NEC

User's Manual

V850E/IA2

32-Bit Single-Chip Microcontrollers

Hardware

μPD703114
μPD703114(A)
μPD70F3114
μPD70F3114(A)

Document No. U15195EJ5V0UD00 (5th edition)
Date Published August 2005 N CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (MAX) and V_{IH} (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (MAX) and V_{IH} (MIN).

② HANDLING OF UNUSED INPUT PINS

Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

③ PRECAUTION AGAINST ESD

A 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 when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up 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 benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

④ STATUS BEFORE INITIALIZATION

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

⑤ POWER ON/OFF SEQUENCE

In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current.

The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

⑥ INPUT OF SIGNAL DURING POWER OFF STATE

Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements.

Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

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 March, 2005. 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: 0211-65030

- **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: 040-265 40 10
- **Tyskland Filial**
  Taeby, Sweden
  Tel: 08-63 87 200
- **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: 021-5888-5400

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

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

J05.6

PREFACE

Readers

This manual is intended for users who wish to understand the functions of the V850E/IA2 and design application systems using it.
The target products are as follows.

- Standard products: μPD703114, 70F3114
- Special grade products: μPD703114(A), 70F3114(A)

Purpose

This manual is intended to give users an understanding of the hardware functions of the V850E/IA2 shown in the **Organization** below.

Organization

This manual is divided into two parts: Hardware (this manual) and Architecture **(V850E1 Architecture User's Manual).**

Hardware	Architecture
• Pin functions	• Data type
• CPU function	• Register set
• On-chip peripheral functions	• Instruction format and instruction set
• Flash memory programming	• Interrupts and exceptions
• Electrical specifications	• 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.

Cautions
1. **The application examples in this manual apply to "standard" quality grade products for general electronic systems. When using an example in this manual for an application that requires a "special" quality grade product, thoroughly evaluate the component and circuit to be actually used to see if they satisfy the special quality grade.**
2. **When using this manual as a manual for a special grade product, read the part numbers as follows.**
   μPD703114 → 703114(A)
   μPD70F3114 → 70F3114(A)

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

- To understand the details of an instruction function
  → Refer to the **V850E1 Architecture User's Manual**.

- To know details of the electrical specifications of the V850E/IA2
  → Refer to **CHAPTER 16 ELECTRICAL SPECIFICATIONS**.

- To understand the overall functions of the V850E/IA2
  → Read this manual according to the **CONTENTS**.

- How to read register formats
  → The name of a bit whose number is in angle brackets (<>) is defined as a reserved word in the device file.
  When the register format of each register describes 0 or 1, other values are prohibited to be specified.

The mark ★ shows major revised points.

Conventions

Data significance:	Higher digits on the left and lower digits on the right
Active low representation:	$\overline{xxx}$ (overscore over pin or signal name)
Memory map address:	Top: higher, bottom: lower
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): $2^{30} = 1,024^3$
Data type:	Word ... 32 bits
	Halfword ... 16 bits
	Byte ... 8 bits

Related Documents

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

Documents related to V850E/IA2

Document Name	Document No.
V850E1 Architecture User's Manual	U14559E
V850E/IA2 Hardware User's Manual	This manual
V850E/IA1, V850E/IA2 AC Motor Inverter Control Using Vector Operation Application Note	U14868E
Inverter Control by V850 Series 120° Excitation Method Control by Zero-Cross Detection Application Note	U17209E
Inverter Control by V850 Series Vector Control by Encoder Application Note	U17324E
Inverter Control by V850 Series Vector Control by Hole Sensor Application Note	U17338E

Documents related to development tools (user's manuals)

Document Name		Document No.
IE-V850E-MC, IE-V850E-MC-A (In-circuit emulator)		U14487E
IE-703114-MC-EM1 (In-circuit emulator option board)		U16533E
CA850 (Ver. 3.00) (C compiler package)	Operation	U17293E
	C Language	U17291E
	Assembly Language	U17292E
	Link Directives	U17294E
PM+ (Ver. 6.00) (Project manager)		U17178E
ID850 (Ver. 3.00) (Integrated debugger)	Operation	U17358E
TW850 (Ver. 2.00) (Performance analysis tuning tool)		U17241E
SM850 (Ver. 2.50) (System simulator)	Operation	U16218E
SM850 (Ver. 2.00 or later) (System simulator)	External Part User Open Interface Specification	U14873E
SM+ (System simulator)	Operation	U17246E
	User Open Interface	U17247E
RX850 (Ver. 3.13 or later) (Real-time OS)	Basics	U13430E
	Installation	U13410E
	Technical	U13431E
RX850 Pro (Ver. 3.15) (Real-time OS)	Basics	U13773E
	Installation	U13774E
	Technical	U13772E
RD850 (Ver. 3.01) (Task debugger)		U13737E
RD850 Pro (Ver. 3.01) (Task debugger)		U13916E
AZ850 (Ver. 3.10) (System performance analyzer)		U14410E
PG-FP4 Flash memory programmer		U15260E

CONTENTS

CHAPTER 1 INTRODUCTION

The V850E/IA2 is a product in NEC Electronics' V850 Series of single-chip microcontrollers. This chapter provides an overview of the V850E/IA2.

1.1 Outline

The V850E/IA2 is a 32-bit single-chip microcontroller that uses high-speed operations to realize high-precision inverter control of motors. It uses the V850E1 CPU of the V850 Series and has on-chip peripheral functions such as ROM, RAM, a bus interface, a DMA controller, timers including a 3-phase sine-wave PWM timer for motors, serial interfaces, and A/D converters.

(1) V850E1 CPU

The V850E1 CPU supports a RISC instruction set in which the instruction execution speed is increased greatly through the use of basic instructions that execute one instruction per clock, and an optimized pipeline. Moreover, it supports multiply instructions using a 32-bit hardware multiplier, saturated product-sum operation instructions, and bit manipulation instructions as optimum instructions for digital servo control applications.

Object code efficiency is increased in the C compiler by using 2-byte-length basic instructions and instructions corresponding to high-level languages, which promote a compact program.

Furthermore, since the interrupt response time, including processing by the on-chip interrupt controller, is also fast, this CPU is ideal for advanced real-time control.

(2) External bus interface function

A bus configuration consisting of a multiplexed address bus (22 bits) and data bus (8 bits or 16 bits selectable) suitable for compact system design is used as the external bus interface. SRAM and ROM memories can be connected.

In the DMA controller, transfer is started using software and transfers between external memories can be made concurrent with internal CPU operations or data transfers. Real-time control such as motor control or communication control can also be realized simultaneously due to high-speed, high-performance CPU instruction execution.

(3) On-chip flash memory (μPD70F3114)

The on-chip flash memory version (μPD70F3114), which has a quickly accessible flash memory on-chip, can shorten system development time since it is possible to rewrite a program with the V850E/IA2 mounted in an application system. Moreover, it can greatly improve maintainability after a system is shipped.

(4) Complete middleware, development environment

The V850E/IA2 can execute JPEG, JBIG, MH/MR/MMR and other middleware at high speeds. Moreover, since middleware for realizing speech recognition, voice synthesis, and other processing also is provided, multimedia systems can be realized easily.

A development environment that integrates an optimized C compiler, debugger, in-circuit emulator, simulator, and system performance analyzer is also provided.

Table 1-1 lists the differences between the V850E/IA1 and V850E/IA2. Table 1-2 lists the differences between the V850E/IA1 and V850E/IA2 register setting values.

Table 1-1. Differences Between V850E/IA1 and V850E/IA2

Item		V850E/IA1	V850E/IA2
★ Maximum operating frequency		50 MHz[Note]	40 MHz
Internal ROM	Mask ROM	μPD703116: 256 KB	μPD703114: 128 KB
	Flash memory	μPD70F3116: 256 KB	μPD70F3114: 128 KB
Internal RAM		10 KB	6 KB
Timer	Timer 00, 01	Provided	Buffer register, compare register, and compare match interrupt added
	Timer 10, 11	Provided	Timer 10: Provided, Timer 11: Not provided
	Timer 20, 21	Provided	Provided
	Timer 3	Provided	TO3 output buffer off function added by INTP4 input
	Timer 4	Provided	Provided
Serial interface	UART0	Provided	Provided
	UART1	Provided	Provided (pins multiplexed with CSI1)
	UART2	Provided	Not provided
	CSI0	Provided	Provided
	CSI1	Provided	Provided (pins multiplexed with UART1)
	FCAN	Provided	Not provided
Debug support function	NBD	Provided	Not provided
A/D converter	Analog input	Total of two circuits: 16 ch A/D converter 0: 8 ch A/D converter 1: 8 ch	Total of two circuits: 14 ch A/D converter 0: 6 ch A/D converter 1: 8 ch
	AV_{DD}, AV_{REF} pins	Independent pins	Alternate-function pins
Supply voltage		V_{DD3} = 3.3 V ±0.3 V V_{DD5} = 5.0 V ±0.5 V	V_{DD} = RV_{DD} = 5.0 V ±0.5 V Internal regulator
Package		144-pin plastic LQFP	100-pin plastic LQFP 100-pin plastic QFP

★ **Note** The maximum operating frequency of the in-circuit emulator is 40 MHz. A frequency of 50 MHz can be supported by upgrading the in-circuit emulator, so contact an NEC Electronics sales representative or distributor.

Remark For details, refer to the user's manual of each product.

Table 1-2. Differences Between V850E/IA1 and V850E/IA2 Register Setting Values

Register Name	V850E/IA1	V850E/IA2
System wait control register (VSWC)	12H	02H
Timer 1/timer 2 clock selection register (PRM02)	00H or 01H	01H (initial value 00H)

Remark For details, refer to the user's manual of each product.

1.2 Features

★ ○ Number of instructions 80

○ Minimum instruction execution time
25 ns (@ internal 40 MHz operation)

○ General-purpose registers 32 bits × 32 registers

○ Instruction set
V850E1 (NB85E) CPU
Signed multiplication (32 bits × 32 bits → 64 bits): 1 or 2 clocks
Saturated operation instructions (with overflow/underflow detection function)
32-bit shift instruction: 1 clock
Bit manipulation instructions
Long/short format load/store instructions
Signed load instructions

○ Memory space
4 MB linear address space (shared by program and data)
Memory block division function: 2 MB/block
Programmable wait function
Idle state insertion function

○ External bus interface
16-bit data bus (address/data multiplexed)
16-/8-bit bus sizing function
External wait function

○ Internal memory

Part Number	Internal ROM	Internal RAM
μPD703114	128 KB (mask ROM)	6 KB
μPD70F3114	128 KB (flash memory)	6 KB

○ Interrupts/exceptions
External interrupts: 16 (including NMI)
Internal interrupts: 42 sources
Exceptions: 1 source
8 levels of priority can be specified

- DMA controller
  - 4-channel configuration
  - Transfer unit: 8 bits/16 bits
  - Maximum transfer count: 65,536 (2^{16})
  - Transfer type: 2-cycle transfer
  - Transfer modes: Single transfer, single-step transfer, block transfer
  - Transfer subjects: Memory ↔ Memory, Memory ↔ I/O, I/O ↔ I/O
  - Transfer requests: On-chip peripheral I/O, software
  - Next address setting function
- I/O lines
  - Input ports: 6
  - I/O ports: 47
- Timer/counter function
  - 16-bit timer for 3-phase sine wave PWM inverter control: 2 channels
  - 16-bit up/down counter/timer for 2-phase encoder input: 1 channel
  - General-purpose 16-bit timer/counter: 2 channels
  - General-purpose 16-bit timer/event counter: 1 channel
  - 16-bit interval timer: 1 channel
- Serial interface
  - Asynchronous serial interface (UART): 2 channels
  - Clocked serial interface (CSI): 2 channels
  - Of the four channels, two channels are used for both CSI and UART and therefore one or the other function must be selected.
- A/D converter
  - 10-bit resolution A/D converter: 6 channels + 8 channels (2 units)
- Regulator
  - Two power supplies, one for the internal CPU and one for the peripheral interface, are not necessary. A 5 V single-power-supply system can be configured by connecting an N-ch transistor (2SD1950 (VL standard product, surface mount type) or 2SD1581 (independent type) is recommended). If a 3.3 V power supply is available, it can be directly connected to the REGIN pin.
- Clock generator
  - Multiplication function (×1, ×2.5, ×5, ×10) using PLL clock synthesizer
  - Divide-by-2 function using external clock input
- Power-saving function
  - HALT, IDLE, and software STOP modes
- Package
  - 100-pin plastic LQFP (fine pitch) (14 × 14)
  - 100-pin plastic QFP (14 × 20)
- CMOS technology
  - Fully static circuits

1.3 Applications

- μPD703114, 70F3114: Consumer equipment (inverter air conditioners)
  Industrial equipment (motor control, general-purpose inverters)
- μPD703114(A), 70F3114(A): Automobile applications (electrical power steering)

1.4 Ordering Information

	Part Number	Package	Quality Grade
	μPD703114GC-×××-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Standard
★	μPD703114GC-×××-8EU-A	100-pin plastic LQFP (fine pitch) (14 × 14)	Standard
	μPD703114GC(A)-×××-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Special
★	μPD703114GC(A)-×××-8EU-A	100-pin plastic LQFP (fine pitch) (14 × 14)	Special
	μPD703114GF-×××-3BA	100-pin plastic QFP (14 × 20)	Standard
★	μPD703114GF-×××-3BA-A	100-pin plastic QFP (14 × 20)	Standard
	μPD70F3114GC-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Standard
★	μPD70F3114GC-8EU-A	100-pin plastic LQFP (fine pitch) (14 × 14)	Standard
	μPD70F3114GC(A)-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Special
★	μPD70F3114GC(A)-8EU-A	100-pin plastic LQFP (fine pitch) (14 × 14)	Special
	μPD70F3114GF-3BA	100-pin plastic QFP (14 × 20)	Standard
★	μPD70F3114GF-3BA-A	100-pin plastic QFP (14 × 20)	Standard

Remarks 1. ××× indicates ROM code suffix.

★ **2.** Products with -A at the end of the part number are lead-free products.

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

1.5 Pin Configuration (Top View)

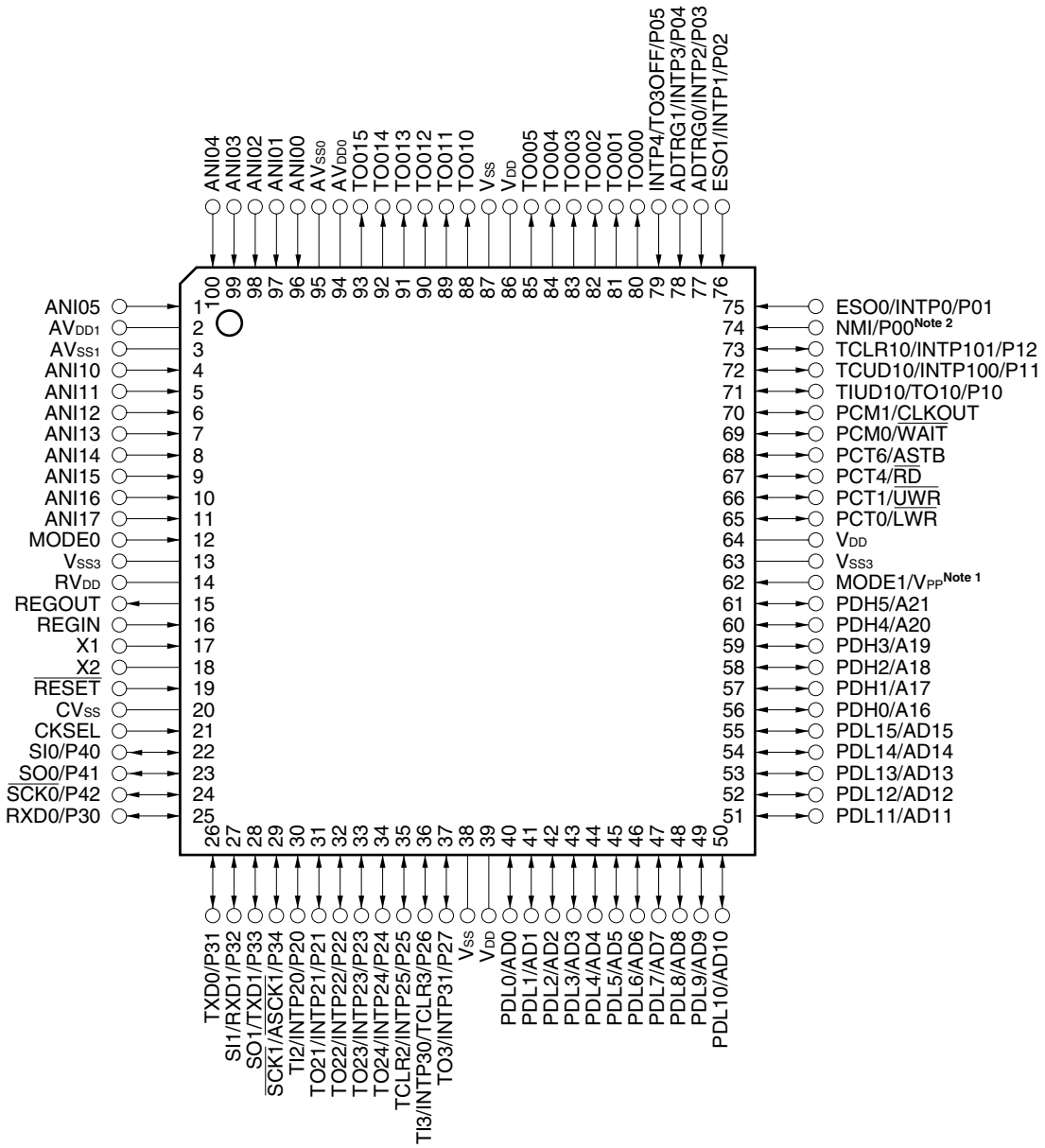
- **100-pin plastic LQFP (fine pitch) (14 × 14)**

