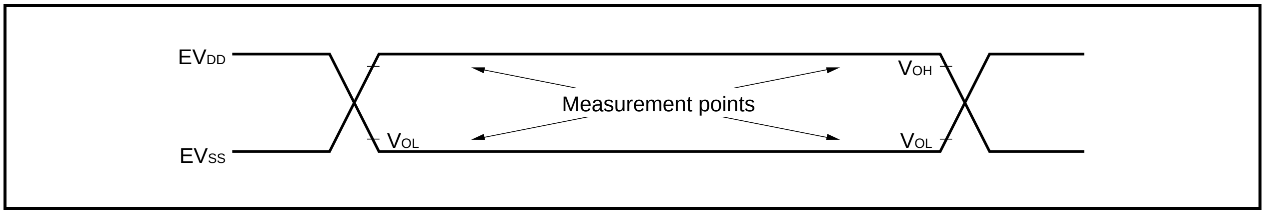
Caution **Shifting to STOP mode and restoring from STOP mode must be performed at V_{DD} = 3.0 V min. (f_{CLK} = 20 MHz) or V_{DD} = 2.7 V min. (f_{CLK} = 10 MHz). Shifting to sub-STOP mode and restoring from sub-STOP mode must be performed at V_{DD} = 2.2 V min. (f_{CLK} = 32.768 kHz).**

AC characteristics

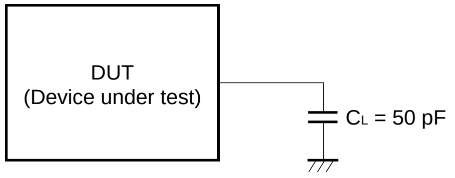
AC test input measurement points



AC test output measurement points



Load conditions



Caution If the load capacitance exceeds 50 pF due to the circuit configuration, reduce the load capacitance of the device to 50 pF or less by inserting a buffer or by some other means.

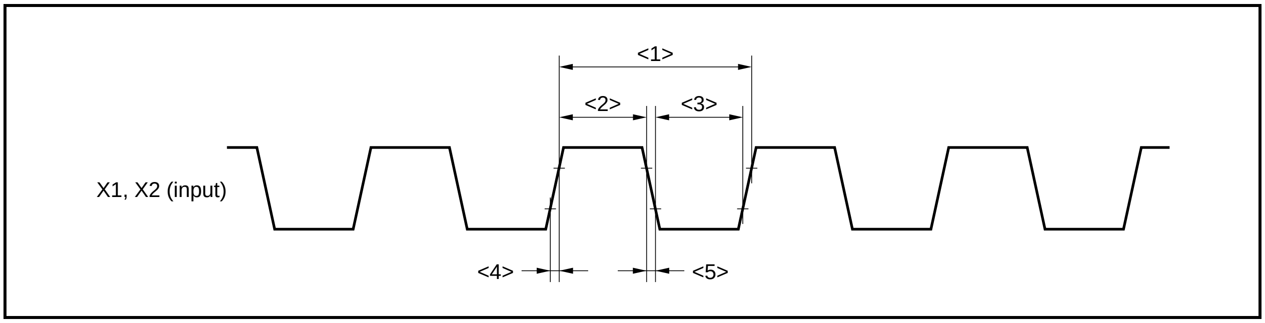
Clock timing

(1) X1, X2 external clock input timing

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
★ Input cycle	t_{CYX}	<1>	X1, X2	50	500	ns
★ High-level width	t_{WXH}	<2>	X1, X2	22.5		ns
★ Low-level width	t_{WXL}	<3>	X1, X2	22.5		ns
Rise time	t_{XR}	<4>			0.5 (<1>–<2>–<3>)	ns
Fall time	t_{XF}	<5>			0.5 (<1>–<2>–<3>)	ns

Caution The duty must be within a range of 45 to 55%.



(2) CLKOUT output timing

(a) T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 3.0 to 3.6 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF

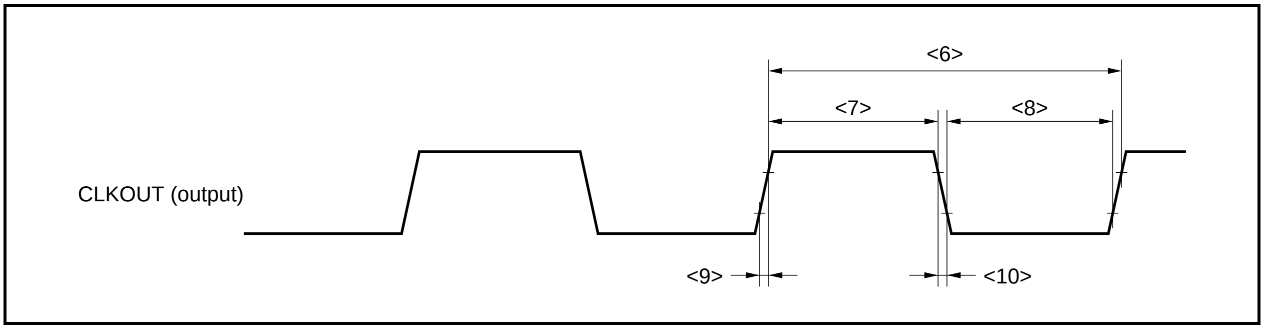
Parameter	Symbol		Conditions	MIN.	MAX.	Unit
★ Output cycle	t_{CYK}	<6>		0.05	31.25	μs
★ High-level width	t_{WKH}	<7>		0.4<6> – 10		ns
★ Low-level width	t_{WKL}	<8>		0.4<6> – 10		ns
Rise time	t_{KR}	<9>			10	ns
Fall time	t_{KF}	<10>			10	ns

★ **(b) T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF**

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output cycle	t_{CYK}	<6>		0.1	31.25	μs
High-level width	t_{WKH}	<7>		0.4<6> – 10		ns
Low-level width	t_{WKL}	<7>		0.4<6> – 10		ns
Rise time	t_{KR}	<9>			10	ns
Fall time	t_{KF}	<10>			10	ns

★ **(c) T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 3.6 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF**

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output cycle	t_{CYK}	<6>		28.57	31.25	μs
High-level width	t_{WKH}	<7>		0.4<6> – 15		ns
Low-level width	t_{WKL}	<8>		0.4<6> – 15		ns
Rise time	t_{KR}	<9>			15	ns
Fall time	t_{KF}	<10>			15	ns



(3) Pin output timing excluding PCM, PCS, PCT, PDH, and PDL

(a) T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 3.0 to 3.6 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF

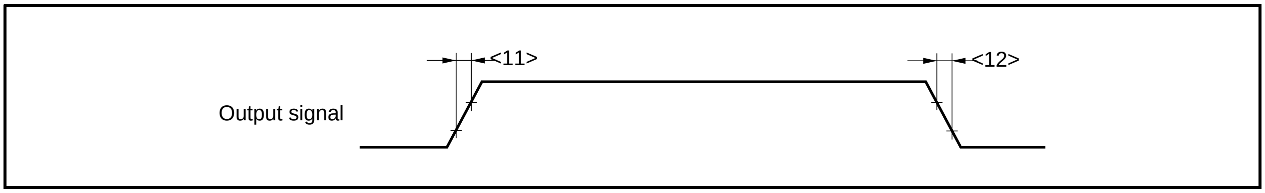
Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Rise time	t_{OR}	<11>			20	ns
Fall time	t_{OF}	<12>			20	ns

(b) T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Rise time	t_{OR}	<11>			25	ns
Fall time	t_{OF}	<12>			25	ns

★ #### (c) T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 3.6 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Rise time	t_{OR}	<11>			30	ns
Fall time	t_{OF}	<12>			30	ns



Bus timing

(1) Read cycle (CLKOUT asynchronous)

(a) $T_A = -40$ to $+85$°C, $V_{DD} = AV_{DD} = EV_{DD} = 3.0$ to 3.6 V, $V_{SS} = AV_{SS} = EV_{SS} = 0$ V, $C_L = 50$ pF

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address setup time (to $\overline{\text{RD}}$↓)	t_{SARD}	<13>		0.5T – 20		ns
Address hold time (from $\overline{\text{RD}}$↑)	t_{HARD}	<14>		–10		ns
$\overline{\text{RD}}$ low-level width	t_{WRDL}	<15>		(1.5 + n) T – 15		ns
Data setup time (to $\overline{\text{RD}}$↑)	t_{SISD}	<16>		15		ns
Data hold time (from $\overline{\text{RD}}$↑)	t_{HISD}	<17>		–2		ns
Data setup time (to address)	t_{SAID}	<18>			(2 + n) T – 30	ns
$\overline{\text{WAIT}}$ setup time (to $\overline{\text{RD}}$↓)	t_{SRDWT1}	<19>			0.5T – 20	ns
	t_{SRDWT2}	<20>			(0.5 + n) T – 20	ns
$\overline{\text{WAIT}}$ hold time (from $\overline{\text{RD}}$↓)	t_{HRDWT1}	<21>		0.5T		ns
	t_{HRDWT2}	<22>		(0.5 + n) T		ns
$\overline{\text{WAIT}}$ setup time (to address)	t_{SAWT1}	<23>			T – 30	ns
	t_{SAWT2}	<24>			(1 + n) T – 30	ns
$\overline{\text{WAIT}}$ hold time (from address)	t_{HAWT1}	<25>		T		ns
	t_{HAWT2}	<26>		(1 + n) T		ns

Remarks 1. T = 1/f_{CPU} (f_{CPU}: CPU operation clock frequency)

2. n: Number of wait clocks inserted in bus cycle
The sampling timing changes when a programmable wait is inserted.

3. The values in the above specifications are the values for when clocks with a 1:1 duty ratio are input from X1.

★ **(b) T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF**

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address setup time (to $\overline{\text{RD}}$↓)	t_{SARD}	<13>		0.5T – 25		ns
Address hold time (from $\overline{\text{RD}}$↑)	t_{HARD}	<14>		–12		ns
$\overline{\text{RD}}$ low-level width	t_{WRDL}	<15>		(1.5 + n) T – 20		ns
Data setup time (to $\overline{\text{RD}}$↑)	t_{SISD}	<16>		15		ns
Data hold time (from $\overline{\text{RD}}$↑)	t_{HISD}	<17>		–2		ns
Data setup time (to address)	t_{SAID}	<18>			(2 + n) T – 35	ns
$\overline{\text{WAIT}}$ setup time (to $\overline{\text{RD}}$↓)	t_{SRDWT1}	<19>			0.5T – 25	ns
	t_{SRDWT2}	<20>			(0.5 + n) T – 25	ns
$\overline{\text{WAIT}}$ hold time (from $\overline{\text{RD}}$↓)	t_{HRDWT1}	<21>		0.5T		ns
	t_{HRDWT2}	<22>		(0.5 + n) T		ns
$\overline{\text{WAIT}}$ setup time (to address)	t_{SAWT1}	<23>			T – 36	ns
	t_{SAWT2}	<24>			(1 + n) T – 36	ns
$\overline{\text{WAIT}}$ hold time (from address)	t_{HAWT1}	<25>		T		ns
	t_{HAWT2}	<26>		(1 + n) T		ns

Remarks 1. T = 1/f_{CPU} (f_{CPU}: CPU operation clock frequency)

2. n: Number of wait clocks inserted in bus cycle
The sampling timing changes when a programmable wait is inserted.