μPD703114GC-×××-8EU μPD70F3114GC-8EU

★ μPD703114GC-×××-8EU-A μPD70F3114GC-8EU-A

μPD703114GC(A)-×××-8EU μPD70F3114GC(A)-8EU

★ μPD703114GC(A)-×××-8EU-A μPD70F3114GC(A)-8EU-A



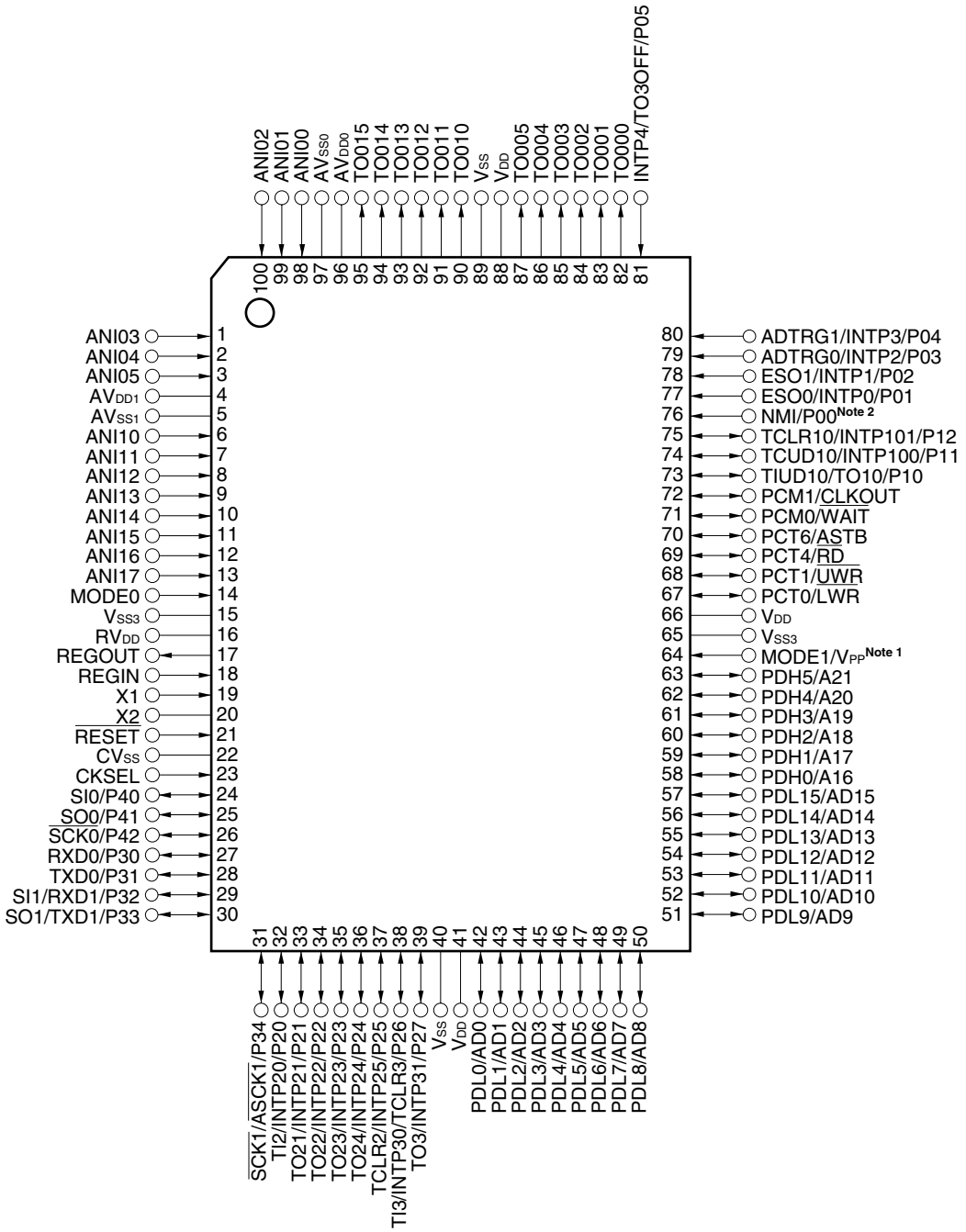
Notes 1. μPD70F3114 only

2. The NMI/P00 pin always functions as the NMI pin. The level of the NMI pin can be read by reading the P0.P00 bit.

- **100-pin plastic QFP (14 × 20)**

  μPD703114GF-×××-3BA μPD70F3114GF-3BA

★ μPD703114GF-×××-3BA-A μPD70F3114GF-3BA-A



Notes 1. μPD70F3114 only

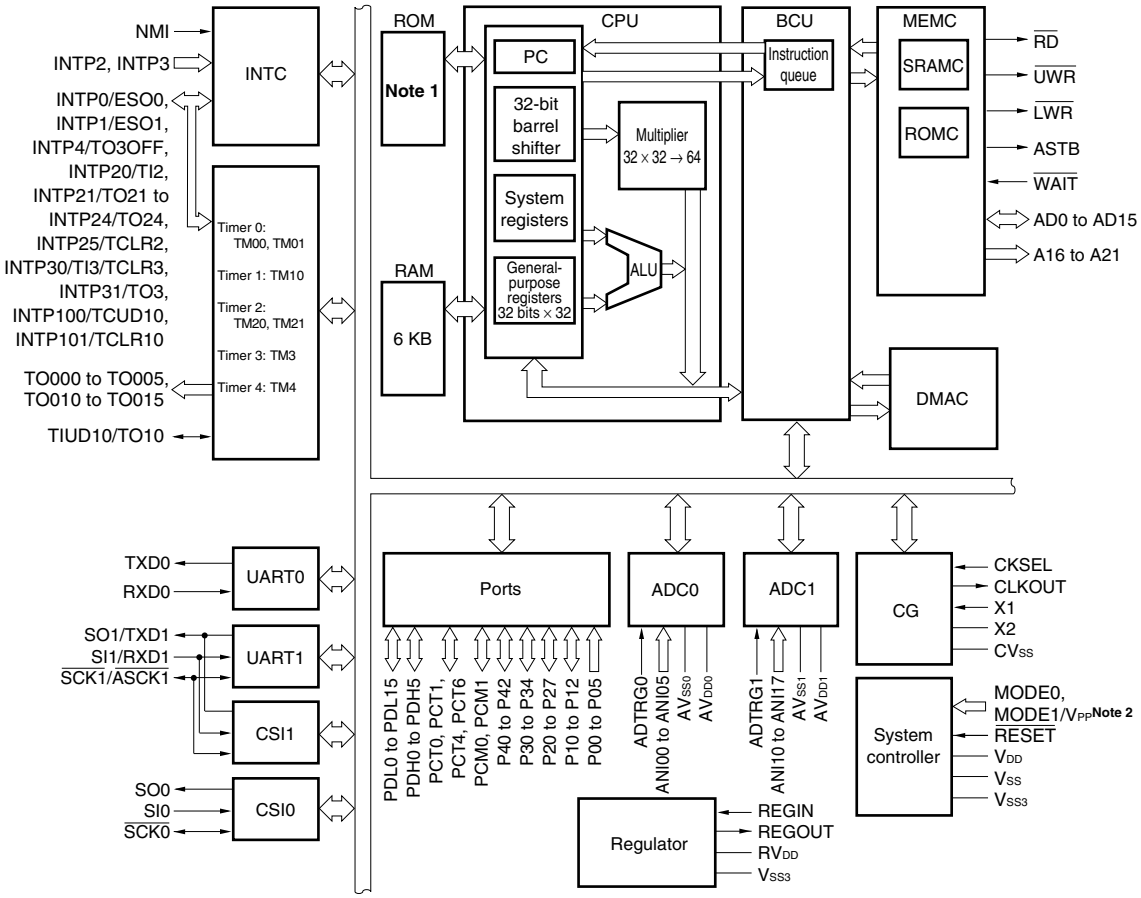
2. The NMI/P00 pin always functions as the NMI pin. The level of the NMI pin can be read by reading the P0.P00 bit.

Pin Identification

A16 to A21:	Address bus
AD0 to AD15:	Address/data bus
ADTRG0, ADTRG1:	A/D trigger input
ANI00 to ANI05, ANI10 to ANI17:	Analog input
$\overline{\text{ASCK1}}$:	Asynchronous serial clock
ASTB:	Address strobe
AV_{DD0}, AV_{DD1}:	Analog power supply
AV_{SS0}, AV_{SS1}:	Analog ground
CKSEL:	Clock generator operating mode select
CLKOUT:	Clock output
CV_{SS}:	Clock generator ground
ESO0, ESO1:	Emergency shut off
INTP0 to INTP4, INTP100, INTP101, INTP20 to INTP25, INTP30, INTP31:	External interrupt input
$\overline{\text{LWR}}$:	Lower write strobe
MODE0, MODE1:	Mode
NMI:	Non-maskable interrupt request
P00 to P05:	Port 0
P10 to P12:	Port 1
P20 to P27:	Port 2
P30 to P34:	Port 3
P40 to P42:	Port 4
PCM0, PCM1:	Port CM
PCT0, PCT1, PCT4, PCT6:	Port CT
PDH0 to PDH5:	Port DH
PDL0 to PLD15:	Port DL
$\overline{\text{RD}}$:	Read strobe
$\overline{\text{RESET}}$:	Reset
REGIN:	Regulator input
REGOUT:	Regulator output
RV_{DD}:	Regulator power supply
RXD0, RXD1:	Receive data
$\overline{\text{SCK0}}$, $\overline{\text{SCK1}}$:	Serial clock
SI0, SI1:	Serial input
SO0, SO1:	Serial output
TCLR10, TCLR2, TCLR3:	Timer clear
TCUD10:	Timer control pulse input
TI2, TI3:	Timer input
TIUD10:	Timer count pulse input
TO000 to TO005, TO010 to TO015, TO10, TO21 to TO24, TO3:	Timer output
TO3OFF:	Timer output 3 off
TXD0, TXD1:	Transmit data
$\overline{\text{UWR}}$:	Upper write strobe
V_{DD}:	Power supply
V_{PP}:	Programming power supply
V_{SS}, V_{SS3}:	Ground
$\overline{\text{WAIT}}$:	Wait
X1, X2:	Crystal

1.6 Configuration of Function Block

1.6.1 Internal block diagram



Notes 1. μPD703114: 128 KB (mask ROM)
μPD70F3114: 128 KB (flash memory)

2. μPD70F3114 only

1.6.2 Internal units

(1) CPU

The CPU uses 5-stage pipeline control to execute address calculation, arithmetic and logical operation, data transfer, and most other instruction processing in one clock.

A multiplier (16 bits × 16 bits → 32 bits or 32 bits × 32 bits → 64 bits), barrel shifter (32-bit), and other dedicated hardware are on-chip to accelerate complex instruction processing.

(2) Bus control unit (BCU)

The BCU starts a required external bus cycle based on a physical address obtained from the CPU. If there is no bus cycle start request from the CPU when fetching an instruction from an external memory area, the BCU generates a prefetch address and prefetches the instruction code. The prefetched instruction code is fetched into the internal instruction queue of the CPU.

(3) Memory controller (MEMC)

The MEMC controls SRAM, ROM, and various I/O for external memory expansion.

(4) DMA controller (DMAC)

The DMAC transfers data between memory and I/O in place of the CPU.

The address mode is two-cycle transfer. The three bus modes are single transfer, single-step transfer, and block transfer.

(5) ROM

The μPD703114 includes mask ROM (128 KB), and the μPD70F3114 includes flash memory (128 KB).

On an instruction fetch, the ROM can be accessed by the CPU in one clock.

When single-chip mode or flash memory programming mode is set, ROM is mapped starting from address 00000000H.

ROM cannot be accessed if ROMless mode is set.

(6) RAM

RAM is mapped starting from address FFFFC000H.

It can be accessed by the CPU in one clock on an instruction fetch or data access.

(7) Interrupt controller (INTC)

The INTC services hardware interrupt requests from on-chip peripheral I/O and external sources (NMI, INTP0 to INTP4, INTP20 to INTP25, INTP30, INTP31, INTP100, INTP101). For these interrupt requests, eight levels of interrupt priority can be defined and multiprocessing controls against the interrupt sources can be performed.

(8) Clock generator (CG)

The CG provides a frequency that is 1, 2.5, 5, or 10 times (using the on-chip PLL) or 0.5 times (not using the on-chip PLL) the input clock (fx) as the internal system clock (fxx). As the input clock, connect an external resonator to pins X1 and X2 (only when using the on-chip PLL synthesizer) or input an external clock from the X1 pin.

(9) Timer/counter function

This unit incorporates a 2-channel 16-bit timer (TM0) for 3-phase sine wave PWM inverter control, a 1-channel 16-bit up/down counter (TM1) that can be used for 2-phase encoder input or as a general-purpose timer, a 2-channel 16-bit general-purpose timer unit (TM2), a 1-channel 16-bit timer/event counter (TM3), and a 1-channel 16-bit interval timer (TM4) on-chip, and can measure the pulse interval or frequency and can output a programmable pulse.

(10) Serial interface

A total of four channels of serial interfaces, including asynchronous serial interface (UART) and clocked serial interface (CSI), are provided. Of these channels, two are used for both UART and CSI, and their function must be selected. Of the other two channels, one is fixed to UART, and one is fixed to CSI.

The UART performs data transfer using pins TXDn and RXDn (n = 0, 1).

The CSI performs data transfer using pins SOn, SIn, and $\overline{\text{SCKn}}$ (n = 0, 1).

(11) A/D converter (ADC)

Two circuits of high-speed, high-resolution 10-bit A/D converters with a total of 14 pins (A/D converter 0: 6 pins, A/D converter 1: 8 pins) are available. The ADC converts using a successive approximation method.

(12) Ports

As shown in the table below, ports function as general-purpose ports and as control pins.

Port	I/O	Control Functions
Port 0	6-bit input	NMI input Timer/counter output stop signal input External interrupt input A/D converter external trigger input Timer 3 output stop signal input
Port 1	3-bit I/O	Timer/counter I/O External interrupt input
Port 2	8-bit I/O	Timer/counter I/O External interrupt input
Port 3	5-bit I/O	Serial interface I/O (UART0, UART1/CSI1)
Port 4	3-bit I/O	Serial interface I/O (CSI0)
Port DH	6-bit I/O	External address bus (A16 to A21)
Port DL	16-bit I/O	External address/data bus (AD0 to AD15)
Port CT	4-bit I/O	External bus interface control signal output
Port CM	2-bit I/O	Wait insertion signal input Internal system clock output

CHAPTER 2 PIN FUNCTIONS

The names and functions of the V850E/IA2 pins are shown below. These pins can be divided by function into port pins and non-port pins.

2.1 List of Pin Functions

(1) Port pins

(1/2)

Pin Name	I/O	Function	Alternate Function
P00	Input	Port 0 6-bit input-only port P00 is the input port that indicates the status of the NMI pin. The level of the NMI pin can be read by reading the P0.P00 bit. When a valid edge is input, the port functions as an NMI input.	NMI
P01			ESO0/INTP0
P02			ESO1/INTP1
P03			ADTRG0/INTP2
P04			ADTRG1/INTP3
P05			INTP4/TO3OFF
P10	I/O	Port 1 3-bit I/O port Input or output can be specified in 1-bit units	TIUD10/TO10
P11			TCUD10/INTP100
P12			TCLR10/INTP101
P20	I/O	Port 2 8-bit I/O port Input or output can be specified in 1-bit units	TI2/INTP20
P21			TO21/INTP21
P22			TO22/INTP22
P23			TO23/INTP23
P24			TO24/INTP24
P25			TCLR2/INTP25
P26			TI3/TCLR3/INTP30
P27			TO3/INTP31
P30	I/O	Port 3 5-bit I/O port Input or output can be specified in 1-bit units	RXD0
P31			TXD0
P32			RXD1/SI1
P33			TXD1/SO1
P34			$\overline{\text{ASCK1}}$/$\overline{\text{SCK1}}$
P40	I/O	Port 4 3-bit I/O port Input or output can be specified in 1-bit units	SI0
P41			SO0
P42			$\overline{\text{SCK0}}$
PCM0	I/O	Port CM 2-bit I/O port Input or output can be specified in 1-bit units	$\overline{\text{WAIT}}$
PCM1			CLKOUT

(2/2)

Pin Name	I/O	Function	Alternate Function
PCT0	I/O	Port CT 4-bit I/O port Input or output can be specified in 1-bit units	$\overline{\text{LWR}}$
PCT1			$\overline{\text{UWR}}$
PCT4			$\overline{\text{RD}}$
PCT6			ASTB
PDH0	I/O	Port DH 6-bit I/O port Input or output can be specified in 1-bit units	A16
PDH1			A17
PDH2			A18
PDH3			A19
PDH4			A20
PDH5			A21
PDL0	I/O	Port DL 16-bit I/O port Input or output can be specified in 1-bit units	AD0
PDL1			AD1
PDL2			AD2
PDL3			AD3
PDL4			AD4
PDL5			AD5
PDL6			AD6
PDL7			AD7
PDL8			AD8
PDL9			AD9
PDL10			AD10
PDL11			AD11
PDL12			AD12
PDL13			AD13
PDL14			AD14
PDL15			AD15

(2) Non-port pins

(1/3)

Pin Name	I/O	Function	Alternate Function
TO000	Output	Timer 00 pulse signal output	–
TO001			–
TO002			–
TO003			–
TO004			–
TO005			–
TO010	Output	Timer 01 pulse signal output	–
TO011			–
TO012			–
TO013			–
TO014			–
TO015			–
TO10	Output	Timer 10 pulse signal output	P10/TIUD10
TO21	Output	Timer 2 pulse signal output	P21/INTP21
TO22			P22/INTP22
TO23			P23/INTP23
TO24			P24/INTP24
TO3	Output	Timer 3 pulse signal output	P27/INTP31
ESO0	Input	Timer 00 or 01 output stop signal input	P01/INTP0
ESO1			P02/INTP1
TIUD10	Input	External count clock input to up/down counter (timer 10)	P10/TO10
TCUD10	Input	Count operation switching signal to up/down counter (timer 10)	P11/INTP100
TCLR10	Input	Clear signal input to up/down counter (timer 10)	P12/INTP101
TI2	Input	Timer 2 or 3 external count clock input	P20/INTP20
TI3			P26/INTP30/TCLR3
TCLR2	Input	Timer 2 or 3 clear signal input	P25/INTP25
TCLR3			P26/INTP30/TI3
INTP0	Input	External maskable interrupt request input	P01/ESO0
INTP1			P02/ESO1
INTP2			P03/ADTRG0
INTP3			P04/ADTRG1
INTP4			P05/TO3OFF
INTP100	Input	External maskable interrupt request input and timer 10 external capture trigger input	P11/TCUD10
INTP101			P12/TCLR10

(2/3)

Pin Name	I/O	Function	Alternate Function
INTP20	Input	External maskable interrupt request input and timer 2 external capture trigger input	P20/TI2
INTP21			P21/TO21
INTP22			P22/TO22
INTP23			P23/TO23
INTP24			P24/TO24
INTP25			P25/TCLR2
INTP30	Input	External maskable interrupt request input and timer 3 external capture trigger input	P26/TI3/TCLR3
INTP31			P27/TO3
TO3OFF	Input	Timer 3 output stop signal input	P05/INTP4
SO0	Output	Serial transmit data output (3-wire) of CSI0 and CSI1	P41
SO1			P33/TXD1
SI0	Input	Serial receive data input (3-wire) of CSI0 and CSI1	P40
SI1			P32/RXD1
$\overline{SCK0}$	I/O	Serial clock I/O (3-wire) of CSI0 and CSI1	P42
$\overline{SCK1}$			P34/$\overline{ASCK1}$
TXD0	Output	Serial transmit data output of UART0 and UART1	P31
TXD1			P33/SO1
RXD0	Input	Serial receive data input of UART0 and UART1	P30
RXD1			P32/SI1
$\overline{ASCK1}$	I/O	UART1 serial clock I/O	P34/$\overline{SCK1}$
ANI00 to ANI05	Input	Analog input to A/D converter	–
ANI10 to ANI17			–
ADTRG0	Input	External trigger input to A/D converter	P03/INTP2
ADTRG1			P04/INTP3
NMI	Input	Non-maskable interrupt request input	P00
MODE0	Input	Specifies V850E/IA2 operation mode	–
MODE1			V_{PP}[Note]
V_{PP}[Note]	–	Power application for flash memory write	MODE1
$\overline{WAIT}$	Input	Control signal input to insert wait in bus cycle	PCM0
$\overline{LWR}$	Output	External data lower byte write strobe signal output	PCT0
$\overline{UWR}$	Output	External data higher byte write strobe signal output	PCT1
$\overline{RD}$	Output	External data bus read strobe signal output	PCT4
ASTB	Output	External data bus address strobe signal output	PCT6
AD0 to AD15	I/O	16-bit address/data bus for external memory	PDL0 to PDL15
A16 to A21	Output	Higher 6-bit address bus for external memory	PDH0 to PDH5
$\overline{RESET}$	Input	System reset input	–
X1	Input	Crystal resonator connection pin for system clock oscillation. Input to X1 pin when providing clocks from outside.	–
X2	–		–

Note μPD70F3114 only

(3/3)

Pin Name	I/O	Function	Alternate Function
CLKOUT	Output	System clock output	PCM1
CKSEL	Input	Input specifying clock generator operation mode	–
AV_{DD0}, AV_{DD1}	–	Positive power supply for A/D converter	–
AV_{SS0}, AV_{SS1}	–	Ground potential for A/D converter	–
CV_{SS}	–	Ground potential for oscillator, PLL and regulator	–
V_{DD}	–	5 V system positive power supply for peripheral interface	–
V_{SS}	–	5 V system ground potential for peripheral interface	–
RV_{DD}	–	Positive power supply pin for regulator (5 V system power supply pin)	–
V_{SS3}	–	Internal 3.3 V system ground pin	–
REGOUT	Output	Regulator output pin	–
REGIN	Input	Regulator input pin (3.3 V system power supply pin)	–

2.2 Pin Status

The following table shows the status of each pin after a reset, in power-saving mode (software STOP mode, IDLE, HALT), and during a DMA transfer.

Pin \ Operating Status	Reset (Single-Chip Mode)	Reset (ROMless Mode)	IDLE Mode/ Software STOP Mode	HALT Mode/ During DMA Transfer
A16 to A21 (PDH0 to PDH5)	Hi-Z	Hi-Z	Hi-Z	Operating
AD0 to AD15 (PDL0 to PDL15)	Hi-Z	Hi-Z	Hi-Z	Operating
$\overline{\text{LWR}}$, $\overline{\text{UWR}}$ (PCT0, PCT1)	Hi-Z	Hi-Z	H	Operating
$\overline{\text{RD}}$ (PCT4)	Hi-Z	Hi-Z	H	Operating
ASTB (PCT6)	Hi-Z	Hi-Z	H	Operating
$\overline{\text{WAIT}}$ (PCM0)	Hi-Z	Hi-Z	–	Operating
CLKOUT (PCM1)	Hi-Z	Operating	L	Operating

Caution When controlling the external bus using an ASIC or the like in standby mode, provide a separate controller.

Remarks Hi-Z: High impedance
H: High-level output
L: Low-level output
–: No input sampling

2.3 Description of Pin Functions

(1) P00 to P05 (Port 0) … Input

P00 to P05 function as a 6-bit input-only port in which all pins are fixed to input.

Besides functioning as an input port, in control mode, P00 to P05 operate as NMI input, timer/counter output stop signal input, external interrupt request input, A/D converter (ADC) external trigger input, and timer 3 output stop signal input. Normally, if port pins also have alternate functions, the mode is selected using a port mode control register. However, there is no such register for P00 to P05. Therefore, the input port cannot be switched with the NMI input pin, timer/counter output stop signal input pin, external interrupt request input pin, A/D converter (ADC) external trigger input pin, and timer 3 output stop signal input pin. Read the status of each pin by reading the port.

(a) Port mode

P00 to P05 are input-only.

(b) Control mode

P00 to P05 also serve as the NMI, ESO0, ESO1, ADTRG0, ADTRG1, INTP0 to INTP4, and TO3OFF pins, but they cannot be switched.

(i) NMI (Non-maskable interrupt request) … Input

This is non-maskable interrupt request input.

(ii) ESO0, ESO1 (Emergency shut off) … Input

These pins input timer 00 and timer 01 output stop signals.

(iii) INTP0 to INTP4 (External interrupt input) … Input

These are external interrupt request input pins.

(iv) ADTRG0, ADTRG1 (A/D trigger input) … Input

These are A/D converter external trigger input pins.

(v) TO3OFF (Timer output 3 off) … Input

This is a timer output stop signal input pin.

(2) P10 to P12 (Port 1) … I/O

P10 to P12 function as a 3-bit I/O port in which input or output can be set in 1-bit units.

Besides functioning as an I/O port, in control mode, P10 to P12 operate as timer/counter I/O and external interrupt request input.

Port or control mode can be selected as the operation mode for each bit, specified by the port 1 mode control register (PMC1).

(a) Port mode

P10 to P12 can be set to input or output in 1-bit units using the port 1 mode register (PM1).

(b) Control mode

P10 to P12 can be set to port or control mode in 1-bit units using PMC1.

(i) TO10 (Timer output) … Output

This pin outputs the timer 10 pulse signal.

(ii) TIUD10 (Timer count pulse input) … Input

This is an external count clock input pin to the up/down counter (timer 10).

(iii) TCUD10 (Timer control pulse input) … Input

This pin inputs count operation switching signals to the up/down counter (timer 10).

(iv) TCLR10 (Timer clear) … Input

This is a clear signal input pin to the up/down counter (timer 10).

(v) INTP100, INTP101 (External interrupt input) … Input

These are external interrupt request input pins and timer 10 external capture trigger input pins.

(3) P20 to P27 (Port 2) … I/O

P20 to P27 function as an 8-bit I/O port in which input or output can be set in 1-bit units.

Besides functioning as an I/O port, in control mode, P20 to P27 operate as timer/counter I/O and external interrupt request input.

Port or control mode can be selected as the operation mode for each bit, specified by the port 2 mode control register (PMC2).

(a) Port mode

P20 to P27 can be set to input or output in 1-bit units using the port 2 mode register (PM2).

(b) Control mode

P20 to P27 can be set to port or control mode in 1-bit units using PMC2.

(i) TO21 to TO24 (Timer output) … Output

These pins output a timer 2 pulse signal.

(ii) TO3 (Timer output) … Output

This pin outputs a timer 3 pulse signal.

(iii) TI2, TI3 (Timer input) … Input

These are timer 2 and timer 3 external count clock input pins.

(iv) TCLR2, TCLR3 (Timer clear) … Input

These are timer 2 and timer 3 clear signal input pins.

(v) INTP20 to INTP25 (External interrupt input) … Input

These are external interrupt request input pins and timer 2 external capture trigger input pins.

(vi) INTP30, INPT31 (External interrupt input) … Input

These are external interrupt request input pins and timer 3 external capture trigger input pins.

(4) P30 to P34 (Port 3) … I/O

P30 to P34 function as a 5-bit I/O port in which input or output can be set in 1-bit units.

Besides functioning as an I/O port, in control mode, P30 to P34 operate as serial interface (UART0, UART1/CSI1) I/O.

Port or control mode can be selected as the operation mode for each bit, specified by the port 3 mode control register (PMC3). The selection of UART/SCI1 is specified by the port 3 function control register (PFC3).

(a) Port mode

P30 to P34 can be set to input or output in 1-bit units using the port 3 mode register (PM3).

(b) Control mode

P30 to P34 can be set to port or control mode in 1-bit units using PMC3.

(i) TXD0, TXD1 (Transmit data) … Output

These pins output serial transmit data of UART0 and UART1.

(ii) RXD0, RXD1 (Receive data) … Input

These pins input serial receive data of UART0 and UART1.

(iii) $\overline{\text{ASCK1}}$ (Asynchronous serial clock) … I/O

This is UART1 serial clock I/O pin.

(iv) SO1 (Serial output) … Output

This pin outputs serial transmit data of CSI1.

(v) SI1 (Serial input) … Input

This pin inputs serial receive data of CSI1.

(vi) $\overline{\text{SCK1}}$ (Serial clock) … I/O

This pin is CSI1 serial clock I/O pin.

(5) P40 to P42 (Port 4) … I/O

P40 to P42 function as a 3-bit I/O port in which input or output can be set in 1-bit units.

Besides functioning as an I/O port, in control mode, P40 to P42 operate as serial interface (CSI0) I/O.

Port or control mode can be selected as the operation mode for each bit, specified by the port 4 mode control register (PMC4).

(a) Port mode

P40 to P42 can be set to input or output in 1-bit units using the port 4 mode register (PM4).

(b) Control mode

P40 to P42 can be set to port or control mode in 1-bit units using PMC4.

(i) SO0 (Serial output) … Output

This pin outputs CSI0 serial transmit data.

(ii) SI0 (Serial input) … Input

This pin inputs CSI0 serial receive data.

(iii) $\overline{\text{SCK0}}$ (Serial clock) … I/O

This is CSI0 serial clock I/O pin.

(6) PCM0, PCM1 (Port CM) … I/O

PCM0 and PCM1 function as a 2-bit I/O port in which input or output can be set in 1-bit units.

Besides functioning as a port, in control mode, PCM0 and PCM1 operate as wait insertion signal input and internal system clock output.

Port or control mode can be selected as the operation mode for each bit, specified by the port CM mode control register (PMCCM).

(a) Port mode

PCM0 and PCM1 can be set to input or output in 1-bit units using the port CM mode register (PMCM).

(b) Control mode

PCM0 and PCM1 can be set to port or control mode in 1-bit units using PMCCM.

(i) $\overline{\text{WAIT}}$ (Wait) … Input

This control signal input pin, which inserts a data wait in a bus cycle, can be input asynchronously to the CLKOUT signal. Sampling is performed at the falling edge of the CLKOUT signal in the T2 or TW state of the bus cycle. If the setup or hold time is not secured within the sampling timing, wait insertion may not be performed.

(ii) CLKOUT (Clock output) … Output

This is an internal system clock output pin. In single-chip mode, output is not performed by the CLKOUT pin because it is in port mode. To perform CLKOUT output, set this pin to control mode using the port CM mode control register (PMCCM). This pin performs CLKOUT output, even during the reset period, in ROMless mode.

(7) PCT0, PCT1, PCT4, PCT6 (Port CT) … I/O

PCT0, PCT1, PCT4, and PCT6 function as a 4-bit I/O port in which input or output can be set in 1-bit units.

Besides functioning as a port, in control mode, these pins operate as control signal output for when memory is expanded externally.

Port or control mode can be selected as the operation mode for each bit, specified by the port CT mode control register (PMCCT).

(a) Port mode

PCT0, PCT1, PCT4, and PCT6 can be set to input or output in 1-bit units using the port CT mode register (PMCT).

(b) Control mode

PCT0, PCT1, PCT4, and PCT6 can be set to port or control mode in 1-bit units using PMCCT.

(i) $\overline{\text{LWR}}$ (Lower byte write strobe) … Output

This is a strobe signal that shows that the bus cycle being executed is a write cycle for SRAM, external ROM, or an external peripheral I/O area.

In the data bus, the lower byte is valid. If the bus cycle is a lower memory write, it becomes active at the falling edge of the CLKOUT signal in the T1 state and becomes inactive at the falling edge of the CLKOUT signal in the T2 state.

(ii) $\overline{\text{UWR}}$ (Higher byte write strobe) … Output

This is a strobe signal that shows that the bus cycle being executed is a write cycle for SRAM, external ROM, or an external peripheral I/O area.

In the data bus, the higher byte is valid. If the bus cycle is a higher memory write, it becomes active at the falling edge of the CLKOUT signal in the T1 state and becomes inactive at the falling edge of the CLKOUT signal in the T2 state.

(iii) $\overline{\text{RD}}$ (Read strobe) … Output

This is a strobe signal that shows that the bus cycle being executed is a read cycle for SRAM, external ROM, or external peripheral I/O. It is inactive in the idle state (TI).

(iv) ASTB (Address strobe) … Output

This is the external address bus latch strobe signal output pin.

Output becomes low level in synchronization with the falling edge of the clock in the T1 state of the bus cycle, and high level in synchronization with the falling edge of the clock in the T3 state.

(8) PDH0 to PDH5 (Port DH) … I/O

PDH0 to PDH5 function as a 6-bit I/O port in which input or output can be set in 1-bit units.

Besides functioning as a port, in control mode (external expansion mode), these pins operate as the address bus (A16 to A21) for when memory is expanded externally.

Port or control mode can be selected as the operation mode for each bit, specified by the port DH mode control register (PMCDH).

(a) Port mode

PDH0 to PDH5 can be set to input or output in 1-bit units using the port DH mode register (PMDH).

(b) Control mode

PDH0 to PDH5 can be specified as A16 to A21 using PMCDH.

(i) A16 to A21 (Address) … Output

These pins output the higher 6-bit address of the 22-bit address in the address bus on an external access.

(9) PDL0 to PDL15 (Port DL) … I/O

PDL0 to PDL15 function as a 16-bit I/O port in which input or output can be set in 1-bit units.

Besides functioning as a port, in control mode (external expansion mode), these pins operate as the address/data bus (AD0 to AD15) for when memory is expanded externally.

Port or control mode can be selected as the operation mode for each bit, specified by the port DL mode control register (PMCDL).

(a) Port mode

PDL0 to PDL15 can be set to input or output in 1-bit units using the port DL mode register (PMDL).

(b) Control mode

PDL0 to PDL15 can be specified as AD0 to AD15 using PMCDL.