3. The values in the above specifications are the values for when clocks with a 1:1 duty ratio are input from X1.

(2) Write cycle (CLKOUT asynchronous)

(a) T_A = –40 to +85°C, $V_{DD} = AV_{DD} = EV_{DD}$ = 3.0 to 3.6 V, $V_{SS} = AV_{SS} = EV_{SS}$ = 0 V, C_L = 50 pF

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address setup time (to $\overline{\text{WRm}}$↓)	t_{SAW}	<27>		T – 20		ns
Address hold time (from $\overline{\text{WRm}}$↑)	t_{HAW}	<28>		0.5T – 15		ns
$\overline{\text{WRm}}$ low-level width	t_{WWRL}	<29>		(0.5 + n) T – 15		ns
Data output time from $\overline{\text{WRm}}$↓	t_{DOSDW}	<30>		–7		ns
Data setup time (to $\overline{\text{WRm}}$↑)	t_{SOSDW}	<31>		(0.5 + n) T – 15		ns
Data hold time (from $\overline{\text{WRm}}$↑)	t_{HOSDW}	<32>		0.5T – 15		ns
Data setup time (to address)	t_{SAOD}	<33>		T – 25		ns
$\overline{\text{WAIT}}$ setup time (to $\overline{\text{WRm}}$↓)	t_{SWRWT1}	<34>		20		ns
	t_{SWRWT2}	<35>		nT – 20		ns
$\overline{\text{WAIT}}$ hold time (from $\overline{\text{WRm}}$↓)	t_{HWRWT1}	<36>		0		ns
	t_{HWRWT2}	<37>		nT		ns
$\overline{\text{WAIT}}$ setup time (to address)	t_{SAWT1}	<38>			T – 30	ns
	t_{SAWT2}	<39>			(1 + n) T – 30	ns
$\overline{\text{WAIT}}$ hold time (from address)	t_{HAWT1}	<40>		T		ns
	t_{HAWT2}	<41>		(1 + n) T		ns

Remarks 1. m = 0, 1

2. T = 1/f_{CPU} (f_{CPU}: CPU operation clock frequency)

3. n: Number of wait clocks inserted in bus cycle
The sampling timing changes when a programmable wait is inserted.

4. The values in the above specifications are the values for when clocks with a 1:1 duty ratio are input from X1.

★ **(b) T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF**

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address setup time (to $\overline{WRm}$↓)	t_{SAW}	<27>		T – 25		ns
Address hold time (from $\overline{WRm}$↑)	t_{HAW}	<28>		0.5T – 20		ns
$\overline{WRm}$ low-level width	t_{WWRL}	<29>		(0.5 + n) T – 20		ns
Data output time from $\overline{WRm}$↓	t_{DOSDW}	<30>		–9		ns
Data setup time (to $\overline{WRm}$↑)	t_{SOSDW}	<31>		(0.5 + n) T – 20		ns
Data hold time (from $\overline{WRm}$↑)	t_{HOSDW}	<32>		0.5T – 20		ns
Data setup time (to address)	t_{SAOD}	<33>		T – 25		ns
$\overline{WAIT}$ setup time (to $\overline{WRm}$↓)	t_{SWRWT1}	<34>		22		ns
	t_{SWRWT2}	<35>		nT – 22		ns
$\overline{WAIT}$ hold time (from $\overline{WRm}$↓)	t_{HWRWT1}	<36>		0		ns
	t_{HWRWT2}	<37>		nT		ns
$\overline{WAIT}$ setup time (to address)	t_{SAWT1}	<38>			T – 36	ns
	t_{SAWT2}	<39>			(1 + n) T – 36	ns
$\overline{WAIT}$ hold time (from address)	t_{HAWT1}	<40>		T		ns
	t_{HAWT2}	<41>		(1 + n) T		ns

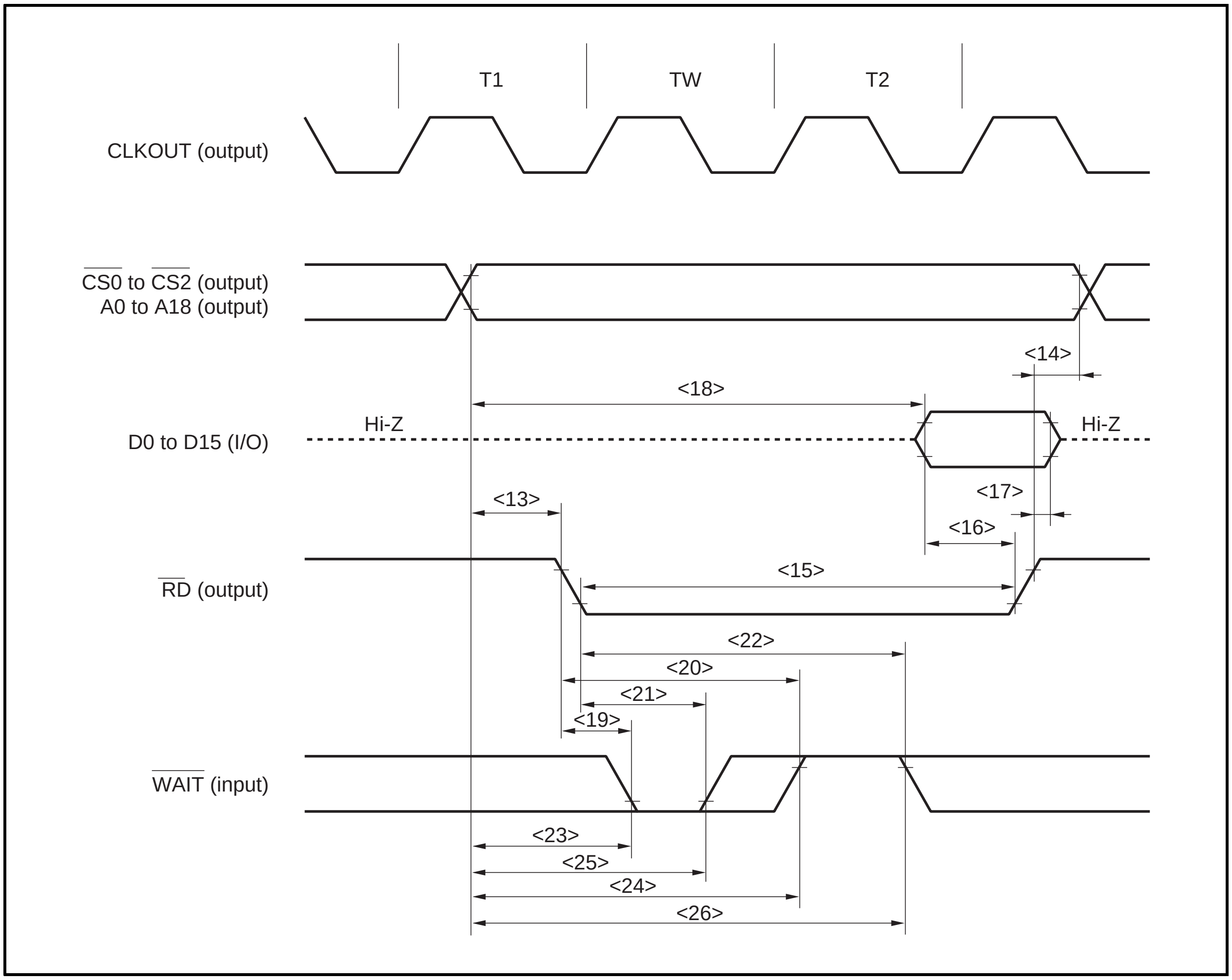
Remarks 1. m = 0, 1

2. T = 1/f_{CPU} (f_{CPU}: CPU operation clock frequency)

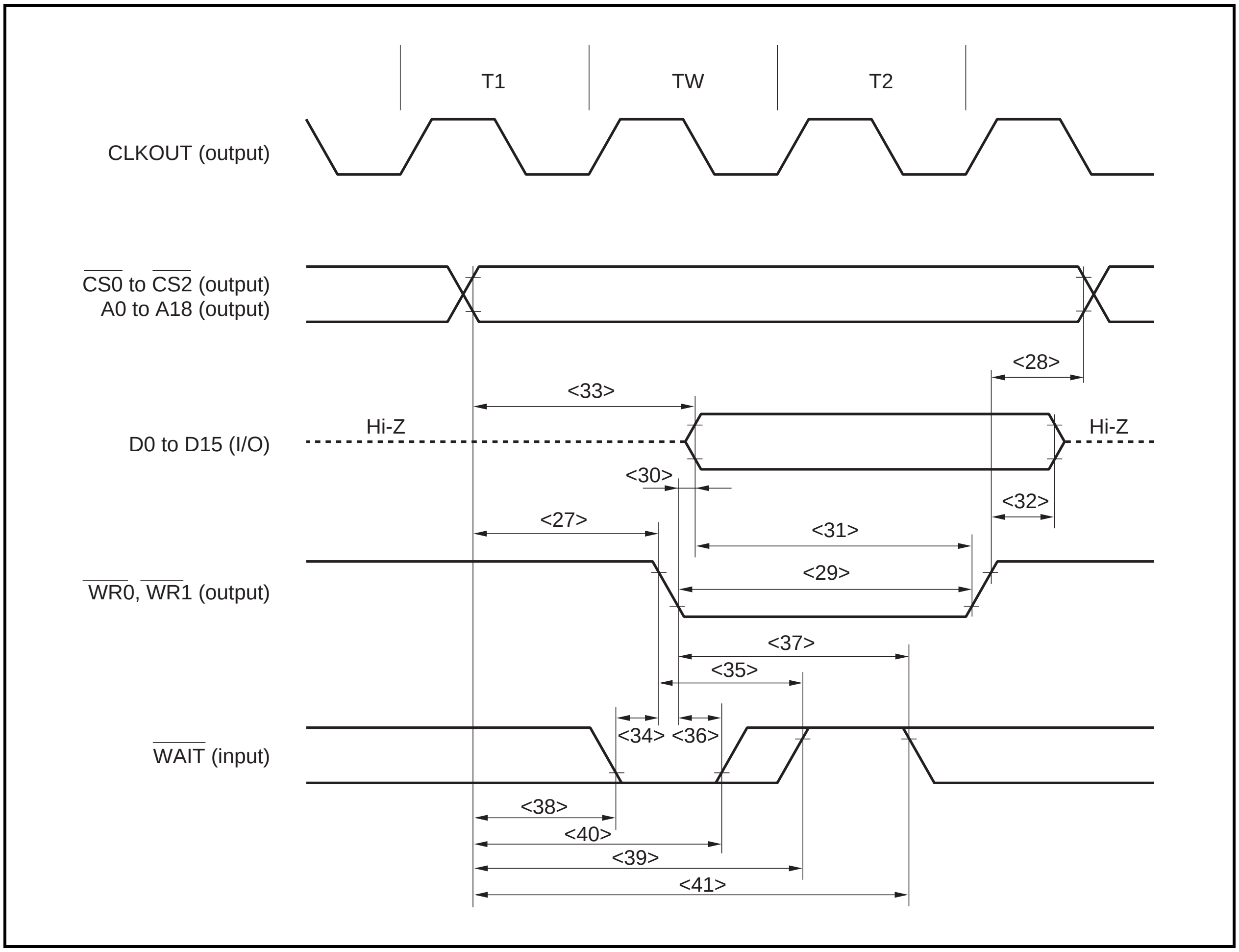
3. n: Number of wait clocks inserted in bus cycle
The sampling timing changes when a programmable wait is inserted.