(i) AD0 to AD15 (Address/data bus) … I/O

This is a multiplexed bus for addresses or data on an external access. When used for addresses (T1 state) these pins output A0 to A15 of the 22-bit address, and when used for data (T2, TW, T3) they are 16-bit data I/O bus pins.

(10) TO000 to TO005 (Timer output) … Output

These pins output the pulse signal of timer 00.

(11) TO010 to TO015 (Timer output) … Output

These pins output the pulse signal of timer 01.

(12) ANI00 to ANI05, ANI10 to ANI17 (Analog input) … Input

These pins input analog signals to the A/D converter.

(13) CKSEL (Clock generator operating mode select) … Input

This is the input pin that specifies the operation mode of the clock generator. Fix this pin so that the input level does not change during operation.

(14) MODE0, MODE1 (Mode) … Input

These are the input pins that specify the operation mode. Operation modes are broadly divided into normal operation modes and flash memory programming mode. The normal operation modes are single-chip mode and ROMless mode (see **3.3 Operation Modes** for details). The operation mode is determined by sampling the status of each of the MODE0 and MODE1 pins on a reset.

Fix these pins so that the input level does not change during operation.

(a) μPD703114

MODE1	MODE0	Operation Mode	
L	L	Normal operation mode	ROMless mode
L	H		Single-chip mode
Other than above		Setting prohibited	

(b) μPD70F3114

MODE1/V_{PP}	MODE0	Operation Mode	
L	L	Normal operation mode	ROMless mode
L	H		Single-chip mode
7.8 V	H	Flash memory programming mode	
Other than above		Setting prohibited	

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

(15) $\overline{\text{RESET}}$ (Reset) … Input

$\overline{\text{RESET}}$ input is asynchronous input. When a signal having a certain low level width is input in asynchronous with the operation clock, a system reset that takes precedence over all operations occurs.

Besides a normal initialize or start, this signal is also used to release a standby mode (HALT, IDLE, software STOP).

(16) X1, X2 (Crystal)

These pins connect a resonator for system clock generation.

They can also input external clocks. In this case, connect the external clock to the X1 pin and leave the X2 pin open.

(17) CV_{SS} (Ground for clock generator)

This is the ground pin for the resonator, PLL and regulator.

(18) V_{DD} (Power supply)

This is the 5 V system positive power supply pin for the peripheral interface.

(19) V_{SS} (Ground)

This is the 5 V system ground pin for the peripheral interface.

(20) RV_{DD} (Regulator power supply)

This is the positive power supply pin for the regulator.
Supply 5 V system power to this pin.

(21) V_{SS3} (Ground)

This is the internal 3.3 V system ground pin.

(22) REGOUT (Regulator output) … Output

This is the regulator output pin.

(23) REGIN (Regulator input) … Input

This is the regulator input pin. Supply 3.3 V system power to this pin.

(24) AV_{DD0}, AV_{DD1} (Analog power supply)

These are the analog positive power supply pins for the A/D converter.

(25) AV_{SS0}, AV_{SS1} (Analog ground)

These are the ground pins for the A/D converter.

2.4 Types of Pin I/O Circuits and Connection of Unused Pins

Connection of a 1 to 10 kΩ resistor is recommended when connecting to V_{DD}, V_{SS}, or CV_{SS} via a resistor.

(1/2)

Pin	I/O Circuit Type	Recommended Connection
P00/NMI	2	Connect directly to V_{SS}.
P01/ESO0/INTP0 P02/ESO1/INTP1		
P03/ADTRG0/INTP2 P04/ADTRG1/INTP3		
P05/INTP4/TO3OFF		
P10/TIUD10/TO10	5-AC	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P11/TCUD10/INTP100		
P12/TCLR10/INTP101		
P20/TI2/INTP20		
P21/TO21/INTP21 to P24/TO24/INTP24		
P25/TCLR2/INTP25		
P26/TI3/TCLR3/INTP30		
P27/TO3/INTP31		
P30/RXD0		
P31/TXD0	5	
P32/RXD1/SI1	5-AC	
P33/TXD1/SO1	5	
P34/$\overline{\text{ASCK1}}$/$\overline{\text{SCK1}}$	5-AC	
P40/SI0		
P41/SO0	5	
P42/$\overline{\text{SCK0}}$	5-AC	
PCM0/$\overline{\text{WAIT}}$	5	
PCM1/CLKOUT		
PCT0/$\overline{\text{LWR}}$		
PCT1/$\overline{\text{UWR}}$		
PCT4/$\overline{\text{RD}}$		
PCT6/ASTB		
PDH0/A16 to PDH5/A21		
PDL0/AD0 to PDL15/AD15		
ANI00 to ANI05	7	Connect to AV_{SS0}.
ANI10 to ANI17		Connect to AV_{SS1}.
TO000 to TO005, TO010 to TO015	4	Leave open.

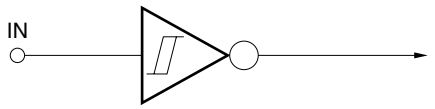
(2/2)

Pin	I/O Circuit Type	Recommended Connection
MODE0	2	–
V_{PP}[Note]/MODE1		
$\overline{\text{RESET}}$		
CKSEL		
X2	–	Leave open.
AV_{SS0}, AV_{SS1}	–	Connect to V_{SS}.
AV_{DD0}, AV_{DD1}	–	Connect to V_{DD}.
REGOUT	–	Leave open.

Note μPD70F3114 only

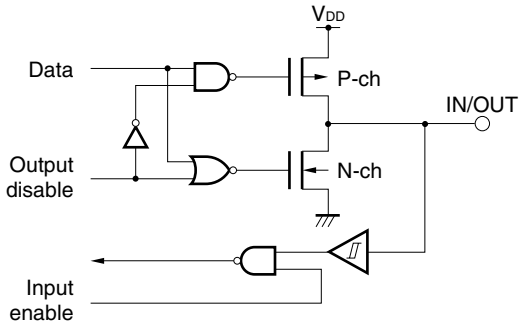
2.5 Pin I/O Circuits

Type 2

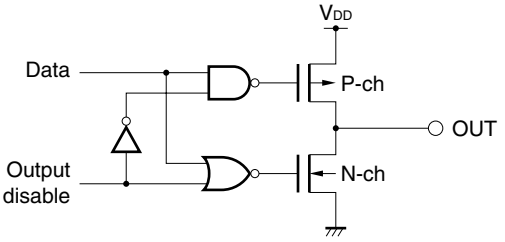


Schmitt-triggered input with hysteresis characteristics

Type 5-AC

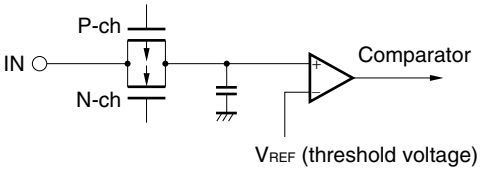


Type 4

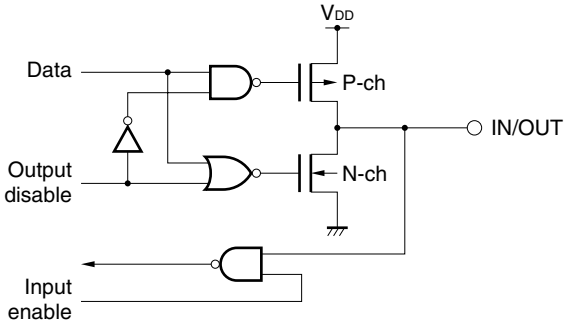


Push-pull output with possible high-impedance output (P-ch, N-ch both off)

Type 7



Type 5



CHAPTER 3 CPU FUNCTION

The CPU of the V850E/IA2 is based on RISC architecture and executes almost all instructions in one clock cycle, using 5-stage pipeline control.

3.1 Features

- Minimum instruction execution time: 25 ns (@ internal 40 MHz operation)
- Memory space Program space: 64 MB linear
  Data space: 4 GB linear
- Thirty-two 32-bit general-purpose registers
- Internal 32-bit architecture
- Five-stage pipeline control
- Multiplication/division instructions
- Saturated operation instructions
- One-clock 32-bit shift instruction
- Load/store instructions in long/short format
- Four types of bit manipulation instructions
  - SET1
  - CLR1
  - NOT1
  - TST1

3.2 CPU Register Set

The registers of the V850E/IA2 can be classified into two categories: a general-purpose program register set and a dedicated system register set. The width of all the registers is 32 bits.

For details, refer to **V850E1 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	(Status saving register during interrupt)	
EIPSW	(Status saving register during interrupt)	
FEPC	(Status saving register during NMI)	
FEPSW	(Status saving register during NMI)	
ECR	(Interrupt source register)	
PSW	(Program status word)	
CTPC	(Status saving register during CALLT execution)	
CTPSW	(Status saving register during CALLT execution)	
DBPC	(Status saving register during exception/debug trap)	
DBPSW	(Status saving register during exception/debug trap)	
CTBP	(CALLT base pointer)	

3.2.1 Program register set

The program register set includes general-purpose registers and a program counter.

(1) General-purpose registers

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

However, r0 and r30 are implicitly used by instructions, and care must be exercised when using these registers. r0 is a register that always holds 0, and is used for operations using 0 and offset 0 addressing. r30 is used, by means of the SLD and SST instructions, as a base pointer for when memory is accessed. Also, r1, r3 to r5, and r31 are implicitly used by the assembler and C compiler. Therefore, before using these registers, their contents must be saved so that they are not lost. The contents must be restored to these registers after they have been used. r2 is sometimes used by a real-time OS. r2 can be used as a register for variables when it is not being used by the real-time OS.

Table 3-1. Program Registers

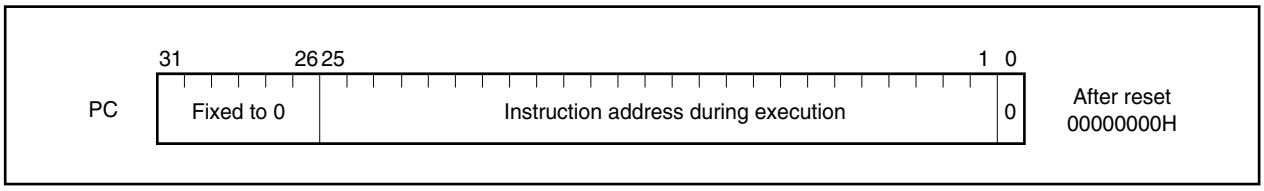
Name	Usage	Operation
r0	Zero register	Always holds 0
r1	Assembler-reserved register	Working register for generating address
r2	Address/data variable register (when not being used by the real-time OS)	
r3	Stack pointer	Used to generate stack frame when function is called
r4	Global pointer	Used to access global variable in data area
r5	Text pointer	Register to indicate the start of the text area (where program code is located)
r6 to r29	Address/data variable registers	
r30	Element pointer	Base pointer for generating address when memory is accessed
r31	Link pointer	Used by compiler when calling function
PC	Program counter	Holds instruction address during program execution

Remark For detailed descriptions of r1, r3 to r5, and r31, which are used by the assembler and C compiler, refer to **CA850 (C Compiler Package) Assembly Language User's Manual (U10543E)**.

(2) Program counter (PC)

This register holds the instruction address during program execution. The lower 26 bits of this register are valid, and bits 31 to 26 are fixed to 0. If a carry occurs from bit 25 to 26, it is ignored.

Bit 0 is fixed to 0, and branching to an odd address cannot be performed.



3.2.2 System register set

System registers control the status of the CPU and hold interrupt information.

To read/write these system registers, specify a system register number indicated below using the system register load/store instruction (LDSR or STSR instruction).

Table 3-2. System Register Numbers

No.	System Register Name	Operand Specification	
		LDSR Instruction	STSR Instruction
0	Status saving register during interrupt (EIPC)[Note 1]	○	○
1	Status saving register during interrupt (EIPSW)[Note 1]	○	○
2	Status saving register during NMI (FEPC)	○	○
3	Status saving register during NMI (FEPSW)	○	○
4	Interrupt source register (ECR)	×	○
5	Program status word (PSW)	○	○
6 to 15	Reserved number for future function expansion (operations that access these register numbers cannot be guaranteed).	×	×
16	Status saving register during CALLT execution (CTPC)	○	○
17	Status saving register during CALLT execution (CTPSW)	○	○
★ 18	Status saving register during exception/debug trap (DBPC)	○[Note 2]	○[Note 2]
★ 19	Status saving register during exception/debug trap (DBPSW)	○[Note 2]	○[Note 2]
20	CALLT base pointer (CTBP)	○	○
21 to 31	Reserved number for future function expansion (operations that access these register numbers cannot be guaranteed).	×	×

Notes 1. Because this register has only one set, to allow multiple interrupts, it is necessary to save this register by program.

2. Access is only possible during the period from when the DBTRAP instruction is executed to when the DBRET instruction is executed.

Caution Even if bit 0 of EIPC, FEPC, or CTPC is set to 1 with the LDSR instruction, bit 0 will be ignored when the program is returned by the RETI instruction after interrupt servicing (because bit 0 of the PC is fixed to 0). When setting the value of EIPC, FEPC, or CTPC, use an even value (bit 0 = 0).

Remark ○: Access allowed
×: Access prohibited

★ **(1) Interrupt status saving registers (EIPC, EIPSW)**

There are two interrupt status saving registers, EIPC and EIPSW.

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

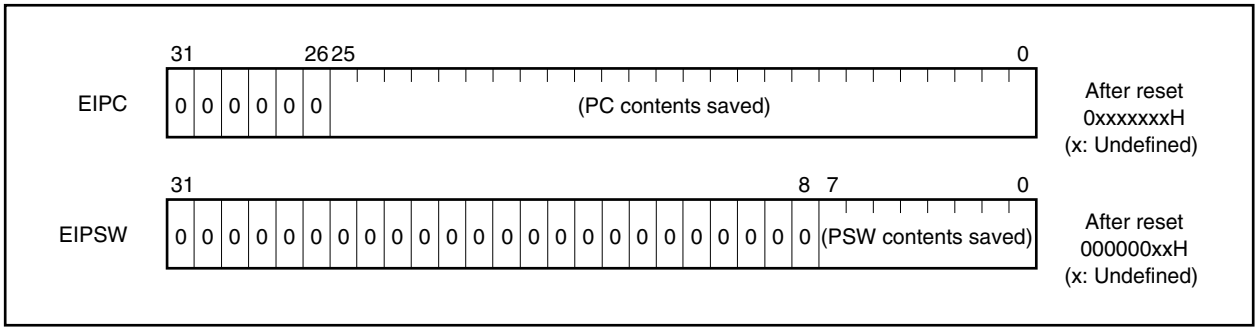
The address of the next instruction following the instruction executed when a software exception or maskable interrupt occurs is saved to EIPC, except for some instructions (refer to **7.8 Periods in Which CPU Does Not Acknowledge Interrupts**).

The current PSW contents are saved to EIPSW.

Since there is only one set of interrupt status saving registers, the contents of these registers must be saved by the program when multiple interrupt servicing is enabled.

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

When the RETI instruction is executed, the values in EIPC and EIPSW are restored to the PC and PSW, respectively.



★ **(2) NMI status saving registers (FEPC, FEPSW)**

There are two NMI status saving registers, FEPC and FEPSW.

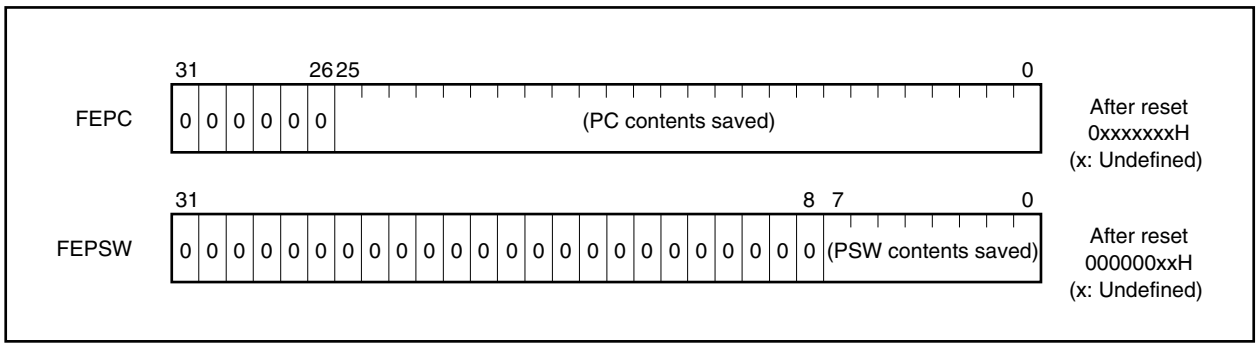
Upon occurrence of a non-maskable interrupt (NMI), the contents of the program counter (PC) are saved to FEPC and the contents of the program status word (PSW) are saved to FEPSW.

The address of the next instruction following the instruction executed when a non-maskable interrupt occurs is saved to FEPC, except for some instructions.

The current PSW contents are saved to FEPSW.

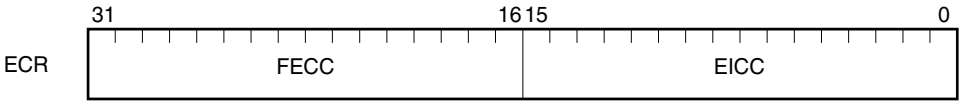
Bits 31 to 26 of FEPC and bits 31 to 8 of FEPSW are reserved (fixed to 0) for future function expansion.

When the RETI instruction has been executed, the values of FEPC and FEPSW are restored to the PC and PSW, respectively.



(3) Interrupt source register (ECR)

Upon occurrence of an interrupt or an exception, the interrupt source register (ECR) holds the source of an interrupt or an exception. The value held by ECR is the exception code coded for each interrupt source. This register is a read-only register, and thus data cannot be written to it using the LDSR instruction.



Bit position	Bit name	Description
31 to 16	FECC	Non-maskable interrupt (NMI) exception code
15 to 0	EICC	Exception, maskable interrupt exception code

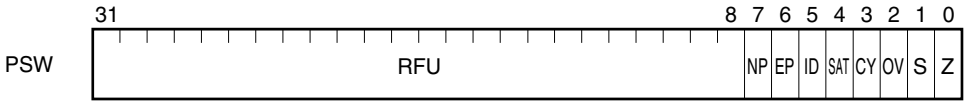
(4) Program status word (PSW)

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

When the contents of this register are changed using the LDSR instruction, the new contents become valid immediately following completion of LDSR instruction execution. Interrupt request acknowledgment is held pending while a write to the PSW is being executed by the LDSR instruction.

Bits 31 to 8 are reserved (fixed to 0) for future function expansion.

(1/2)



Bit position	Flag name	Description
31 to 8	RFU	Reserved field. Fixed to 0.
7	NP	Indicates that non-maskable interrupt (NMI) servicing is in progress. This flag is set to 1 when an NMI request is acknowledged, and disables multiple interrupts. 0: NMI servicing not in progress 1: NMI servicing in progress
6	EP	Indicates that exception processing is in progress. This flag is set to 1 when an exception occurs. Moreover, interrupt requests can be acknowledged even when this bit is set. 0: Exception processing not in progress 1: Exception processing in progress
5	ID	Indicates whether maskable interrupt request acknowledgment is enabled. 0: Interrupt enabled (EI) 1: Interrupt disabled (DI)
4	SAT[Note]	Indicates that the result of executing a saturated operation instruction has overflowed and that the calculation result is saturated. Since this is a cumulative flag, it is set to 1 when the result of a saturated operation instruction becomes saturated, and it is not cleared to 0 even if the operation results of successive instructions do not become saturated. This flag is neither set nor cleared when arithmetic operation instructions are executed. 0: Not saturated 1: Saturated
3	CY	Indicates whether carry or borrow occurred as the result of an operation. 0: No carry or borrow occurred 1: Carry or borrow occurred
2	OV[Note]	Indicates whether overflow occurred during an operation. 0: No overflow occurred 1: Overflow occurred.
1	S[Note]	Indicates whether the result of an operation is negative. 0: Operation result is positive or 0. 1: Operation result is negative.
0	Z	Indicates whether operation result is 0. 0: Operation result is not 0. 1: Operation result is 0.

Remark **Note** is explained on the following page.

(2/2)

Note During saturated operation, the saturated operation results are determined by the contents of the OV flag and S flag. The SAT flag is set (to 1) only when the OV flag is set (to 1) during saturated operation.

Operation result status	Flag status			Saturated operation result
	SAT	OV	S	
Maximum positive value exceeded	1	1	0	7FFFFFFFH
Maximum negative value exceeded	1	1	1	80000000H
Positive (maximum value not exceeded)	Holds value before operation	0	0	Actual operation result
Negative (maximum value not exceeded)			1	

★ **(5) CALLT execution status saving registers (CTPC, CTPSW)**

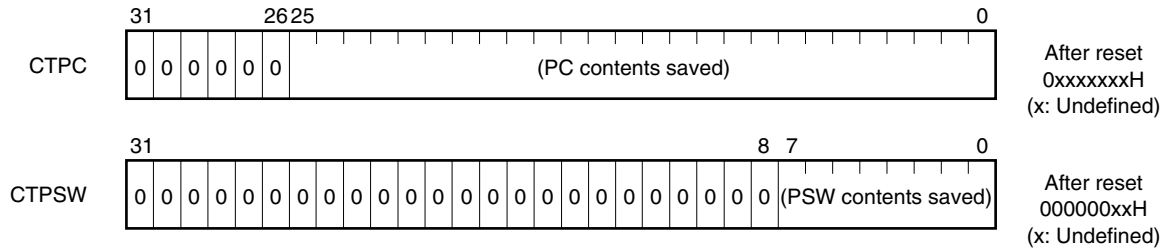
There are two CALLT execution status saving registers, CTPC and CTPSW.

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

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

The current PSW contents are saved to CTPSW.

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



★ **(6) Exception/debug trap status saving registers (DBPC, DBPSW)**

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

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

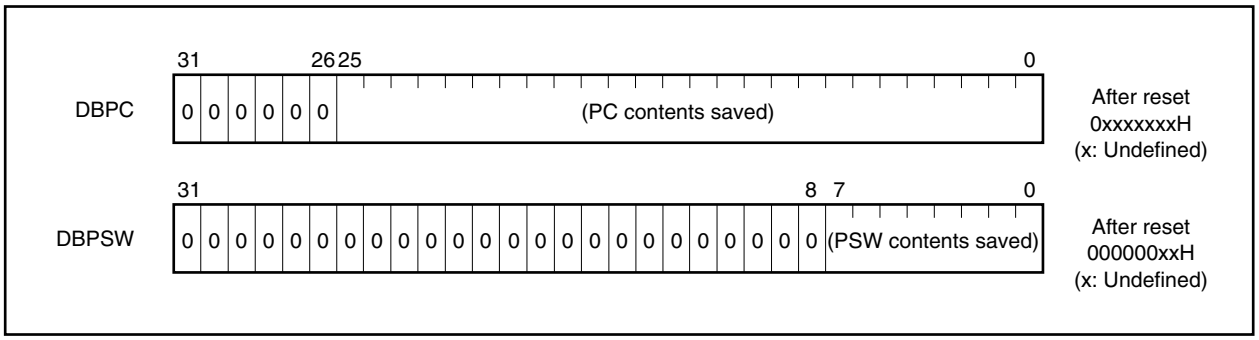
The contents saved to DBPC consist of the address of the next instruction after the instruction executed when an exception trap or debug trap occurs.

The current PSW contents are saved to DBPSW.

These registers can be read or written only in the period between DBTRAP instruction execution and DBRET instruction execution.

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

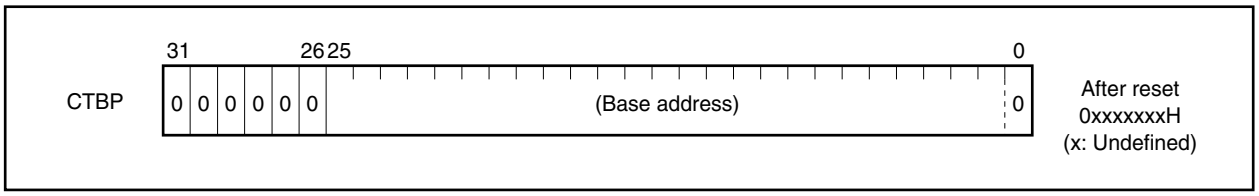
When the DBRET instruction has been executed, the values of DBPC and DBPSW are restored to the PC and PSW, respectively.



★ **(7) CALLT base pointer (CTBP)**

The CALLT base pointer (CTBP) is used to specify table addresses and generate target addresses (bit 0 is fixed to 0).

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



3.3 Operation Modes

3.3.1 Operation modes

The V850E/IA2 has the following operation modes. Mode specification is carried out by the MODE0 and MODE1 pins.

(1) Normal operation mode

(a) Single-chip mode

Access to the internal ROM is enabled.

In single-chip mode, after the system reset is cleared, each pin related to the bus interface enters the port mode, program execution branches to the reset entry address of the internal ROM, and instruction processing starts. By setting the PMCDH, PMCDL, PMCCT, and PMCCM registers to control mode by instruction, an external device can be connected to the external memory area.

(b) ROMless mode

After the system reset is cleared, each pin related to the bus interface enters the control mode, program execution branches to the external device's (memory) reset entry address, and instruction processing starts. Fetching of instructions and data access for internal ROM becomes impossible.

In ROMless mode, the data bus is a 16-bit data bus.

(2) Flash memory programming mode (μPD70F3114 only)

If this mode is specified, it becomes possible for the flash programmer to run a program to the internal flash memory.

The initial values of the registers differ depending on the mode.

Operation Mode		PMCDH	PMCDL	PMCCT	PMCCM	BSC
Normal operation mode	ROMless mode	FFH	FFFFH	53H	03H	5555H
	Single-chip mode	00H	0000H	00H	00H	5555H

3.3.2 Operation mode specification

The operation mode is specified according to the status of the MODE0 and MODE1 pins. In an application system, fix the specification of these pins and do not change them during operation. Operation is not guaranteed if these pins are changed during operation.

(a) μPD703114

MODE1	MODE0	Operation Mode		Remark
L	L	Normal operation mode	ROMless mode	16-bit data bus
L	H		Single-chip mode	Internal ROM area is allocated from address 000000H.
Other than above		Setting prohibited		

(b) μPD70F3114

MODE1/V_{PP}	MODE0	Operation Mode		Remark
L	L	Normal operation mode	ROMless mode	16-bit data bus
L	H		Single-chip mode	Internal ROM area is allocated from address 000000H.
7.8 V	H	Flash memory programming mode		–
Other than above		Setting prohibited		

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

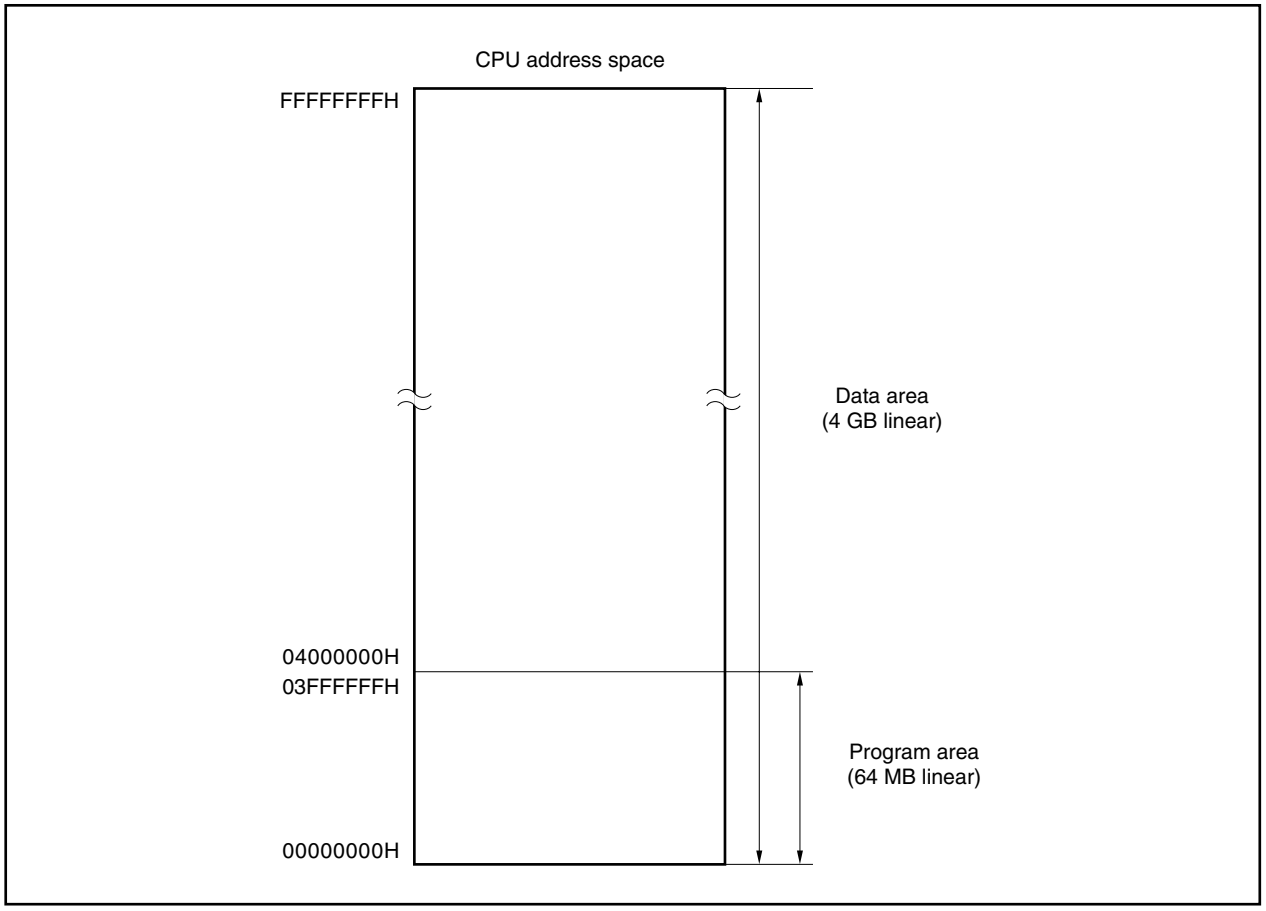
3.4 Address Space

3.4.1 CPU address space

The V850E1 CPU of the V850E/IA2 is of 32-bit architecture and supports up to 4 GB of linear address space (data space) during operand addressing (data access). Also, in instruction address addressing, a maximum of 64 MB of linear address space (program space) is supported.

Figure 3-1 shows the CPU address space.

Figure 3-1. CPU Address Space

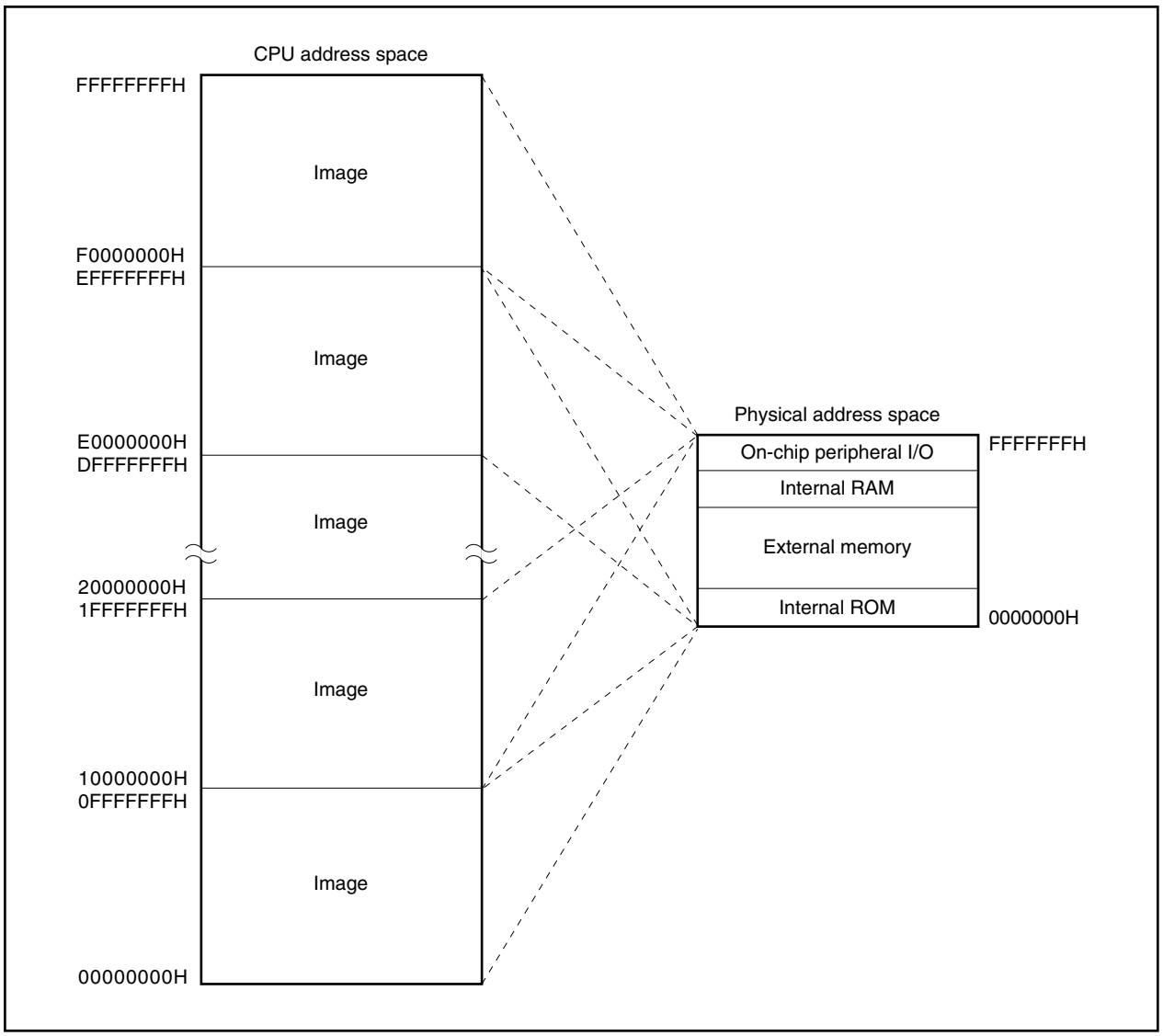


3.4.2 Image

16 images, each containing a 256 MB physical address space, are seen in the 4 GB CPU address space. In actuality, the same 256 MB physical address space is accessed regardless of the values of bits 31 to 28 of the CPU address. Figure 3-2 shows the image of the virtual addressing space.