4. The values in the above specifications are the values for when clocks with a 1:1 duty ratio are input from X1.

Read cycle (CLKOUT asynchronous, 1 wait)



Write cycle (CLKOUT asynchronous, 1 wait)



Reset/interrupt timing

★ **(a) T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF**

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ high-level width	t_{WRSH}	<42>		500		ns
$\overline{\text{RESET}}$ low-level width	t_{WRSL}	<43>		500		ns
NMI high-level width	t_{WNIH}	<44>		500		ns
NMI low-level width	t_{WNIL}	<45>		500		ns
INTPn high-level width	t_{WITHn}	<46>	n = 0 to 2	500		ns
INTPn low-level width	t_{WITLn}	<47>	n = 0 to 2	500		ns

Remark T = 1/f_{XX}

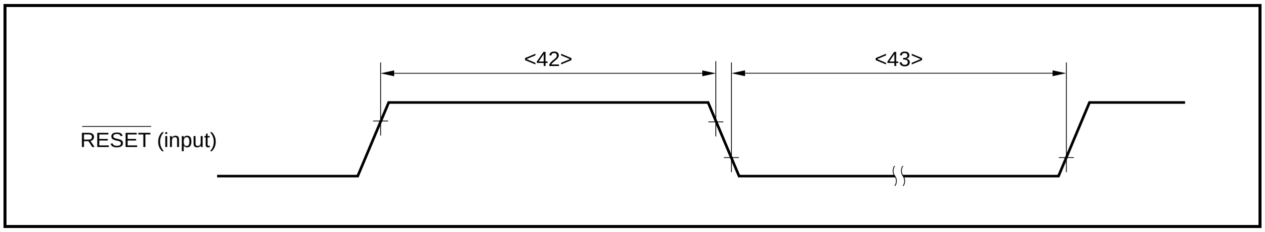
★ **(b) T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 3.6 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF**

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ high-level width[Note]	t_{WRSH}	<42>		600		ns
$\overline{\text{RESET}}$ low-level width[Note]	t_{WRSL}	<43>		600		ns
NMI high-level width	t_{WNIH}	<44>		600		ns
NMI low-level width	t_{WNIL}	<45>		600		ns
INTPn high-level width	t_{WITHn}	<46>	n = 0 to 2	600		ns
INTPn low-level width	t_{WITLn}	<47>	n = 0 to 2	600		ns

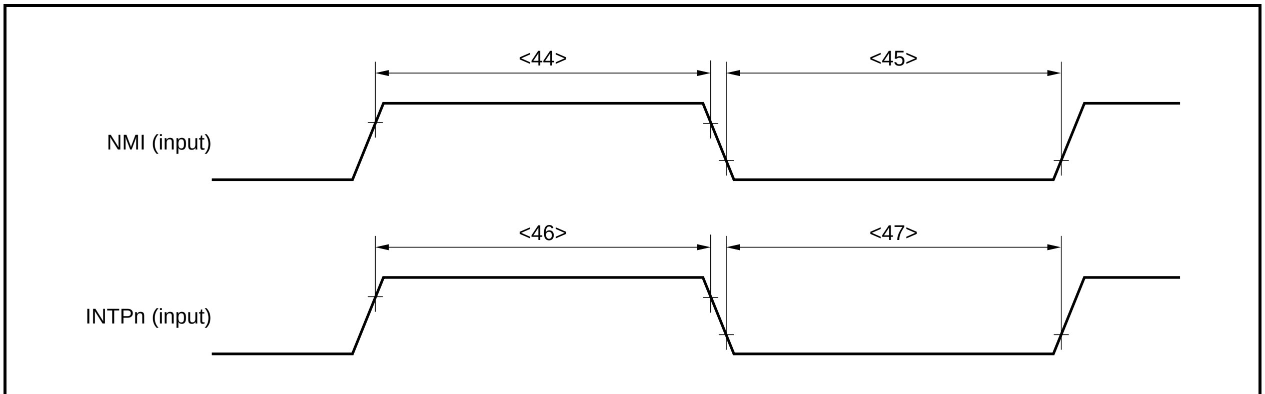
Note Release reset when V_{DD} is 2.7 V or higher.

Remark T = 1/f_{XX}

Reset



Interrupt



Remark n = 0 to 2

Timer timing

★ **(a) T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF**

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
TIn high-level width	t_{WTIHn}	n = 0m0, 0m1[Note], 10, 11, 20, 21	2T + 20		ns
TIn low-level width	t_{WTILn}	n = 0m0, 0m1[Note], 10, 11, 20, 21	2T + 20		ns
TCLRm high-level width	t_{WTCHm}	m = 10, 11	2T + 20		ns
TCLRm low-level width	t_{WTCLm}	m = 10, 11	2T + 20		ns
INTPm high-level width	t_{WITHm}	m = 100, 101, 110, 111	2T + 20		ns
INTPm low-level width	t_{WITLm}	m = 100, 101, 110, 111	2T + 20		ns

Note T is equal to one cycle of the TM0m (m = 0 to 3) count clock when TI0m0 or TI0m1 (m = 0 to 3) is input as the capture trigger. It is 1/f_{XX} when TI0m0 or TI0m1 is input as the external clock.

Remark T = 1/f_{XX}

★ CSI timing

(1) Master mode

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKn}}$ cycle time	t_{KCYn}	<48>	EV_{DD} = 3.0 to 3.6 V	200		ns
			EV_{DD} = 2.7 to 3.6 V	400		ns
$\overline{\text{SCKn}}$ high-/low-level width	t_{KHn}, t_{KLn}	<49>	EV_{DD} = 3.0 to 3.6 V	$t_{KCYn}/2 - 20$		ns
			EV_{DD} = 2.7 to 3.6 V	$t_{KCYn}/2 - 25$		ns
SIn setup time (to $\overline{\text{SCKn}}$↑)	t_{SIKn}	<50>	EV_{DD} = 3.0 to 3.6 V	30		ns
			EV_{DD} = 2.7 to 3.6 V	30		ns
SIn setup time (to $\overline{\text{SCKn}}$↓)	t_{SIKn}	<50>	EV_{DD} = 3.0 to 3.6 V	30		ns
			EV_{DD} = 2.7 to 3.6 V	30		ns
SIn hold time (from $\overline{\text{SCKn}}$↑)	t_{KSIn}	<51>	EV_{DD} = 3.0 to 3.6 V	30		ns
			EV_{DD} = 2.7 to 3.6 V	30		ns
SIn hold time (from $\overline{\text{SCKn}}$↓)	t_{KSIn}	<51>	EV_{DD} = 3.0 to 3.6 V	30		ns
			EV_{DD} = 2.7 to 3.6 V	30		ns
Delay time from $\overline{\text{SCKn}}$↑ to SOn output	t_{KSOn}	<52>	EV_{DD} = 3.0 to 3.6 V		30	ns
			EV_{DD} = 2.7 to 3.6 V		30	ns
Delay time from $\overline{\text{SCKn}}$↓ to SOn output	t_{KSOn}	<52>	EV_{DD} = 3.0 to 3.6 V		30	ns
			EV_{DD} = 2.7 to 3.6 V		30	ns
Hold time from $\overline{\text{SCKn}}$↑ to SOn output	t_{HSKSOn}	<53>	EV_{DD} = 3.0 to 3.6 V	$t_{KCYn}/2 - 20$		ns
			EV_{DD} = 2.7 to 3.6 V	$t_{KCYn}/2 - 25$		ns
Hold time from $\overline{\text{SCKn}}$↓ to SOn output	t_{HSKSOn}	<53>	EV_{DD} = 3.0 to 3.6 V	$t_{KCYn}/2 - 20$		ns
			EV_{DD} = 2.7 to 3.6 V	$t_{KCYn}/2 - 25$		ns