Physical address x0000000H can be seen as CPU address 00000000H, and in addition, can be seen as address 10000000H, address 20000000H, … , address E0000000H, or address F0000000H.

Figure 3-2. Image on Address Space



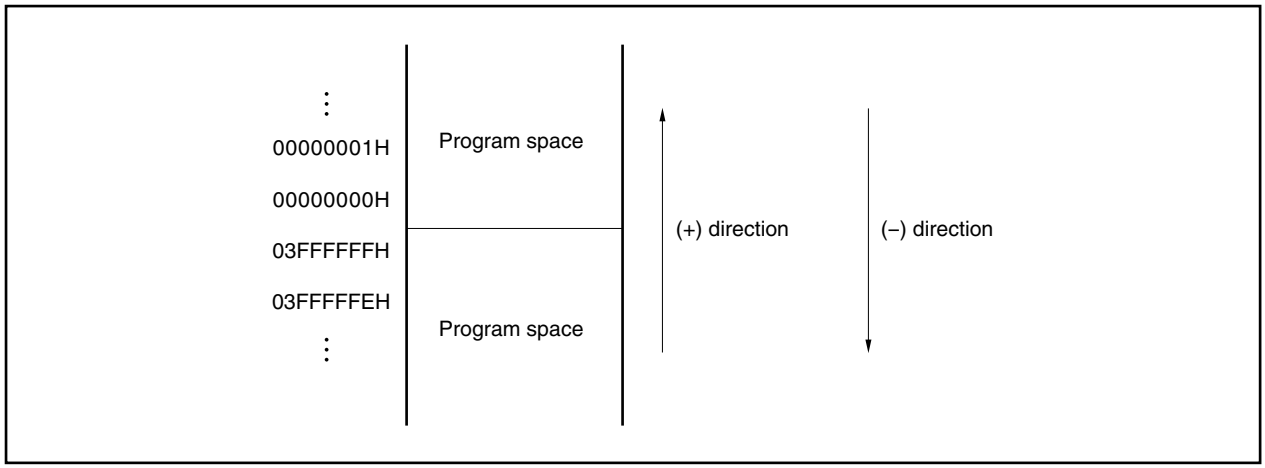
3.4.3 Wrap-around of CPU address space

(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. Even if a carry or borrow occurs from bit 25 to 26 as a result of branch address calculation, the higher 6 bits ignore the carry or borrow.

Therefore, the upper-limit address of the program space, address 03FFFFFFH, and the lower-limit address 00000000H become contiguous addresses. Wrap-around refers to a situation like this whereby the lower-limit address and upper-limit address become contiguous.

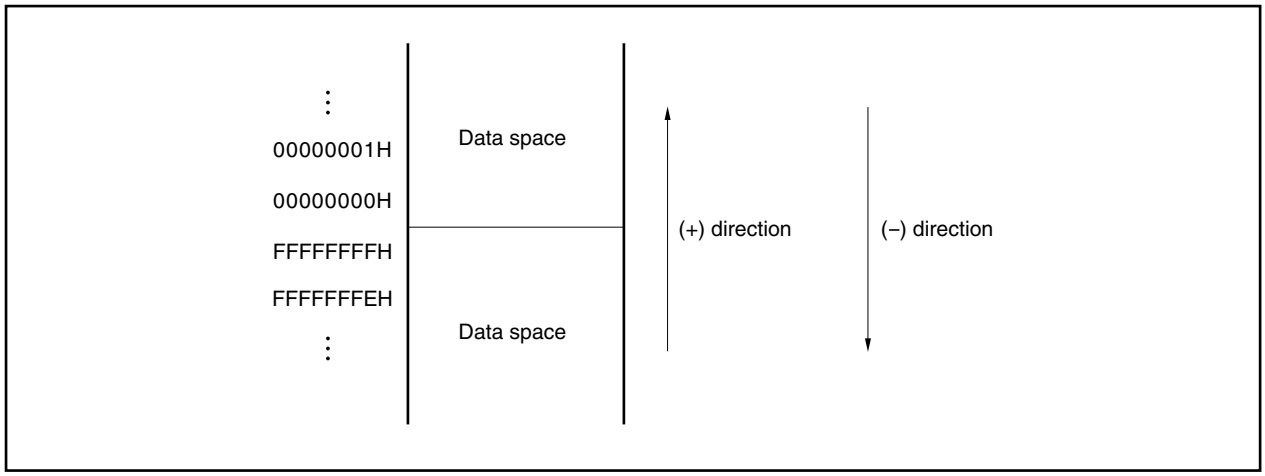
Caution The 4 KB area of 03FFF000H to 03FFFFFFH can be seen as an image of 0FFFF000H to 0FFFFFFFH. No instruction can be fetched from this area because this area is defined as on-chip peripheral I/O area. Therefore, do not execute any branch address calculation in which the result will reside in any part of this area.



(2) Data space

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

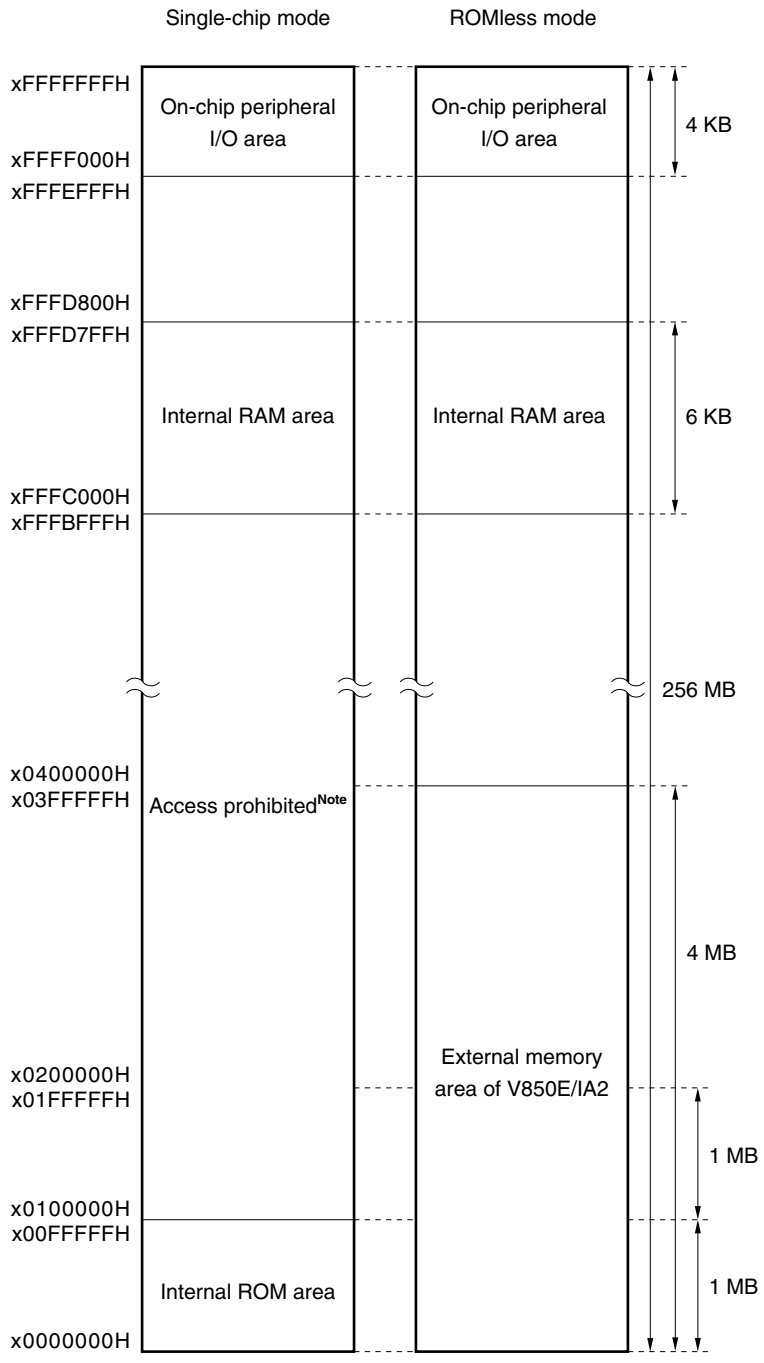
Therefore, the upper-limit address of the program space, address FFFFFFFFH, and the lower-limit address 00000000H are contiguous addresses, and the data space is wrapped around at the boundary of these addresses.



3.4.4 Memory map

The V850E/IA2 reserves areas as shown in Figure 3-3. Each mode is specified by the MODE0 and MODE1 pins.

Figure 3-3. Memory Map



Note By setting the PMCDH, PMCDL, PMCCT, and PMCCM registers to control mode, this area can be used as external memory area.

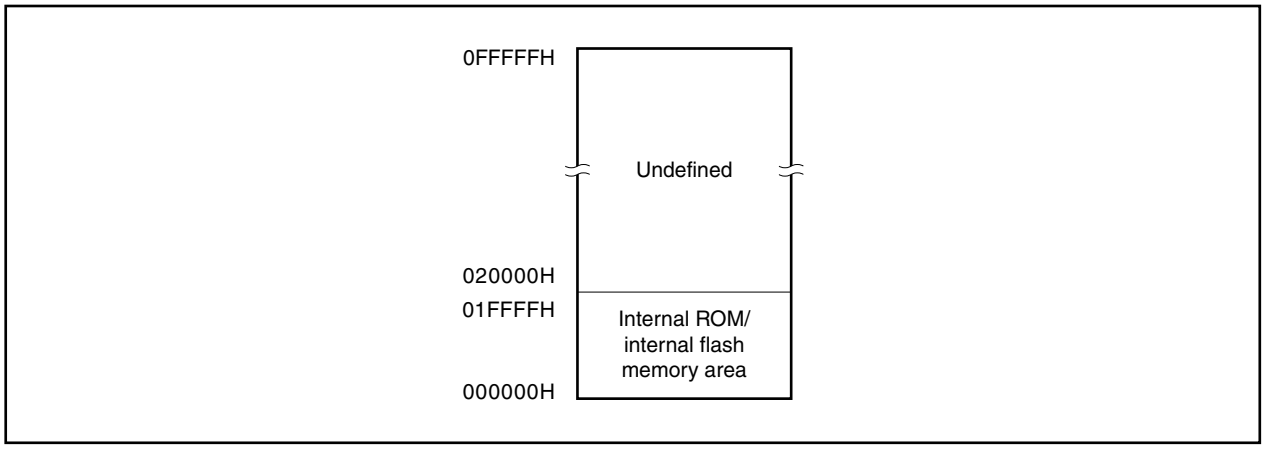
3.4.5 Area

(1) Internal ROM/internal flash memory area

(a) Memory map

1 MB of internal ROM/internal flash memory area, addresses 00000H to FFFFFH, is reserved.
Actually, internal ROM/internal flash memory of 128 KB is mapped to addresses 000000H to 01FFFFH.
Addresses 020000H to 0FFFFFH are undefined.

Figure 3-4. Internal ROM/Internal Flash Memory Area



(b) Interrupt/exception table

The V850E/IA2 increases the interrupt response speed by assigning handler addresses corresponding to interrupts/exceptions.

The collection of these handler addresses is called an interrupt/exception table, which is located in the internal ROM area. When an interrupt/exception request is acknowledged, execution jumps to the handler address, and the program written at that memory location is executed. Table 3-3 shows the sources of interrupts/exceptions, and the corresponding addresses.

Remark When in ROMless mode, in order to resume correct operation after reset, provide a handler address to the reset routine at address 0 of the external memory.

Table 3-3. Interrupt/Exception Table

Start Address of Interrupt/Exception Table	Interrupt/Exception Source	Start Address of Interrupt/Exception Table	Interrupt/Exception Source
00000000H	RESET	00000230H	INTP24/INTCC24
00000010H	NMI0	00000240H	INTP25/INTCC25
00000040H	TRAP0n (n = 0 to F)	00000250H	INTTM3
00000050H	TRAP1n (n = 0 to F)	00000260H	INTP30/INTCC30
00000060H	ILGOP/DBG0	00000270H	INTP31/INTCC31
00000080H	INTP0	00000280H	INTCM4
00000090H	INTP1	00000290H	INTDMA0
000000A0H	INTP2	000002A0H	INTDMA1
000000B0H	INTP3	000002B0H	INTDMA2
000000C0H	INTP4	000002C0H	INTDMA3
000000F0H	INTDET0	00000310H	INTCSI0
00000100H	INTDET1	00000320H	INTCSI1
00000110H	INTTM00	00000330H	INTSR0
00000120H	INTCM003	00000340H	INTST0
00000130H	INTTM01	00000350H	INTSER0
00000140H	INTCM013	00000360H	INTSR1
00000150H	INTP100/INTCC100	00000370H	INTST1
00000160H	INTP101/INTCC101	000003A0H	INTAD0
00000170H	INTCM100	000003B0H	INTAD1
00000180H	INTCM101	000003F0H	INTCM010
000001D0H	INTTM20	00000400H	INTCM011
000001E0H	INTTM21	00000410H	INTCM012
000001F0H	INTP20/INTCC20	00000420H	INTCM014
00000200H	INTP21/INTCC21	00000430H	INTCM015
00000210H	INTP22/INTCC22	00000440H	INTCM004
00000220H	INTP23/INTCC23	00000450H	INTCM005

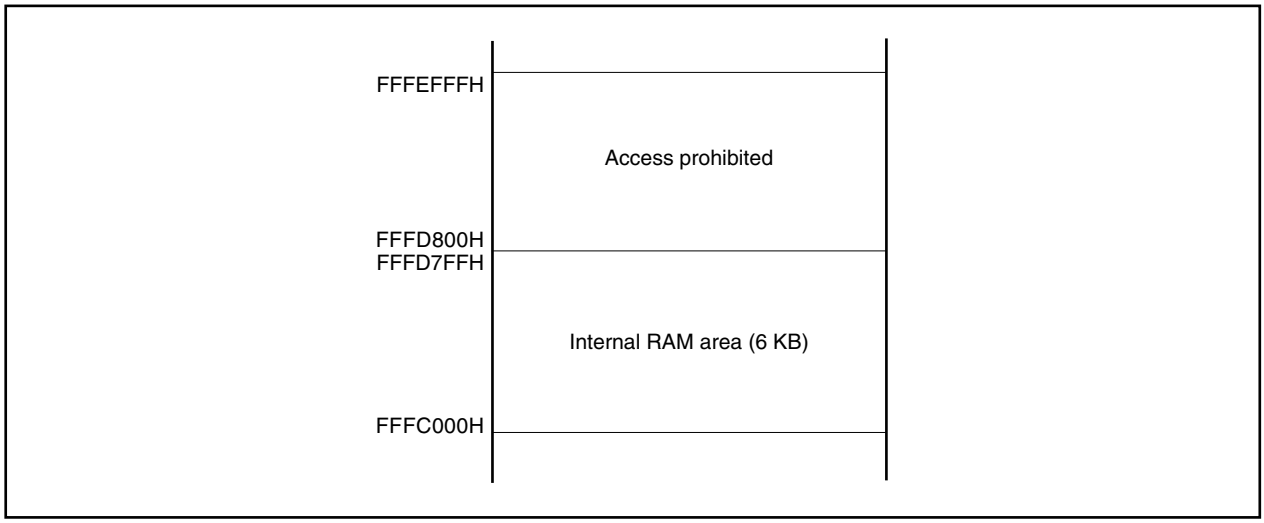
(2) Internal RAM area

12 KB of memory, addresses FFFC000H to FFFEFFFH, are reserved for the internal RAM area.

The 12 KB area of 3FFC000H to 3FFEFFFH can be seen as an image of FFFC000H to FFFEFFFH.

In the V850E/IA2, 6 KB of memory, addresses FFFC000H to FFFD7FFH, are provided as physical internal RAM.

Access to the area of addresses FFFD800H to FFFEFFFH is prohibited.

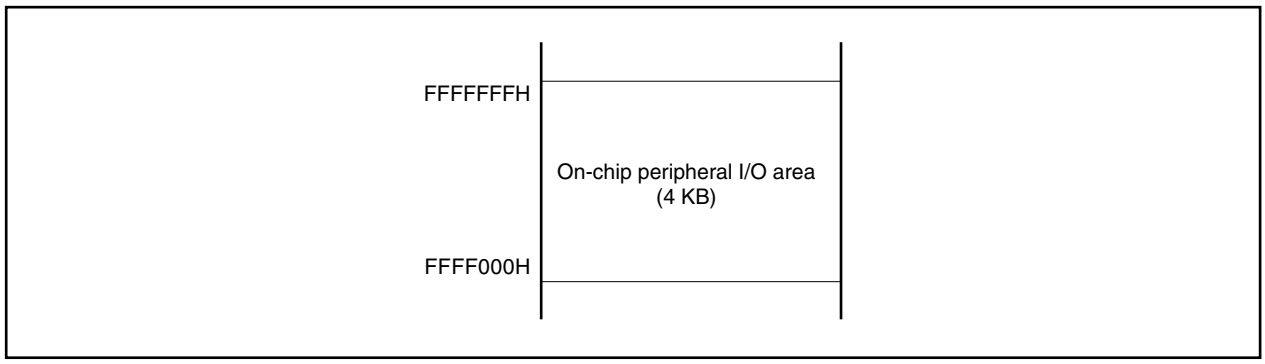


(3) On-chip peripheral I/O area

4 KB of memory, addresses FFFF000H to FFFFFFFH, are provided as an on-chip peripheral I/O area.

An image of addresses FFFF000H to FFFFFFFH can be seen in the area between addresses 3FFF000H and 3FFFFFFH[Note].

Note Access to the area of addresses 3FFF000H to 3FFFFFFH is prohibited. To access the on-chip peripheral I/O, specify addresses FFFF000H to FFFFFFFH.



On-chip peripheral I/O registers associated with the operation mode specification and the state monitoring for the on-chip peripheral I/O are all memory-mapped to the on-chip peripheral I/O area. Program fetches cannot be executed from this area.

Cautions 1. The least significant bit of an address is not decoded. Therefore, if byte access is executed in the register at an odd address (2n + 1), the register at the even address (2n) will be accessed because of the hardware specification.

2. In the V850E/IA2, no registers exist that are capable of word access, but if a register is word accessed, halfword access is performed twice in the order of lower address, then higher address of the word area, ignoring the lower 2 bits of the address.

3. For registers in which byte access is possible, if halfword access is executed, the higher 8 bits become undefined during the read operation, and the lower 8 bits of data are written to the register during the write operation.

4. Addresses that are not defined as registers are reserved for future expansion. If these addresses are accessed, the operation is undefined and not guaranteed.

5. Addresses 3FFF000H to 3FFFFFFH cannot be specified as the source/destination address of DMA transfer. Be sure to use addresses FFFF000H to FFFFFFFH for the source/destination address of DMA transfer.

(4) External memory area

4 MB are available for external memory area.

- Single-chip mode: x100000H to x3FFFFFH
- ROMless mode: x000000H to x3FFFFFH

Note that the internal ROM, internal RAM, and on-chip peripheral I/O areas cannot be accessed as external memory areas.

3.4.6 External memory expansion

By setting the port n mode control register (PMCn) to control mode, an external device can be connected to the external memory space using each pin of ports DH, DL, CT, and CM. Each register is set by selecting control mode for each pin of these ports using PMCn (n = DH, DL, CT, CM).

Note that the status after reset differs as shown below in accordance with the operating mode specification set by the MODE0 and MODE1 pins (refer to **3.3 Operation Modes** for details of the operation modes).

(a) In the case of ROMless mode

Because each pin of ports DH, DL, CT, and CM enters control mode following a reset, external memory can be used without making changes to the port n mode control register (PMCn) (the external data bus width is 16 bits).

(b) In the case of single-chip mode

Since the internal ROM area is accessed after a reset, each pin of ports DH, DL, CT, and CM enters the port mode, and external devices cannot be used.

To use external memory, set the port n mode control register (PMCn).

Remark n = DH, DL, CT, CM

3.4.7 Recommended use of address space

The architecture of the V850E/IA2 requires that a register that serves as a pointer be secured for address generation when accessing operand data in the data space. Operand data access from instruction can be directly executed at the address in this pointer register ±32 KB. However, because there is a limit to which general-purpose registers are used as a pointer register, by minimizing the deterioration of address calculation performance when changing the pointer value, the number of usable general-purpose registers for handling variables is maximized, and the program size can be saved.

To enhance the efficiency of using the pointer in connection with of the memory map of the V850E/IA2, the following points are recommended.

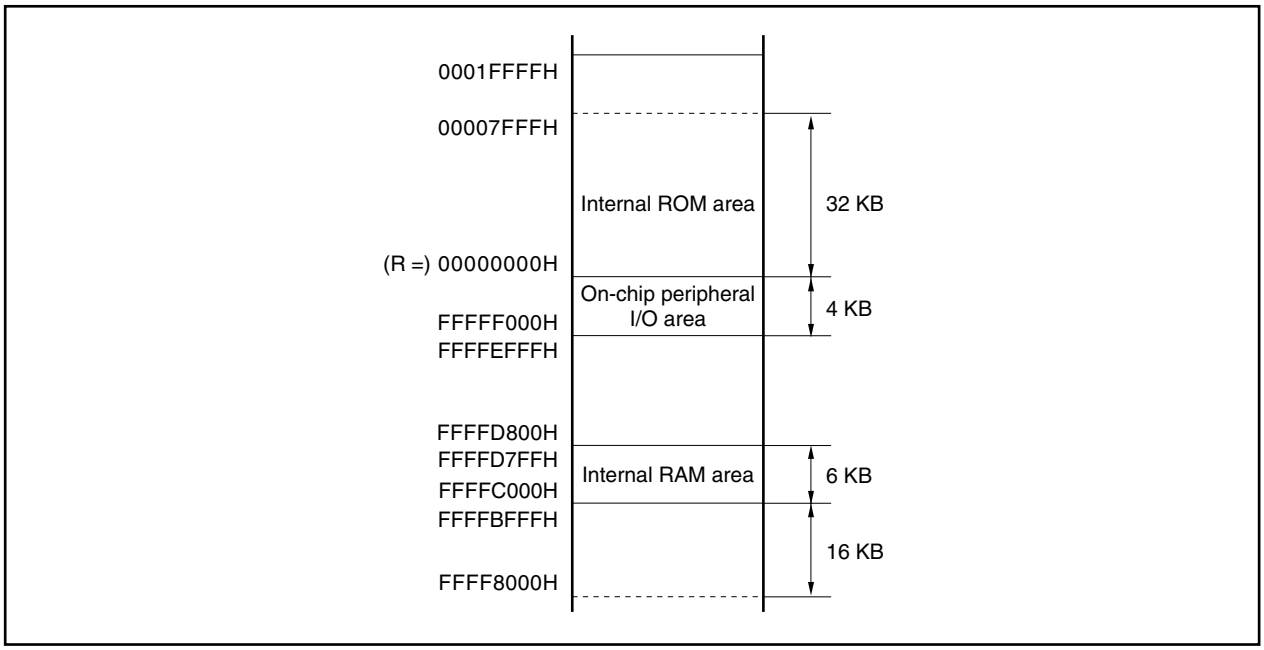
(1) Program space

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. Therefore, a contiguous 64 MB space, starting from address 00000000H, corresponds to the memory map of the program space.

(2) Data space

For the efficient use of resources that make use of the wrap-around feature of the data space, the continuous 16 MB address spaces 00000000H to 00FFFFFFH and FF000000H to FFFFFFFFH of the 4 GB CPU are used as the data space. With the V850E/IA2, a 256 MB physical address space is seen as 16 images in the 4 GB CPU address space. The highest bit (bit 25) of this 26-bit address is assigned as address sign-extended to 32 bits.

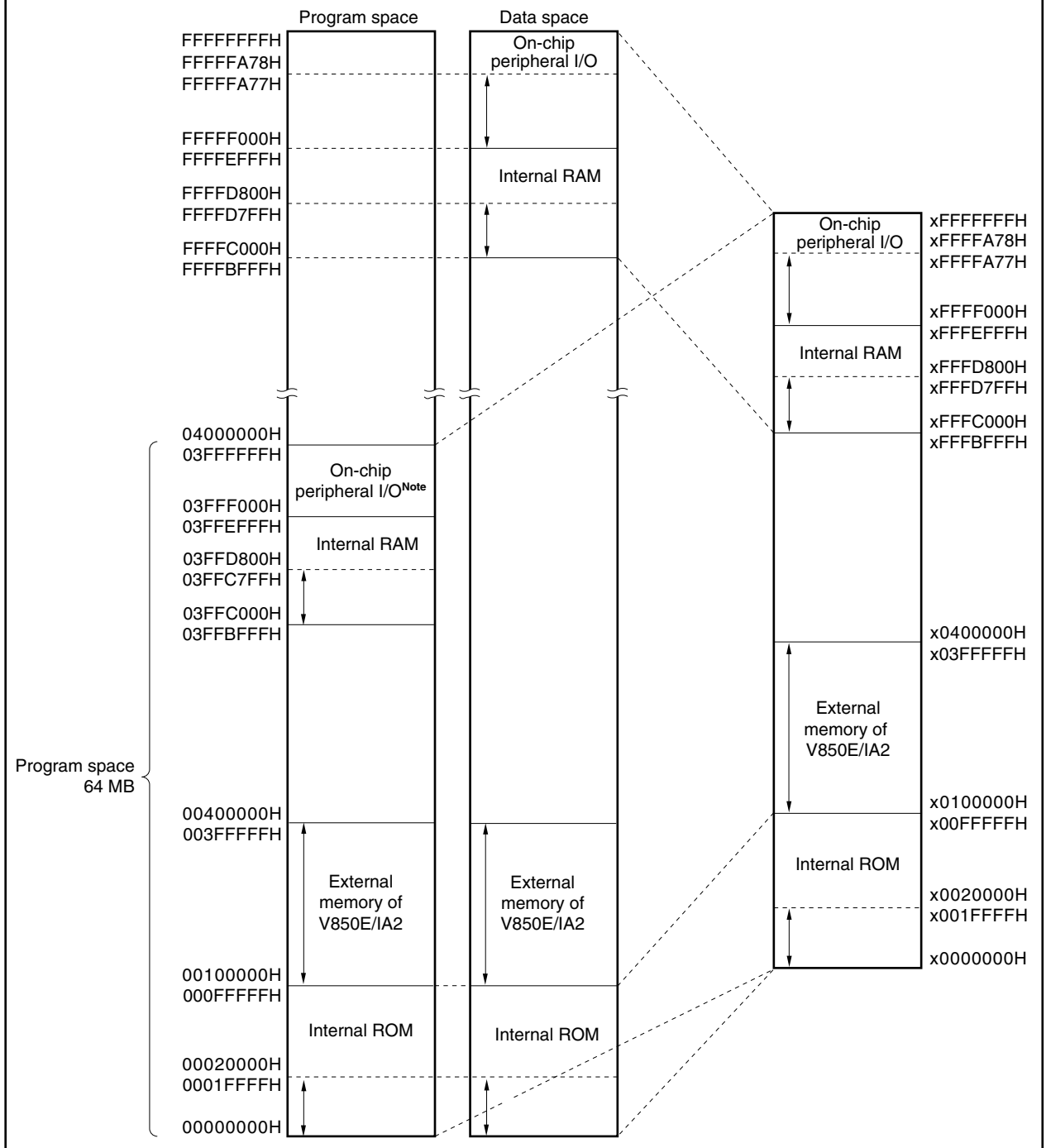
Example Application of wrap-around



When R = r0 (zero register) is specified with the LD/ST disp16 [R] instruction, an addressing range of 00000000H ±32 KB can be referenced by the sign-extended disp 16. By mapping the external memory in the 16 KB area in the figure, all resources of internal hardware can be accessed with one pointer.

The zero register (r0) is a register set to 0 by the hardware, and eliminates the need for additional registers for the pointer.

Figure 3-5. Recommended Memory Map



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

Remarks 1. The arrows indicate the recommended area.

2. This is a recommended memory map when the V850E/IA2 is set to single-chip mode, and used in external expansion mode.

3.4.8 On-chip peripheral I/O registers

(1/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF004H	Port DL	PDL	R/W			√	Undefined
FFFFF004H	Port DLL	PDLL	R/W	√	√		Undefined
FFFFF005H	Port DLH	PDLH	R/W	√	√		Undefined
FFFFF006H	Port DH	PDH	R/W	√	√		Undefined
FFFFF00AH	Port CT	PCT	R/W	√	√		Undefined
FFFFF00CH	Port CM	PCM	R/W	√	√		Undefined
FFFFF024H	Port DL mode register	PMDL	R/W			√	FFFFH
FFFFF024H	Port DL mode register L	PMDLL	R/W	√	√		FFH
FFFFF025H	Port DL mode register H	PMDLH	R/W	√	√		FFH
FFFFF026H	Port DH mode register	PMDH	R/W	√	√		FFH
FFFFF02AH	Port CT mode register	PMCT	R/W	√	√		FFH
FFFFF02CH	Port CM mode register	PMCM	R/W	√	√		FFH
FFFFF044H	Port DL mode control register	PMCDL	R/W			√	0000H/FFFFH
FFFFF044H	Port DL mode control register L	PMCDLL	R/W	√	√		00H/FFH
FFFFF045H	Port DL mode control register H	PMCDLH	R/W	√	√		00H/FFH
FFFFF046H	Port DH mode control register	PMCDH	R/W	√	√		00H/FFH
FFFFF04AH	Port CT mode control register	PMCCT	R/W	√	√		00H/53H
FFFFF04CH	Port CM mode control register	PMCCM	R/W	√	√		00H/03H
FFFFF060H	Chip area selection control register 0	CSC0	R/W			√	2C11H
FFFFF062H	Chip area selection control register 1	CSC1	R/W			√	2C11H
FFFFF066H	Bus size configuration register	BSC	R/W			√	5555H
FFFFF06EH	System wait control register	VSWC	R/W		√		77H
FFFFF080H	DMA source address register 0L	DSA0L	R/W			√	Undefined
FFFFF082H	DMA source address register 0H	DSA0H	R/W			√	Undefined
FFFFF084H	DMA destination address register 0L	DDA0L	R/W			√	Undefined
FFFFF086H	DMA destination address register 0H	DDA0H	R/W			√	Undefined
FFFFF088H	DMA source address register 1L	DSA1L	R/W			√	Undefined
FFFFF08AH	DMA source address register 1H	DSA1H	R/W			√	Undefined
FFFFF08CH	DMA destination address register 1L	DDA1L	R/W			√	Undefined
FFFFF08EH	DMA destination address register 1H	DDA1H	R/W			√	Undefined
FFFFF090H	DMA source address register 2L	DSA2L	R/W			√	Undefined
FFFFF092H	DMA source address register 2H	DSA2H	R/W			√	Undefined
FFFFF094H	DMA destination address register 2L	DDA2L	R/W			√	Undefined
FFFFF096H	DMA destination address register 2H	DDA2H	R/W			√	Undefined
FFFFF098H	DMA source address register 3L	DSA3L	R/W			√	Undefined
FFFFF09AH	DMA source address register 3H	DSA3H	R/W			√	Undefined
FFFFF09CH	DMA destination address register 3L	DDA3L	R/W			√	Undefined

(2/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF09EH	DMA destination address register 3H	DDA3H	R/W			√	Undefined
FFFFF0C0H	DMA transfer count register 0	DBC0	R/W			√	Undefined
FFFFF0C2H	DMA transfer count register 1	DBC1	R/W			√	Undefined
FFFFF0C4H	DMA transfer count register 2	DBC2	R/W			√	Undefined
FFFFF0C6H	DMA transfer count register 3	DBC3	R/W			√	Undefined
FFFFF0D0H	DMA addressing control register 0	DADC0	R/W			√	0000H
FFFFF0D2H	DMA addressing control register 1	DADC1	R/W			√	0000H
FFFFF0D4H	DMA addressing control register 2	DADC2	R/W			√	0000H
FFFFF0D6H	DMA addressing control register 3	DADC3	R/W			√	0000H
FFFFF0E0H	DMA channel control register 0	DCHC0	R/W	√	√		00H
FFFFF0E2H	DMA channel control register 1	DCHC1	R/W	√	√		00H
FFFFF0E4H	DMA channel control register 2	DCHC2	R/W	√	√		00H
FFFFF0E6H	DMA channel control register 3	DCHC3	R/W	√	√		00H
FFFFF0F0H	DMA disable status register	DDIS	R		√		00H
FFFFF0F2H	DMA restart register	DRST	R/W		√		00H
FFFFF100H	Interrupt mask register 0	IMR0	R/W			√	FFFFH
FFFFF100H	Interrupt mask register 0L	IMR0L	R/W	√	√		FFH
FFFFF101H	Interrupt mask register 0H	IMR0H	R/W	√	√		FFH
FFFFF102H	Interrupt mask register 1	IMR1	R/W			√	FFFFH
FFFFF102H	Interrupt mask register 1L	IMR1L	R/W	√	√		FFH
FFFFF103H	Interrupt mask register 1H	IMR1H	R/W	√	√		FFH
FFFFF104H	Interrupt mask register 2	IMR2	R/W			√	FFFFH
FFFFF104H	Interrupt mask register 2L	IMR2L	R/W	√	√		FFH
FFFFF105H	Interrupt mask register 2H	IMR2H	R/W	√	√		FFH
FFFFF106H	Interrupt mask register 3	IMR3	R/W			√	FFFFH
FFFFF106H	Interrupt mask register 3L	IMR3L	R/W	√	√		FFH
FFFFF107H	Interrupt mask register 3H	IMR3H	R/W	√	√		FFH
FFFFF110H	Interrupt control register	P0IC0	R/W	√	√		47H
FFFFF112H	Interrupt control register	P0IC1	R/W	√	√		47H
FFFFF114H	Interrupt control register	P0IC2	R/W	√	√		47H
FFFFF116H	Interrupt control register	P0IC3	R/W	√	√		47H
FFFFF118H	Interrupt control register	P0IC4	R/W	√	√		47H
FFFFF11EH	Interrupt control register	DETIC0	R/W	√	√		47H
FFFFF120H	Interrupt control register	DETIC1	R/W	√	√		47H
FFFFF122H	Interrupt control register	TM0IC0	R/W	√	√		47H
FFFFF124H	Interrupt control register	CM03IC0	R/W	√	√		47H
FFFFF126H	Interrupt control register	TM0IC1	R/W	√	√		47H
FFFFF128H	Interrupt control register	CM03IC1	R/W	√	√		47H