Remark n = 0, 1

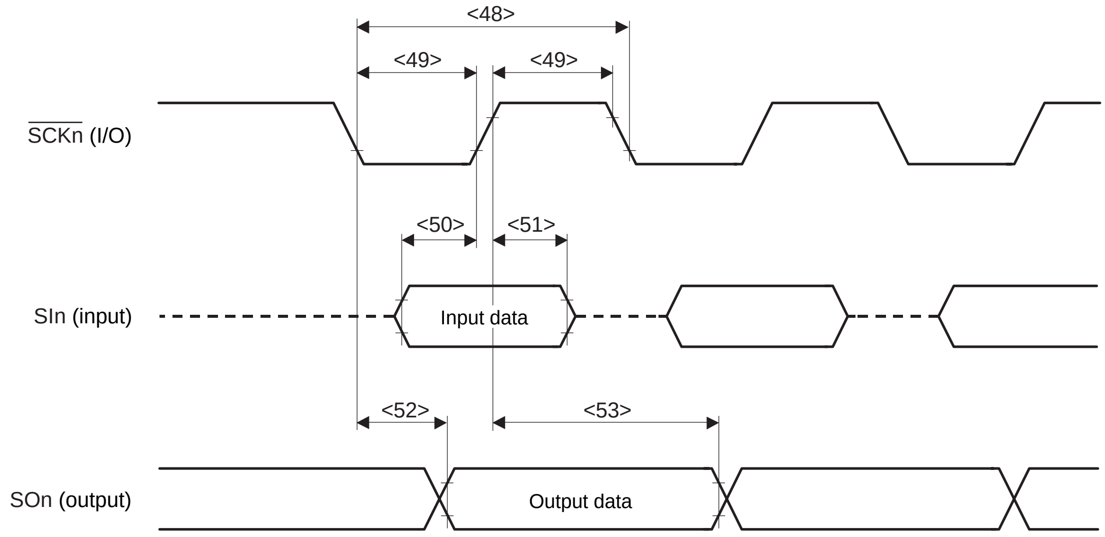
(2) Slave mode

(T_A = –40 to +85°C, $V_{DD} = AV_{DD} = EV_{DD}$ = 2.7 to 3.6 V, $V_{SS} = AV_{SS} = EV_{SS}$ = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKn}}$ cycle time	t_{KCYn}	<48>	EV_{DD} = 3.0 to 3.6 V	200		ns
			EV_{DD} = 2.7 to 3.6 V	400		ns
$\overline{\text{SCKn}}$ high-/low-level width	t_{KHn}, t_{KLn}	<49>	EV_{DD} = 3.0 to 3.6 V	90		ns
			EV_{DD} = 2.7 to 3.6 V	190		ns
SIn setup time (to $\overline{\text{SCKn}}$↑)	t_{SIKn}	<50>	EV_{DD} = 3.0 to 3.6 V	50		ns
			EV_{DD} = 2.7 to 3.6 V	50		ns
SIn setup time (to $\overline{\text{SCKn}}$↓)	t_{SIKn}	<50>	EV_{DD} = 3.0 to 3.6 V	50		ns
			EV_{DD} = 2.7 to 3.6 V	50		ns
SIn hold time (from $\overline{\text{SCKn}}$↑)	t_{KSIn}	<51>	EV_{DD} = 3.0 to 3.6 V	50		ns
			EV_{DD} = 2.7 to 3.6 V	50		ns
SIn hold time (from $\overline{\text{SCKn}}$↓)	t_{KSIn}	<51>	EV_{DD} = 3.0 to 3.6 V	50		ns
			EV_{DD} = 2.7 to 3.6 V	50		ns
Delay time from $\overline{\text{SCKn}}$↑ to SOn output	t_{KSOn}	<52>	EV_{DD} = 3.0 to 3.6 V		50	ns
			EV_{DD} = 2.7 to 3.6 V		50	ns
Delay time from $\overline{\text{SCKn}}$↓ to SOn output	t_{KSOn}	<52>	EV_{DD} = 3.0 to 3.6 V		50	ns
			EV_{DD} = 2.7 to 3.6 V		50	ns
Hold time from $\overline{\text{SCKn}}$↑ to SOn output	t_{HSKSOn}	<53>	EV_{DD} = 3.0 to 3.6 V	t_{KHn}		ns
			EV_{DD} = 2.7 to 3.6 V	t_{KHn}		ns
Hold time from $\overline{\text{SCKn}}$↓ to SOn output	t_{HSKSOn}	<53>	EV_{DD} = 3.0 to 3.6 V	t_{KLn}		ns
			EV_{DD} = 2.7 to 3.6 V	t_{KLn}		ns

Remark n = 0, 1

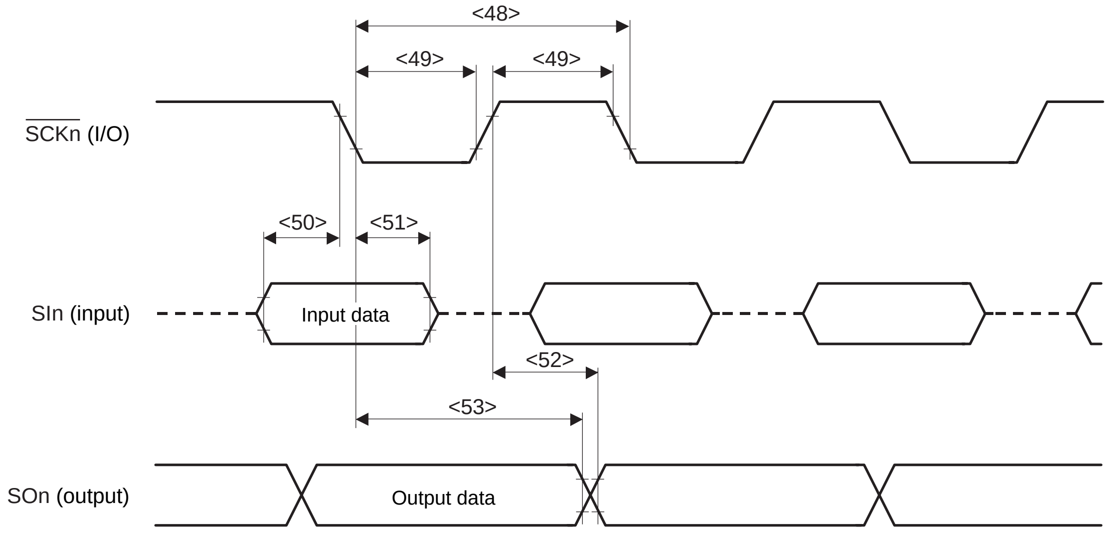
(3) Timing when CSICn.CKPn and CSICn.DAPn bits = 00



Remarks 1. Broken lines indicate high impedance.

2. n = 0, 1

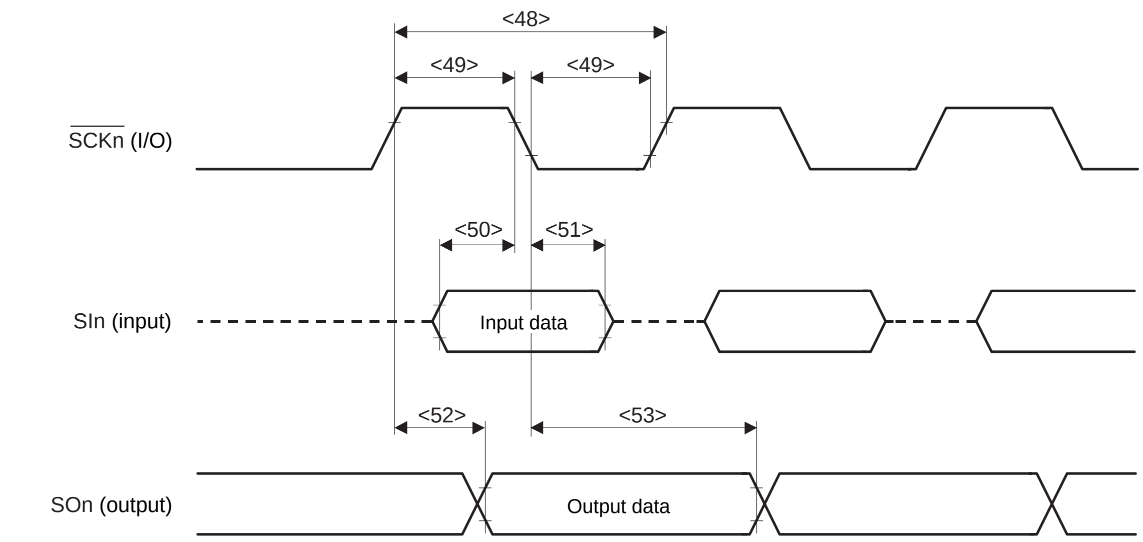
(4) Timing when CSICn.CKPn and CSICn.DAPn bits = 01



Remarks 1. Broken lines indicate high impedance.

2. n = 0, 1

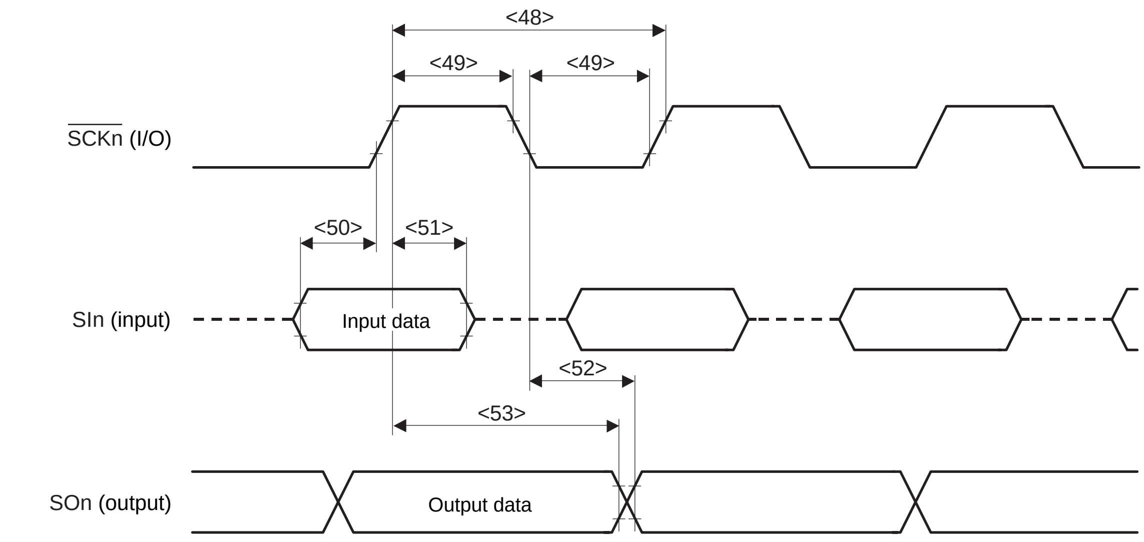
(5) Timing when CSICn.CKPn and CSICn.DAPn bits = 10



Remarks 1. Broken lines indicate high impedance.

2. n = 0, 1

(6) Timing when CSICn.CKPn and CSICn.DAPn bits = 11



Remarks 1. Broken lines indicate high impedance.

2. n = 0, 1

A/D converter characteristics

(1) Recommended operating conditions (T_A = –40 to +85°C, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Supply voltage	AV_{DD}		3.0	3.3	3.6	V
	EV_{DD}		3.0	3.3	3.6	V
	V_{DD}		3.0	3.3	3.6	V
Clock frequency	f_{XX}			20		MHz
Operating ambient temperature	T_A		–40	27	+85	°C

(2) Reference (under recommended operating conditions)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
External reference potential (input)	AV_{REFIN}		1.2		1.25	V
Internal reference potential (output)	AV_{REFOUT}		–5.0%	1.226	+5.0%	V
★ Internal reference potential temperature coefficient		**Note**		52		ppm/°C
Reference smoothing capacitance	C_{REF}			10		μF

★ **Note** Temperature coefficient of –40°C to +25°C and +25°C to +100°C

(3) Analog input specifications

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input signal DC level	Voltage CH			–20	0	+20	mV
	Current CH		Gain: ×2	–10	0	+10	mV
			Gain: ×16	–1.25	0	+1.25	mV
Input signal range	Voltage CH			–0.375		+0.375	V
	Current CH		Gain: ×2	–0.1875		+0.1875	V
			Gain: ×16	–23.4		+23.4	mV
Input gain[Note]	Voltage CH			–7%	1	+7%	–
	Current CH		Gain: ×2	–7%	2	+7%	–
			Gain: ×16	–7%	16	+7%	–
Input impedance	Voltage CH			100	125		kΩ
	Current CH		Gain: ×2	60	75		kΩ
			Gain: ×16	60	75		kΩ

Note The gain of all the current CHs is ×16 if the ×16 gain mode is selected for even one current CH.

Remarks 1. n = 0 to 5

2. Voltage CH: Channels 0, 2, and 4
Current CH: Channels 1, 3, and 5

(4) A/D converter and system specifications

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
A/D converter and system (fs = 4,340 Hz)							
System clock	f_{XX}				20		MHz
ΔΣ operation clock	DSCLK	f_{XX}/12			1.667		MHz
Oversampling frequency	fos	DSCLK/3			555.6		kHz
Sampling frequency	fs	fos/128			4.34		kHz
Data width					16		bit
S/N	SNR	0 dB, 60 Hz single sine wave input	Voltage CH	70	76		dB
			Current CH, Gain: ×2	70	76		dB
★			Current CH, Gain: ×16	62	69		dB
THD	THD	0 dB, 60 Hz single sine wave input	Voltage CH		–80	–72	dB
			Current CH, Gain: ×2		–80	–72	dB
			Current CH, Gain: ×16		–80	–72	dB
Inter-channel isolation	XT			80			dB
★ Operating current	IAV_{DD}				4.6	10.0	mA
A/D converter and system (fs = 2,170 Hz)							
System clock	f_{XX}				20		MHz
ΔΣ operation clock	DSCLK	f_{XX}/24			0.833		MHz
Oversampling frequency	fos	DSCLK/3			277.8		kHz
Sampling frequency	fs	fos/128			2.17		kHz
Data width					16		bit
S/N	SNR	0 dB, 60 Hz single sine wave input	Voltage CH	70	76		dB
			Current CH, Gain: ×2	70	76		dB
★			Current CH, Gain: ×16	62	69		dB
THD	THD	0 dB, 60 Hz single sine wave input	Voltage CH		–80	–72	dB
			Current CH, Gain: ×2		–80	–72	dB
			Current CH, Gain: ×16		–80	–72	dB
Inter-channel isolation	XT			80			dB
★ Operating current	IAV_{DD}				4.6	10.0	mA

Remarks 1. Voltage CH: Channels 0, 2, and 4
Current CH: Channels 1, 3, and 5

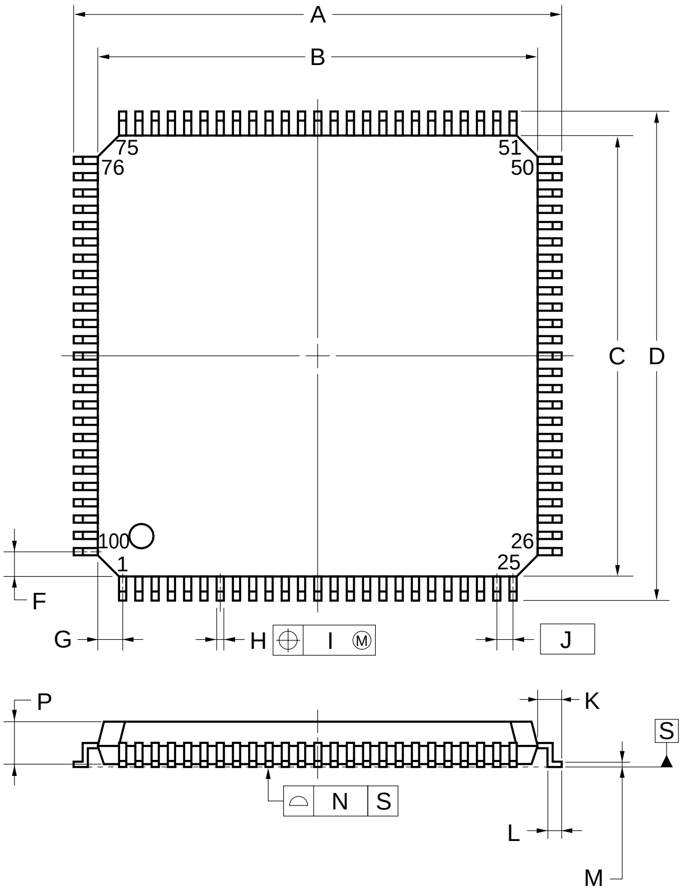
★ **2.** S/N: Ratio of the signal frequency component to the sum of the parameters other than the signal frequency and harmonic component when a signal (0 dB, 60 Hz) is input

★ **3.** THD: Sum of the harmonic component when a signal (0 dB, 60 Hz) is input

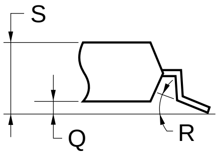
(5) Digital filter specifications

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Digital filter characteristics (fs = 4,340 Hz)						
Pass region (low region)	fchpf	–3 dB		0.73		Hz
In-band ripple 1	rp1	50 Hz at center, 45 Hz to 55 Hz 60 Hz at center, 54 Hz to 66 Hz	–0.01		+0.01	dB
In-band ripple 2	rp2	50 Hz at center, 45 Hz to 275 Hz 60 Hz at center, 54 Hz to 330 Hz	–0.1		+0.1	dB
In-band ripple 3	rp3	50 Hz at center, 45 Hz to 1,100 Hz 60 Hz at center, 54 Hz to 1,320 Hz	–0.1		+0.1	dB
Block region (high region)	fatt	–80 dB		3,020		Hz
Attenuation out of band	ATT		–80			dB
Digital filter characteristics (fs = 2,170 Hz)						
Pass region (low region)	fchpf	–3 dB		0.365		Hz
In-band ripple 1	rp1	50 Hz at center, 45 Hz to 55 Hz 60 Hz at center, 54 Hz to 66 Hz	–0.01		+0.01	dB
In-band ripple 2	rp2	50 Hz at center, 45 Hz to 275 Hz 60 Hz at center, 54 Hz to 330 Hz	–0.1		+0.1	dB
Block region (high region)	fatt	–80 dB		1,510		Hz
Attenuation out of band	ATT		–80			dB

100-PIN PLASTIC LQFP (FINE PITCH) (14x14)



detail of lead end



NOTE

Each lead centerline is located within 0.08 mm of
its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	16.00±0.20
B	14.00±0.20
C	14.00±0.20
D	16.00±0.20
F	1.00
G	1.00
H	$0.22^{+0.05}_{-0.04}$
I	0.08
J	0.50 (T.P.)
K	1.00±0.20
L	0.50±0.20
M	$0.17^{+0.03}_{-0.07}$
N	0.08
P	1.40±0.05
Q	0.10±0.05
R	$3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$
S	1.60 MAX.

S100GC-50-8EU, 8EA-2

★

CHAPTER 22 RECOMMENDED SOLDERING CONDITIONS

The V850ES/PM1 should be soldered and mounted under the following recommended conditions. For soldering methods and conditions other than those recommended below, contact an NEC Electronics sales representative.

For details of the recommended soldering conditions, see the Semiconductor Device Mount Manual website (http://www.necel.com/pkg/en/mount/index.html).

Table 22-1. Surface Mounting Type Soldering Conditions

(1) μPD703228GC-××× -8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 30 seconds max. (at 210°C or higher), Count: Three times or less, Exposure limit: 3 days[Note] (after that, prebake at 125°C for 36 hours)	IR60-363-3
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

APPENDIX A REGISTER INDEX

(1/6)

Symbol	Name	Unit	Page
ADCR0	A/D conversion result register 0	ADC	258
ADCR1	A/D conversion result register 1	ADC	258
ADCR2	A/D conversion result register 2	ADC	258
ADCR3	A/D conversion result register 3	ADC	258
ADCR4	A/D conversion result register 4	ADC	258
ADCR5	A/D conversion result register 5	ADC	258
ADIC	Interrupt control register	INTC	325
ADLY	A/D clock delay setting register	ADC	258
ADM	A/D converter mode register	ADC	257
ASIF0	Asynchronous serial interface transmit status register 0	UART	276
ASIF1	Asynchronous serial interface transmit status register 1	UART	276
ASIM0	Asynchronous serial interface mode register 0	UART	272
ASIM1	Asynchronous serial interface mode register 1	UART	272
ASIS0	Asynchronous serial interface status register 0	UART	275
ASIS1	Asynchronous serial interface status register 1	UART	275
AWC	Address wait control register	BCU	135
BCC	Bus cycle control register	BCU	136
BRGC0	Baud rate generator control register 0	UART	294
BRGC1	Baud rate generator control register 1	UART	294
BSC	Bus size configuration register	BCU	125
CC100	16-bit timer capture/compare register 100	TM1	194
CC101	16-bit timer capture/compare register 101	TM1	194
CC110	16-bit timer capture/compare register 110	TM1	194
CC111	16-bit timer capture/compare register 111	TM1	194
CCIC100	Interrupt control register	INTC	325
CCIC101	Interrupt control register	INTC	325
CCIC110	Interrupt control register	INTC	325
CCIC111	Interrupt control register	INTC	325
CKSR0	Clock select register 0	UART	293
CKSR1	Clock select register 1	UART	293
CORAD0	Correction address register 0	ROMC	365
CORAD0H	Correction address register 0H	ROMC	365
CORAD0L	Correction address register 0L	ROMC	365
CORAD1	Correction address register 1	ROMC	365
CORAD1H	Correction address register 1H	ROMC	365
CORAD1L	Correction address register 1L	ROMC	365
CORAD2	Correction address register 2	ROMC	365
CORAD2H	Correction address register 2H	ROMC	365
CORAD2L	Correction address register 2L	ROMC	365

(2/6)

Symbol	Name	Unit	Page
CORAD3	Correction address register 3	ROMC	365
CORAD3H	Correction address register 3H	ROMC	365
CORAD3L	Correction address register 3L	ROMC	365
CORCN	Correction control register	ROMC	366
CR000	16-bit timer capture/compare register 000	TM0	150
CR001	16-bit timer capture/compare register 001	TM0	151
CR010	16-bit timer capture/compare register 010	TM0	150
CR011	16-bit timer capture/compare register 011	TM0	151
CR020	16-bit timer capture/compare register 020	TM0	150
CR021	16-bit timer capture/compare register 021	TM0	151
CR030	16-bit timer capture/compare register 030	TM0	150
CR031	16-bit timer capture/compare register 031	TM0	151
CR2	16-bit timer compare register 2	TM2	220
CR20	8-bit timer compare register 20	TM2	220
CR21	8-bit timer compare register 21	TM2	220
CRC00	Capture/compare control register 00	TM0	155
CRC01	Capture/compare control register 01	TM0	155
CRC02	Capture/compare control register 02	TM0	155
CRC03	Capture/compare control register 03	TM0	155
CSIC0	Clocked serial interface clock select register 0	CSI	303
CSIC1	Clocked serial interface clock select register 1	CSI	303
CSIIC0	Interrupt control register	INTC	325
CSIIC1	Interrupt control register	INTC	325
CSIM0	Clocked serial interface mode register 0	CSI	302
CSIM1	Clocked serial interface mode register 1	CSI	302
DAY	Day count register	RTC	243
DAYB	Day count setting register	RTC	243
DWC0	Data wait control register 0	BCU	133
HOUR	Hour count register	RTC	242
HOURB	Hour count setting register	RTC	243
HOURDAY	Day/hour count register	RTC	54
HOURDAYB	Day/hour count setting register	RTC	54
HPFC0	High-pass filter control register 0	ADC	258
IMR0	Interrupt mask register 0	INTC	327
IMR0H	Interrupt mask register 0H	INTC	327
IMR0L	Interrupt mask register 0L	INTC	327
IMR1	Interrupt mask register 1	INTC	327
IMR1H	Interrupt mask register 1H	INTC	327
IMR1L	Interrupt mask register 1L	INTC	327
INTF0	External interrupt falling edge specification register 0	INTC	331
INTR0	External interrupt rising edge specification register 0	INTC	331
ISPR	In-service priority register	INTC	328
MIN	Minute count register	RTC	242
MINB	Minute count setting register	RTC	242

(3/6)

Symbol	Name	Unit	Page
OSTS	Oscillation stabilization time select register	Standby	345
OVFIC10	Interrupt control register	INTC	325
OVFIC11	Interrupt control register	INTC	325
P0	Port 0	Port	68
P1	Port 1	Port	71
P2	Port 2	Port	74
P3	Port 3	Port	76
P4	Port 4	Port	80
P9	Port 9	Port	84
P9H	Port 9H	Port	84
P9L	Port 9L	Port	84
PCC	Processor clock control register	CG	144
PCM	Port CM	Port	88
PCS	Port CS	Port	90
PCT	Port CT	Port	92
PDH	Port DH	Port	94
PDL	Port DL	Port	97
PDLH	Port DLH	Port	97
PDLL	Port DLL	Port	97
PFC0	Port 0 function control register	Port	69
PFC1	Port 1 function control register	Port	73
PFC3	Port 3 function control register	Port	79
PFC4	Port 4 function control register	Port	82
PFC9	Port 9 function control register	Port	87
PFC9H	Port 9 function control register H	Port	87
PFC9L	Port 9 function control register L	Port	87
PIC0	Interrupt control register	INTC	325
PIC1	Interrupt control register	INTC	325
PIC2	Interrupt control register	INTC	325
PM0	Port 0 mode register	Port	68
PM1	Port 1 mode register	Port	71
PM2	Port 2 mode register	Port	74
PM3	Port 3 mode register	Port	77
PM4	Port 4 mode register	Port	80
PM9	Port 9 mode register	Port	84
PM9H	Port 9 mode register H	Port	84
PM9L	Port 9 mode register L	Port	84
PMC0	Port 0 mode control register	Port	69
PMC1	Port 1 mode control register	Port	72
PMC2	Port 2 mode control register	Port	75
PMC3	Port 3 mode control register	Port	78
PMC4	Port 4 mode control register	Port	81
PMC9	Port 9 mode control register	Port	85
PMC9H	Port 9 mode control register H	Port	85

(4/6)

Symbol	Name	Unit	Page
PMC9L	Port 9 mode control register L	Port	85
PMCCM	Port CM mode control register	Port	89
PMCCS	Port CS mode control register	Port	91
PMCCT	Port CT mode control register	Port	93
PMCDH	Port DH mode control register	Port	95
PMCDL	Port DL mode control register	Port	98
PMCDLH	Port DL mode control register H	Port	98
PMCDLL	Port DL mode control register L	Port	98
PMCM	Port CM mode register	Port	88
PMCS	Port CS mode register	Port	90
PMCT	Port CT mode register	Port	92
PMDH	Port DH mode register	Port	94
PMDL	Port DL mode register	Port	97
PMDLH	Port DL mode register H	Port	97
PMDLL	Port DL mode register L	Port	97
PRCMD	Command register	CPU	58
PRM00	Prescaler mode register 00	TM0	158
PRM01	Prescaler mode register 01	TM0	158
PRM02	Prescaler mode register 02	TM0	159
PRM03	Prescaler mode register 03	TM0	159
PSC	Power save control register	Standby	344
PSMR	Power save mode register	Standby	344
PU0	Pull-up resistor option register 0	Port	70
PU1	Pull-up resistor option register 1	Port	73
PU2	Pull-up resistor option register 2	Port	75
PU3	Pull-up resistor option register 3	Port	79
PU4	Pull-up resistor option register 4	Port	82
PWMB0	PWM buffer register 0	PWM	264
PWMB1	PWM buffer register 1	PWM	264
PWMB2	PWM buffer register 2	PWM	264
PWMB3	PWM buffer register 3	PWM	264
PWMC0	PWM control register 0	PWM	263
PWMC1	PWM control register 1	PWM	263
PWMC2	PWM control register 2	PWM	263
PWMC3	PWM control register 3	PWM	263
ROVIC	Interrupt control register	INTC	325
RTCC	RTC control register	RTC	53
RTCC0	RTC control register 0	RTC	239
RTCC1	RTC control register 1	RTC	240
RTCIC	Interrupt control register	INTC	325
RXB0	Receive buffer register 0	UART	277
RXB1	Receive buffer register 1	UART	277
SEC	Second count register	RTC	241
SECB	Second count setting register	RTC	241

(5/6)

Symbol	Name	Unit	Page
SECMIN	Minute/second count register	RTC	53
SECMINB	Minute/second count setting register	RTC	54
SES10	Valid edge select register 10	TM1	200
SES11	Valid edge select register 11	TM1	200
SIO0	Serial I/O shift register 0	CSI	304
SIO1	Serial I/O shift register 1	CSI	304
SIOE0	Receive-only serial I/O shift register 0	CSI	305
SIOE1	Receive-only serial I/O shift register 1	CSI	305
SOTB0	Clocked serial interface transmit buffer register 0	CSI	305
SOTB1	Clocked serial interface transmit buffer register 1	CSI	305
SREIC0	Interrupt control register	INTC	325
SREIC1	Interrupt control register	INTC	325
SRIC0	Interrupt control register	INTC	325
SRIC1	Interrupt control register	INTC	325
STIC0	Interrupt control register	INTC	325
STIC1	Interrupt control register	INTC	325
SUBC	Sub-count register	RTC	241
SUBCH	Sub-count register H	RTC	53
SUBCL	Sub-count register L	RTC	53
SYS	System status register	CPU	59
TCL2	Timer clock select register 2	TM2	53
TCL20	Timer clock selection register 20	TM2	222
TCL21	Timer clock selection register 21	TM2	222
TM00	16-bit timer counter 00	TM0	149
TM01	16-bit timer counter 01	TM0	149
TM02	16-bit timer counter 02	TM0	149
TM03	16-bit timer counter 03	TM0	149
TM10	16-bit timer counter 10	TM1	192
TM11	16-bit timer counter 11	TM1	192
TM2	16-bit timer counter 2	TM2	53
TM20	8-bit timer counter 20	TM2	220
TM21	8-bit timer counter 21	TM2	220
TMC00	16-bit timer mode control register 00	TM0	153
TMC01	16-bit timer mode control register 01	TM0	153
TMC02	16-bit timer mode control register 02	TM0	153
TMC03	16-bit timer mode control register 03	TM0	153
TMC100	16-bit timer mode control register 100	TM1	196
TMC101	16-bit timer mode control register 101	TM1	198
TMC110	16-bit timer mode control register 110	TM1	196
TMC111	16-bit timer mode control register 111	TM1	198
TMC2	16-bit timer mode control register 2	TM2	53
TMC20	8-bit timer mode control register 20	TM2	223
TMC21	8-bit timer mode control register 21	TM2	223

(6/6)

Symbol	Name	Unit	Page
TMIC000	Interrupt control register	INTC	325
TMIC001	Interrupt control register	INTC	325
TMIC010	Interrupt control register	INTC	325
TMIC011	Interrupt control register	INTC	325
TMIC020	Interrupt control register	INTC	325
TMIC021	Interrupt control register	INTC	325
TMIC030	Interrupt control register	INTC	325
TMIC031	Interrupt control register	INTC	325
TMIC20	Interrupt control register	INTC	325
TMIC21	Interrupt control register	INTC	325
TOC00	16-bit timer output control register 00	TM0	156
TOC01	16-bit timer output control register 01	TM0	156
TOC02	16-bit timer output control register 02	TM0	156
TOC03	16-bit timer output control register 03	TM0	156
TXB0	Transmit buffer register 0	UART	278
TXB1	Transmit buffer register 1	UART	278
VSWC	System wait control register	BCU	60
WDCS	Watchdog timer clock select register	WDT	249
WDRES	WDT reset status register	CG	253
WDTIC	Interrupt control register	INTC	325
WDTM	Watchdog timer mode register	WDT	250
WEEK	Week count register	RTC	244
WEEKB	Week count setting register	RTC	244
WEEKBH	Week count setting register H	RTC	54
WEEKBL	Week count setting register L	RTC	54
WEEKH	Week count register H	RTC	54
WEEKL	Week count register L	RTC	54

APPENDIX B INSTRUCTION SET LIST

B.1 Conventions

(1) Symbols used to describe operands

Symbol	Explanation
reg1	General-purpose registers: Used as source registers.
reg2	General-purpose registers: Used mainly as destination registers. Also used as source register in some instructions.
reg3	General-purpose registers: Used mainly to store the remainders of division results and the higher 32 bits of multiplication results.
bit#3	3-bit data for specifying the bit number
immX	X bit immediate data
dispX	X bit displacement data
regID	System register number
vector	5-bit data that specifies the trap vector (00H to 1FH)
cccc	4-bit data that shows the condition codes
sp	Stack pointer (r3)
ep	Element pointer (r30)
listX	X item register list

(2) Symbols used to describe opcodes

Symbol	Explanation
R	1-bit data of a code that specifies reg1 or regID
r	1-bit data of the code that specifies reg2
w	1-bit data of the code that specifies reg3
d	1-bit displacement data
I	1-bit immediate data (indicates the higher bits of immediate data)
i	1-bit immediate data
cccc	4-bit data that shows the condition codes
CCCC	4-bit data that shows the condition codes of Bcond instruction
bbb	3-bit data for specifying the bit number
L	1-bit data that specifies a program register in the register list

(3) Symbols used in operations

Symbol	Explanation
←	Input for
GR [ ]	General-purpose register
SR [ ]	System register
zero-extend (n)	Expand n with zeros until word length.
sign-extend (n)	Expand n with signs until word length.
load-memory (a, b)	Read size b data from address a.
store-memory (a, b, c)	Write data b into address a in size c.
load-memory-bit (a, b)	Read bit b of address a.
store-memory-bit (a, b, c)	Write c to bit b of address a.
saturated (n)	Execute saturated processing of n (n is a 2’s complement). If, as a result of calculations, n ≥ 7FFFFFFFH, let it be 7FFFFFFFH. n ≤ 80000000H, let it be 80000000H.
result	Reflects the results in a flag.
Byte	Byte (8 bits)
Half-word	Halfword (16 bits)
Word	Word (32 bits)
+	Addition
–	Subtraction
ll	Bit concatenation
×	Multiplication
÷	Division
%	Remainder from division results
AND	Logical product
OR	Logical sum
XOR	Exclusive OR
NOT	Logical negation
logically shift left by	Logical shift left
logically shift right by	Logical shift right
arithmetically shift right by	Arithmetic shift right

(4) Symbols used in execution clock

Symbol	Explanation
i	If executing another instruction immediately after executing the first instruction (issue).
r	If repeating execution of the same instruction immediately after executing the first instruction (repeat).
l	If using the results of instruction execution in the instruction immediately after the execution (latency).

(5) Symbols used in flag operations

Symbol	Explanation
(Blank)	No change
0	Clear to 0
×	Set or cleared in accordance with the results.
R	Previously saved values are restored.

(6) Condition codes

Condition Name (cond)	Condition Code (cccc)	Condition Formula	Explanation
V	0 0 0 0	OV = 1	Overflow
NV	1 0 0 0	OV = 0	No overflow
C/L	0 0 0 1	CY = 1	Carry Lower (Less than)
NC/NL	1 0 0 1	CY = 0	No carry Not lower (Greater than or equal)
Z	0 0 1 0	Z = 1	Zero
NZ	1 0 1 0	Z = 0	Not zero
NH	0 0 1 1	(CY or Z) = 1	Not higher (Less than or equal)
H	1 0 1 1	(CY or Z) = 0	Higher (Greater than)
S/N	0 1 0 0	S = 1	Negative
NS/P	1 1 0 0	S = 0	Positive
T	0 1 0 1	—	Always (Unconditional)
SA	1 1 0 1	SAT = 1	Saturated
LT	0 1 1 0	(S xor OV) = 1	Less than signed
GE	1 1 1 0	(S xor OV) = 0	Greater than or equal signed
LE	0 1 1 1	((S xor OV) or Z) = 1	Less than or equal signed
GT	1 1 1 1	((S xor OV) or Z) = 0	Greater than signed

B.2 Instruction Set (In Alphabetical Order)

(1/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
ADD	reg1,reg2	rrrrr001110RRRRR	GR[reg2]←GR[reg2]+GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010010iiiii	GR[reg2]←GR[reg2]+sign-extend(imm5)		1	1	1	×	×	×	×	
ADDI	imm16,reg1,reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1	×	×	×	×	
AND	reg1,reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]		1	1	1		0	×	×	
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]AND zero-extend(imm16)		1	1	1		0	×	×	
Bcond	disp9	ddddd1011dddcccc **Note 1**	if conditions are satisfied then PC←PC+sign-extend(disp9)	When conditions are satisfied	2 Note 2	2 Note 2	2 Note 2					
				When conditions are not satisfied	1	1	1					
BSH	reg2,reg3	rrrrr11111100000 wwwww01101000010	GR[reg3]←GR[reg2] (23 : 16) ll GR[reg2] (31 : 24) ll GR[reg2] (7 : 0) ll GR[reg2] (15 : 8)		1	1	1	×	0	×	×	
BSW	reg2,reg3	rrrrr11111100000 wwwww01101000000	GR[reg3]←GR[reg2] (7 : 0) ll GR[reg2] (15 : 8) ll GR[reg2] (23 : 16) ll GR[reg2] (31 : 24)		1	1	1	×	0	×	×	
CALLT	imm6	0000001000iiiiii	CTPC←PC+2(return PC) CTPSW←PSW adr←CTBP+zero-extend(imm6 logically shift left by 1) PC←CTBP+zero-extend(Load-memory(adr,Half-word))		4	4	4					
CLR1	bit#3, disp16[reg1]	10bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,0)		3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,0)		3 Note 3	3 Note 3	3 Note 3				×	
CMOV	cccc,imm5,reg2,reg3	rrrrr111111iiiii wwwww011000cccc0	if conditions are satisfied then GR[reg3]←sign-extend(imm5) else GR[reg3]←GR[reg2]		1	1	1					
	cccc,reg1,reg2,reg3	rrrrr111111RRRRR wwwww011001cccc0	if conditions are satisfied then GR[reg3]←GR[reg1] else GR[reg3]←GR[reg2]		1	1	1					
CMP	reg1,reg2	rrrrr001111RRRRR	result←GR[reg2]–GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010011iiiii	result←GR[reg2]–sign-extend(imm5)		1	1	1	×	×	×	×	
CTRET		0000011111100000 0000000101000100	PC←CTPC PSW←CTPSW		3	3	3	R	R	R	R	R
DBRET		0000011111100000 0000000101000110	PC←DBPC PSW←DBPSW		3	3	3	R	R	R	R	R

(2/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
DBTRAP		1111100001000000	DBPC←PC+2 (returned PC) DBPSW←PSW PSW.NP←1 PSW.EP←1 PSW.ID←1 PC←00000060H	3	3	3					
DI		0000011111100000 0000000101100000	PSW.ID←1	1	1	1					
DISPOSE	imm5,list12	0000011001iiiiiL LLLLLLLLLLL00000	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	imm5,list12[reg1]	0000011001iiiiiL LLLLLLLLLLLRRRRR **Note 5**	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded PC←GR[reg1]	n+3 Note 4	n+3 Note 4	n+3 Note 4					
DIV	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000000	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVH	reg1,reg2	rrrrr000010RRRRR	GR[reg2]←GR[reg2]÷GR[reg1][Note 6]	35	35	35		×	×	×	
	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000000	GR[reg2]←GR[reg2]÷GR[reg1][Note 6] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVHU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000010	GR[reg2]←GR[reg2]÷GR[reg1][Note 6] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
DIVU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000010	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
EI		1000011111100000 0000000101100000	PSW.ID←0	1	1	1					
HALT		0000011111100000 0000000100100000	Stop	1	1	1					
HSW	reg2,reg3	rrrrr11111100000 wwwww01101000100	GR[reg3]←GR[reg2](15 : 0) ll GR[reg2] (31 : 16)	1	1	1	×	0	×	×	
JARL	disp22,reg2	rrrrr11110dddddd ddddddddddddddd0 **Note 7**	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	2	2	2					
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]	3	3	3					
JR	disp22	0000011110dddddd ddddddddddddddd0 **Note 7**	PC←PC+sign-extend(disp22)	2	2	2					
LD.B	disp16[reg1],reg2	rrrrr111000RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	**Note 11**					
LD.BU	disp16[reg1],reg2	rrrrr11110bRRRRR ddddddddddddddd1 **Notes 8, 10**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	**Note 11**					

(3/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
LD.H	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Half-word))	1	1	Note 11					
LDSR	reg2,regID	rrrrr111111RRRRR 0000000000100000 **Note 12**	SR[regID]←GR[reg2] Other than regID = PSW	1	1	1					
			regID = PSW	1	1	1	×	×	×	×	×
LD.HU	disp16[reg1],reg2	rrrrr111111RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Half-word)	1	1	Note 11					
LD.W	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←Load-memory(adr,Word)	1	1	Note 11					
MOV	reg1,reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]	1	1	1					
	imm5,reg2	rrrrr010000iiiii	GR[reg2]←sign-extend(imm5)	1	1	1					
	imm32,reg1	00000110001RRRRR iiiiiiiiiiiiiiii IIIIIIIIIIIIIIII	GR[reg1]←imm32	2	2	2					
MOVEA	imm16,reg1,reg2	rrrrr110001RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)	1	1	1					
MOVHI	imm16,reg1,reg2	rrrrr110010RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+(imm16 ll 0^{16})	1	1	1					
MUL	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1] **Note 14**	1	4	5					
	imm9,reg2,reg3	rrrrr111111iiiii wwwww01001IIII00 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xsign-extend(imm9)	1	4	5					
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2][Note 6]xGR[reg1][Note 6]	1	1	2					
	imm5,reg2	rrrrr010111iiiii	GR[reg2]←GR[reg2][Note 6]xsign-extend(imm5)	1	1	2					
MULHI	imm16,reg1,reg2	rrrrr110111RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1][Note 6]ximm16	1	1	2					
MULU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100010	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1] **Note 14**	1	4	5					
	imm9,reg2,reg3	rrrrr111111iiiii wwwww01001IIII10 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xzero-extend(imm9)	1	4	5					
NOP		0000000000000000	Pass at least one clock cycle doing nothing.	1	1	1					
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])	1	1	1		0	×	×	
NOT1	bit#3,disp16[reg1]	01bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,Z flag)	3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,Z flag)	3 Note 3	3 Note 3	3 Note 3				×	

(4/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
OR	reg1,reg2	rrrrr001000RRRRR	GR[reg2]←GR[reg2]OR GR[reg1]	1	1	1		0	×	×	
ORI	imm16,reg1,reg2	rrrrr110100RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]OR zero-extend(imm16)	1	1	1		0	×	×	
PREPARE	list12,imm5	0000011110iiiiiL LLLLLLLLLLL00001	Store-memory(sp–4,GR[reg in list12],Word) sp←sp–4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5)	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	list12,imm5, sp/imm[Note 15]	0000011110iiiiiL LLLLLLLLLLLff011 imm16/imm32 **Note 16**	Store-memory(sp–4,GR[reg in list12],Word) sp←sp–4 repeat 1 step above until all regs in list12 is stored sp←sp–zero-extend (imm5) ep←sp/imm	n+2 Note 4 Note17	n+2 Note 4 Note17	n+2 Note 4 Note17					
RETI		0000011111100000 0000000101000000	if PSW.EP=1 then PC ←EIPC PSW ←EIPSW else if PSW.NP=1 then PC ←FEPC PSW ←FEPSW else PC ←EIPC PSW ←EIPSW	3	3	3	R	R	R	R	R
SAR	reg1,reg2	rrrrr111111RRRRR 0000000010100000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010101iiiii	GR[reg2]←GR[reg2]arithmetically shift right by zero-extend (imm5)	1	1	1	×	0	×	×	
SASF	cccc,reg2	rrrrr1111110cccc 0000001000000000	if conditions are satisfied then GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000001H else GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000000H	1	1	1					
SATADD	reg1,reg2	rrrrr000110RRRRR	GR[reg2]←saturated(GR[reg2]+GR[reg1])	1	1	1	×	×	×	×	×
	imm5,reg2	rrrrr010001iiiii	GR[reg2]←saturated(GR[reg2]+sign-extend(imm5))	1	1	1	×	×	×	×	×
SATSUB	reg1,reg2	rrrrr000101RRRRR	GR[reg2]←saturated(GR[reg2]–GR[reg1])	1	1	1	×	×	×	×	×
SATSUBI	imm16,reg1,reg2	rrrrr110011RRRRR iiiiiiiiiiiiiiii	GR[reg2]←saturated(GR[reg1]–sign-extend(imm16))	1	1	1	×	×	×	×	×
SATSUBR	reg1,reg2	rrrrr000100RRRRR	GR[reg2]←saturated(GR[reg1]–GR[reg2])	1	1	1	×	×	×	×	×
SETF	cccc,reg2	rrrrr1111110cccc 0000000000000000	if conditions are satisfied then GR[reg2]←00000001H else GR[reg2]←00000000H	1	1	1					

(5/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SET1	bit#3,disp16[reg1]	00bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,1)	3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100000	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,1)	3 Note 3	3 Note 3	3 Note 3				×	
SHL	reg1,reg2	rrrrr111111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010110iiiii	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	×	0	×	×	
SHR	reg1,reg2	rrrrr111111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010100iiiii	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	×	0	×	×	
SLD.B	disp7[ep],reg2	rrrrr0110ddddddd	adr←ep+zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.BU	disp4[ep],reg2	rrrrr0000110dddd **Note 18**	adr←ep+zero-extend(disp4) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.H	disp8[ep],reg2	rrrrr1000ddddddd **Note 19**	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr,Half-word))	1	1	Note 9					
SLD.HU	disp5[ep],reg2	rrrrr0000111dddd **Notes 18, 20**	adr←ep+zero-extend(disp5) GR[reg2]←zero-extend(Load-memory(adr,Half-word))	1	1	Note 9					
SLD.W	disp8[ep],reg2	rrrrr1010dddddd0 **Note 21**	adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr,Word)	1	1	Note 9					
SST.B	reg2,disp7[ep]	rrrrr0111ddddddd	adr←ep+zero-extend(disp7) Store-memory(adr,GR[reg2],Byte)	1	1	1					
SST.H	reg2,disp8[ep]	rrrrr1001ddddddd **Note 19**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Half-word)	1	1	1					
SST.W	reg2,disp8[ep]	rrrrr1010dddddd1 **Note 21**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Word)	1	1	1					
ST.B	reg2,disp16[reg1]	rrrrr111010RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Byte)	1	1	1					
ST.H	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Half-word)	1	1	1					
ST.W	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Word)	1	1	1					
STSR	regID,reg2	rrrrr111111RRRRR 0000000001000000	GR[reg2]←SR[regID]	1	1	1					

(6/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1,reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]–GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1,reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]–GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	00000000010RRRRR	adr←(PC+2) + (GR[reg1] logically shift left by 1) PC←(PC+2) + (sign-extend (Load-memory (adr,Half-word))) logically shift left by 1	5	5	5					
SXB	reg1	00000000101RRRRR	GR[reg1]←sign-extend(GR[reg1] (7 : 0))	1	1	1					
SXH	reg1	00000000111RRRRR	GR[reg1]←sign-extend(GR[reg1] (15 : 0))	1	1	1					
TRAP	vector	00000111111iiiii 0000000100000000	EIPC ←PC+4 (Return PC) EIPSW ←PSW ECR.EICC ←Interrupt Code PSW.EP ←1 PSW.ID ←1 PC ←00000040H (when vector is 00H to 0FH) 00000050H (when vector is 10H to 1FH)	3	3	3					
TST	reg1,reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1		0	×	×	
TST1	bit#3,disp16[reg1]	11bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit (adr,bit#3))	3 Note 3	3 Note 3	3 Note 3				×	
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100110	adr←GR[reg1] Z flag←Not (Load-memory-bit (adr,reg2))	3 Note 3	3 Note 3	3 Note 3				×	
XOR	reg1,reg2	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	×	×	
XORI	imm16,reg1,reg2	rrrrr110101RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] XOR zero-extend (imm16)	1	1	1		0	×	×	
ZXB	reg1	00000000100RRRRR	GR[reg1]←zero-extend (GR[reg1] (7 : 0))	1	1	1					
ZXH	reg1	00000000110RRRRR	GR[reg1]←zero-extend (GR[reg1] (15 : 0))	1	1	1					

Notes 1. dddddddd: Higher 8 bits of disp9.

2. 3 if there is an instruction that rewrites the contents of the PSW immediately before.

3. If there is no wait state (3 + the number of read access wait states).

4. n is the total number of list12 load registers. (According to the number of wait states. Also, if there are no wait states, n is the number of list12 registers. If n = 0, same operation as when n = 1)

5. RRRRR: other than 00000.

6. The lower halfword data only are valid.

7. ddddddddddddddddddddd: The higher 21 bits of disp22.

8. ddddddddddddddd: The higher 15 bits of disp16.

9. According to the number of wait states (1 if there are no wait states).

10. b: bit 0 of disp16.

11. According to the number of wait states (2 if there are no wait states).

Notes 12. In this instruction, for convenience of mnemonic description, the source register is made reg2, but the reg1 field is used in the opcode. Therefore, the meaning of register specification in the mnemonic description and in the opcode differs from other instructions.

rrrrr = regID specification
RRRRR = reg2 specification

13. iiiii: Lower 5 bits of imm9.
IIII: Higher 4 bits of imm9.

14. Do not specify the same register for general-purpose registers reg1 and reg3.

15. sp/imm: specified by bits 19 and 20 of the sub-opcode.

16. ff = 00: Load sp in ep.
01: Load sign expanded 16-bit immediate data (bits 47 to 32) in ep.
10: Load 16-bit logically left shifted 16-bit immediate data (bits 47 to 32) in ep.
11: Load 32-bit immediate data (bits 63 to 32) in ep.

17. If imm = imm32, n + 3 clocks.

18. rrrrr: Other than 00000.

19. ddddddd: Higher 7 bits of disp8.

20. dddd: Higher 4 bits of disp5.

21. dddddd: Higher 6 bits of disp8.

★

APPENDIX C REVISION HISTORY

C.1 Major Revisions in This Edition

(1/2)

Page	Description
p.28	Modification of **Table 2-2 Operating Status of Each Pin in Each Operation Mode**
p.29	Modification of I/O Circuit Type of P98 to P915 in **2.3 Types of Pin I/O Circuits, I/O Buffer Power Supplies, and Connection of Unused Pins**
p.31	Modification of **Figure 2-1 Pin I/O Circuits**
p.61	Addition of **3.4.8 (2) Access to special on-chip peripheral I/O registers**
p.99	Addition of **4.4 Block Diagram**
p.120	Addition of **4.6 Cautions**
p.122	Modification of **5.2.1 Pin status when internal ROM, internal RAM, or on-chip peripheral I/O is accessed**
p.135	Addition of **Caution** to **5.5.4 (1) Address wait control register (AWC)**
p.144	Addition of **Caution 3** to **6.3 (1) Processor clock control register (PCC)**
p.145	Modification of **6.3. (1) (a) Example of setting main clock operation → subclock operation**
p.146	Modification of **6.3. (1) (b) Example of setting subclock operation → main clock operation**
p.147	Addition of **6.4.3 External clock input function**
p.152	Addition of **Caution 2** to **7.2 (3) 16-bit timer capture/compare register 0n1 (CR0n1)**
p.155	Modification of **Caution 3** in **7.3 (2) Capture/compare control register 0n (CRC0n)**
p.157	Addition of **Caution 6** to **7.3 (3) 16-bit timer output control register 0n (TOC0n)**
p.160	Addition of **Setting method** to **7.4.1 Operation as interval timer (16 bits)**
p.162	Modification of **Figure 7-4 Timing of Interval Timer Operation**
p.163	Addition of **Setting method** to **7.4.2 PPG output operation**
p.165	Addition of **Figure 7-6 Configuration of PPG Output**
p.166	Addition of **Figure 7-7 PPG Output Operation Timing**
p.167	Addition of **Setting method** to **7.4.3 Pulse width measurement**
p.175	Addition of **Setting method** to **7.4.4 Operation as external event counter**
p.178	Addition of **Setting method** to **7.4.5 Square-wave output operation**
p.181	Addition of **Setting method** to **7.4.6 One-shot pulse output operation**
p.187	Addition of **<2>** to **7.4.7 (3) Data hold timing of capture register**
p.189	Modification of **(b)** in **7.4.7 (8) Capture operation**
p.192	Addition of **Caution 3** to **8.3 (1) 16-bit timer counters 10 and 11 (TM10 and TM11)**
p.194	Addition of **Caution** to **8.3 (2) 16-bit timer capture/compare registers 1n0 and 1n1 (CC1n0 and CC1n1)**
p.196	Addition of **Caution 3** to **8.4 (1) 16-bit timer mode control registers 100 and 110 (TMC100 and TMC110)**
p.214	Addition of **Note** to **8.6 (4) Cycle measurement**
p.250	Addition of **Caution** to **11.3 (2) Watchdog timer mode register (WDTM)**
p.261	Addition of **(7)** to **12.5 Cautions**
p.264	Addition of **Caution** to **13.3 (2) PWM buffer register n (PWMBn)**

(2/2)

Page	Description
p.265	Modification of **Figure 13-2 PWMn Operation Timing**
p.266	Modification of **Figure 13-3 Operation Timing When PWMBn Register Is Set to 00H/FFH**
p.272	Addition of **Caution 2** to **14.3 (1) Asynchronous serial interface mode register n (ASIMn)**
p.275	Addition of **Caution 2** to **14.3 (2) Asynchronous serial interface status register n (ASISn)**
p.283	Addition of **Caution** to **14.5 (3) Continuous transmission operation**
p.288	Modification of **Figure 14-7 Asynchronous Serial Interface Reception Completion Interrupt Timing**
p.344	Addition of **Note** to **17.2 (1) Power save control register (PSC)**
p.346	Addition of **Caution 2** to **17.3.1 Setting and operation status**
p.347	Modification of **Note** in **Table 17-3 Operation Status in HALT Mode**
p.354	Addition of **17.7.3 Registers to which access is disabled in subclock operation mode**
p.354	Addition of **Note** to **Table 17-8 Operation Status in Subclock Operation Mode**
p.361	Addition of **Note 2** and modification of **Note 3** in **Table 18-1 Hardware Status on $\overline{\text{RESET}}$ Pin Input**
p.367	Correction of **Figure 19-2 ROM Correction Operation and Program Flow**
p.371	Addition of oscillation frequency to **Recommended oscillator (2) External clock input**
p.372	Modification of description of oscillation stabilization time in **Recommended oscillator (3) Subclock oscillator**
p.373	Modification of conditions of power supply voltage in **DC characteristics 1 (a)**
p.373	Modification of **DC characteristics 1 (b)**
p.374	Modification of conditions of power supply voltage in **DC characteristics 2**
p.374	Modification of I_{DD1} and addition of **Note 2** to I_{DD3} in **DC characteristics 2**
p.374	Addition of **Note 3** to I_{DD4}, I_{DD5}, and I_{DD6} in **DC characteristics 2**
p.375	Addition of **Note 1** to I_{DDDR} in **Data retention characteristics**
pp.378 to 380	Modification of t_{CYX}, t_{WXH}, and t_{WXL} of **(1)**, and t_{CYK}, t_{WKH}, and t_{WKL} of **(2)**, addition of **(2) (b)**, modification of **(2) (c)** and **(3) (b)**, and addition of **(3) (c)** in **Clock timing**
pp.382 and 384	Modification of **(1) (b)** and **(2) (b)** in **Bus timing**
p.387	Modification of conditions of power supply voltage in **Reset/interrupt timing (a)**
p.387	Modification of **Reset/interrupt timing (b)**
p.389	Modification of conditions of power supply voltage in **Timer timing (a)**
p.390	Modification of **CSI timing**
p.394	Modification of internal reference potential temperature coefficient and addition of **Note** in **(2) Reference** in **A/D converter characteristics**
p.395	Modification of S/N and operating current and addition of **Remarks 2** and **3** in **(4) A/D converter and system specifications** in **A/D converter characteristics**
p.398	Addition of **CHAPTER 22 RECOMMENDED SOLDERING CONDITIONS**
p.415	Addition of **APPENDIX C REVISION HISTORY**