(3/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF12AH	Interrupt control register	CC10IC0	R/W	√	√		47H
FFFFF12CH	Interrupt control register	CC10IC1	R/W	√	√		47H
FFFFF12EH	Interrupt control register	CM10IC0	R/W	√	√		47H
FFFFF130H	Interrupt control register	CM10IC1	R/W	√	√		47H
FFFFF13AH	Interrupt control register	TM2IC0	R/W	√	√		47H
FFFFF13CH	Interrupt control register	TM2IC1	R/W	√	√		47H
FFFFF13EH	Interrupt control register	CC2IC0	R/W	√	√		47H
FFFFF140H	Interrupt control register	CC2IC1	R/W	√	√		47H
FFFFF142H	Interrupt control register	CC2IC2	R/W	√	√		47H
FFFFF144H	Interrupt control register	CC2IC3	R/W	√	√		47H
FFFFF146H	Interrupt control register	CC2IC4	R/W	√	√		47H
FFFFF148H	Interrupt control register	CC2IC5	R/W	√	√		47H
FFFFF14AH	Interrupt control register	TM3IC0	R/W	√	√		47H
FFFFF14CH	Interrupt control register	CC3IC0	R/W	√	√		47H
FFFFF14EH	Interrupt control register	CC3IC1	R/W	√	√		47H
FFFFF150H	Interrupt control register	CM4IC0	R/W	√	√		47H
FFFFF152H	Interrupt control register	DMAIC0	R/W	√	√		47H
FFFFF154H	Interrupt control register	DMAIC1	R/W	√	√		47H
FFFFF156H	Interrupt control register	DMAIC2	R/W	√	√		47H
FFFFF158H	Interrupt control register	DMAIC3	R/W	√	√		47H
FFFFF162H	Interrupt control register	CSIIC0	R/W	√	√		47H
FFFFF164H	Interrupt control register	CSIIC1	R/W	√	√		47H
FFFFF166H	Interrupt control register	SRIC0	R/W	√	√		47H
FFFFF168H	Interrupt control register	STIC0	R/W	√	√		47H
FFFFF16AH	Interrupt control register	SEIC0	R/W	√	√		47H
FFFFF16CH	Interrupt control register	SRIC1	R/W	√	√		47H
FFFFF16EH	Interrupt control register	STIC1	R/W	√	√		47H
FFFFF174H	Interrupt control register	ADIC0	R/W	√	√		47H
FFFFF176H	Interrupt control register	ADIC1	R/W	√	√		47H
FFFFF17EH	Interrupt control register	CM00IC1	R/W	√	√		47H
FFFFF180H	Interrupt control register	CM01IC1	R/W	√	√		47H
FFFFF182H	Interrupt control register	CM02IC1	R/W	√	√		47H
FFFFF184H	Interrupt control register	CM04IC1	R/W	√	√		47H
FFFFF186H	Interrupt control register	CM05IC1	R/W	√	√		47H
FFFFF188H	Interrupt control register	CM04IC0	R/W	√	√		47H
FFFFF18AH	Interrupt control register	CM05IC0	R/W	√	√		47H
FFFFF1FAH	In-service priority register	ISPR	R	√	√		00H
FFFFF1FCH	Command register	PRCMD	W		√		Undefined

(4/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF1FEH	Power save control register	PSC	R/W	√	√		00H
FFFFF200H	A/D scan mode register 00	ADSCM00	R/W			√	0000H
FFFFF200H	A/D scan mode register 00L	ADSCM00L	R/W	√	√		00H
FFFFF201H	A/D scan mode register 00H	ADSCM00H	R/W	√	√		00H
FFFFF202H	A/D scan mode register 01	ADSCM01	R/W			√	0000H
FFFFF202H	A/D scan mode register 01L	ADSCM01L	R		√		00H
FFFFF203H	A/D scan mode register 01H	ADSCM01H	R/W	√	√		00H
FFFFF204H	A/D voltage detection mode register 0	ADETM0	R/W			√	0000H
FFFFF204H	A/D voltage detection mode register 0L	ADETM0L	R/W	√	√		00H
FFFFF205H	A/D voltage detection mode register 0H	ADETM0H	R/W	√	√		00H
FFFFF210H	A/D conversion result register 00	ADCR00	R			√	0000H
FFFFF212H	A/D conversion result register 01	ADCR01	R			√	0000H
FFFFF214H	A/D conversion result register 02	ADCR02	R			√	0000H
FFFFF216H	A/D conversion result register 03	ADCR03	R			√	0000H
FFFFF218H	A/D conversion result register 04	ADCR04	R			√	0000H
FFFFF21AH	A/D conversion result register 05	ADCR05	R			√	0000H
FFFFF240H	A/D scan mode register 10	ADSCM10	R/W			√	0000H
FFFFF240H	A/D scan mode register 10L	ADSCM10L	R/W	√	√		00H
FFFFF241H	A/D scan mode register 10H	ADSCM10H	R/W	√	√		00H
FFFFF242H	A/D scan mode register 11	ADSCM11	R/W			√	0000H
FFFFF242H	A/D scan mode register 11L	ADSCM11L	R		√		00H
FFFFF243H	A/D scan mode register 11H	ADSCM11H	R/W	√	√		00H
FFFFF244H	A/D voltage detection mode register 1	ADETM1	R/W			√	0000H
FFFFF244H	A/D voltage detection mode register 1L	ADETM1L	R/W	√	√		00H
FFFFF245H	A/D voltage detection mode register 1H	ADETM1H	R/W	√	√		00H
FFFFF250H	A/D conversion result register 10	ADCR10	R			√	0000H
FFFFF252H	A/D conversion result register 11	ADCR11	R			√	0000H
FFFFF254H	A/D conversion result register 12	ADCR12	R			√	0000H
FFFFF256H	A/D conversion result register 13	ADCR13	R			√	0000H
FFFFF258H	A/D conversion result register 14	ADCR14	R			√	0000H
FFFFF25AH	A/D conversion result register 15	ADCR15	R			√	0000H
FFFFF25CH	A/D conversion result register 16	ADCR16	R			√	0000H
FFFFF25EH	A/D conversion result register 17	ADCR17	R			√	0000H
FFFFF280H	A/D internal trigger select register 0	ITRG0	R/W	√	√		00H
FFFFF288H	A/D internal trigger select register 1	ITRG1	R/W	√	√		00H
FFFFF300H	Regulator control register	REGC	R/W	√	√		00H
FFFFF400H	Port 0	P0	R	√	√		Undefined
FFFFF402H	Port 1	P1	R/W	√	√		Undefined

(5/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF404H	Port 2	P2	R/W	√	√		Undefined
FFFFF406H	Port 3	P3	R/W	√	√		Undefined
FFFFF408H	Port 4	P4	R/W	√	√		Undefined
FFFFF422H	Port 1 mode register	PM1	R/W	√	√		FFH
FFFFF424H	Port 2 mode register	PM2	R/W	√	√		FFH
FFFFF426H	Port 3 mode register	PM3	R/W	√	√		FFH
FFFFF428H	Port 4 mode register	PM4	R/W	√	√		FFH
FFFFF442H	Port 1 mode control register	PMC1	R/W	√	√		00H
FFFFF444H	Port 2 mode control register	PMC2	R/W	√	√		00H
FFFFF446H	Port 3 mode control register	PMC3	R/W	√	√		00H
FFFFF448H	Port 4 mode control register	PMC4	R/W	√	√		00H
FFFFF462H	Port 1 function control register	PFC1	R/W	√	√		00H
FFFFF464H	Port 2 function control register	PFC2	R/W	√	√		00H
FFFFF466H	Port 3 function control register	PFC3	R/W	√	√		00H
FFFFF480H	Bus cycle type configuration register 0	BCT0	R/W			√	CCCCH
FFFFF482H	Bus cycle type configuration register 1	BCT1	R/W			√	CCCCH
FFFFF484H	Data wait control register 0	DWC0	R/W			√	3333H
FFFFF486H	Data wait control register 1	DWC1	R/W			√	3333H
FFFFF488H	Address wait control register	AWC	R/W			√	0000H
FFFFF48AH	Bus cycle control register	BCC	R/W			√	AAAAH
FFFFF540H	Timer 4	TM4	R			√	0000H
FFFFF542H	Compare register 4	CM4	R/W			√	0000H
FFFFF544H	Timer control register 4	TMC4	R/W	√	√		00H
FFFFF570H	Dead time timer reload register 0	DTRR0	R/W			√	0FFFH
FFFFF572H	Buffer register CM00	BFCM00	R/W			√	FFFFH
FFFFF574H	Buffer register CM01	BFCM01	R/W			√	FFFFH
FFFFF576H	Buffer register CM02	BFCM02	R/W			√	FFFFH
FFFFF578H	Buffer register CM03	BFCM03	R/W			√	FFFFH
FFFFF57AH	Timer control register 00	TMC00	R/W			√	0508H
FFFFF57AH	Timer control register 00L	TMC00L	R/W	√	√		08H
FFFFF57BH	Timer control register 00H	TMC00H	R/W	√	√		05H
FFFFF57CH	Timer unit control register 00	TUC00	R/W	√	√		01H
FFFFF57DH	Timer output mode register 0	TOMR0	R/W		√		00H
FFFFF57EH	PWM software timing output register 0	PSTO0	R/W	√	√		00H
FFFFF57FH	PWM output enable register 0	POER0	R/W	√	√		00H
FFFFF580H	TOMR write enable register 0	SPEC0	R/W			√	0000H
FFFFF59CH	Buffer register CM04	BFCM04	R/W			√	FFFFH

(6/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF59EH	Buffer register CM05	BFCM05	R/W			√	FFFFH
FFFFF5B0H	Dead time timer reload register 1	DTRR1	R/W			√	0FFFH
FFFFF5B2H	Buffer register CM10	BFCM10	R/W			√	FFFFH
FFFFF5B4H	Buffer register CM11	BFCM11	R/W			√	FFFFH
FFFFF5B6H	Buffer register CM12	BFCM12	R/W			√	FFFFH
FFFFF5B8H	Buffer register CM13	BFCM13	R/W			√	FFFFH
FFFFF5BAH	Timer control register 01	TMC01	R/W			√	0508H
FFFFF5BAH	Timer control register 01L	TMC01L	R/W	√	√		08H
FFFFF5BBH	Timer control register 01H	TMC01H	R/W	√	√		05H
FFFFF5BCH	Timer unit control register 01	TUC01	R/W	√	√		01H
FFFFF5BDH	Timer output mode register 1	TOMR1	R/W		√		00H
FFFFF5BEH	PWM software timing output register 1	PSTO1	R/W	√	√		00H
FFFFF5BFH	PWM output enable register 1	POER1	R/W	√	√		00H
FFFFF5C0H	TOMR write enable register 1	SPEC1	R/W			√	0000H
FFFFF5D0H	Timer 0 clock select register	PRM01	R/W	√	√		00H
FFFFF5D8H	Timer 1/timer 2 clock selection register	PRM02	R/W	√	√		00H
FFFFF5DCH	Buffer register CM14	BFCM14	R/W			√	FFFFH
FFFFF5DEH	Buffer register CM15	BFCM15	R/W			√	FFFFH
FFFFF5E0H	Timer 10	TM10	R/W			√	0000H
FFFFF5E2H	Compare register 100	CM100	R/W			√	0000H
FFFFF5E4H	Compare register 101	CM101	R/W			√	0000H
FFFFF5E6H	Capture/compare register 100	CC100	R/W			√	0000H
FFFFF5E8H	Capture/compare register 101	CC101	R/W			√	0000H
FFFFF5EAH	Capture/compare control register 0	CCR0	R/W	√	√		00H
FFFFF5EBH	Timer unit mode register 0	TUM0	R/W	√	√		00H
FFFFF5ECH	Timer control register 10	TMC10	R/W	√	√		00H
FFFFF5EDH	Signal edge selection register 10	SESA10	R/W	√	√		00H
FFFFF5EEH	Prescaler mode register 10	PRM10	R/W	√	√		07H
FFFFF5EFH	Status register 0	STATUS0	R	√	√		00H
FFFFF5F6H	CC101 capture input selection register	CSL10	R/W	√	√		00H
FFFFF5F8H	Timer 10 noise elimination time select register	NRC10	R/W	√	√		00H
FFFFF620H	Timer connection selection register 0	TMIC0	R/W	√	√		00H
FFFFF630H	Timer 2 input filter mode register 0	FEM0	R/W	√	√		00H
FFFFF631H	Timer 2 input filter mode register 1	FEM1	R/W	√	√		00H
FFFFF632H	Timer 2 input filter mode register 2	FEM2	R/W	√	√		00H
FFFFF633H	Timer 2 input filter mode register 3	FEM3	R/W	√	√		00H
FFFFF634H	Timer 2 input filter mode register 4	FEM4	R/W	√	√		00H

(7/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF635H	Timer 2 input filter mode register 5	FEM5	R/W	√	√		00H
FFFFF640H	Timer 2 clock stop register 0	STOPTE0	R/W			√	0000H
FFFFF640H	Timer 2 clock stop register 0L	STOPTE0L	R		√		00H
FFFFF641H	Timer 2 clock stop register 0H	STOPTE0H	R/W	√	√		00H
FFFFF642H	Timer 2 count clock/control edge selection register 0	CSE0	R/W			√	0000H
FFFFF642H	Timer 2 count clock/control edge selection register 0L	CSE0L	R/W	√	√		00H
FFFFF643H	Timer 2 count clock/control edge selection register 0H	CSE0H	R/W	√	√		00H
FFFFF644H	Timer 2 subchannel input event edge selection register 0	SESE0	R/W			√	0000H
FFFFF644H	Timer 2 subchannel input event edge selection register 0L	SESE0L	R/W	√	√		00H
FFFFF645H	Timer 2 subchannel input event edge selection register 0H	SESE0H	R/W	√	√		00H
FFFFF646H	Timer 2 time base control register 0	TCRE0	R/W			√	0000H
FFFFF646H	Timer 2 time base control register 0L	TCRE0L	R/W	√	√		00H
FFFFF647H	Timer 2 time base control register 0H	TCRE0H	R/W	√	√		00H
FFFFF648H	Timer 2 output control register 0	OCTLE0	R/W			√	0000H
FFFFF648H	Timer 2 output control register 0L	OCTLE0L	R/W	√	√		00H
FFFFF649H	Timer 2 output control register 0H	OCTLE0H	R/W	√	√		00H
FFFFF64AH	Timer 2 subchannel 0, 5 capture/compare control register	CMSE050	R/W			√	0000H
FFFFF64CH	Timer 2 subchannel 1, 2 capture/compare control register	CMSE120	R/W			√	0000H
FFFFF64EH	Timer 2 subchannel 3, 4 capture/compare control register	CMSE340	R/W			√	0000H
FFFFF650H	Timer 2 subchannel 1 sub capture/compare register	CVSE10	R/W			√	0000H
FFFFF652H	Timer 2 subchannel 1 main capture/compare register	CVPE10	R			√	0000H
FFFFF654H	Timer 2 subchannel 2 sub capture/compare register	CVSE20	R/W			√	0000H
FFFFF656H	Timer 2 subchannel 2 main capture/compare register	CVPE20	R			√	0000H
FFFFF658H	Timer 2 subchannel 3 sub capture/compare register	CVSE30	R/W			√	0000H
FFFFF65AH	Timer 2 subchannel 3 main capture/compare register	CVPE30	R			√	0000H
FFFFF65CH	Timer 2 subchannel 4 sub capture/compare register	CVSE40	R/W			√	0000H

(8/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF65EH	Timer 2 subchannel 4 main capture/compare register	CVPE40	R			√	0000H
FFFFF660H	Timer 2 subchannel 0 capture/compare register	CVSE00	R/W			√	0000H
FFFFF662H	Timer 2 subchannel 5 capture/compare register	CVSE50	R/W			√	0000H
FFFFF664H	Timer 2 time base status register 0	TBSTATE0	R/W			√	0101H
FFFFF664H	Timer 2 time base status register 0L	TBSTATE0L	R/W	√	√		01H
FFFFF665H	Timer 2 time base status register 0H	TBSTATE0H	R/W	√	√		01H
FFFFF666H	Timer 2 capture/compare 1 to 4 status register 0	CCSTATE0	R/W			√	0000H
FFFFF666H	Timer 2 capture/compare 1 to 4 status register 0L	CCSTATE0L	R/W	√	√		00H
FFFFF667H	Timer 2 capture/compare 1 to 4 status register 0H	CCSTATE0H	R/W	√	√		00H
FFFFF668H	Timer 2 output delay register 0	ODELE0	R/W			√	0000H
FFFFF668H	Timer 2 output delay register 0L	ODELE0L	R/W	√	√		00H
FFFFF669H	Timer 2 output delay register 0H	ODELE0H	R/W	√	√		00H
FFFFF66AH	Timer 2 software event capture register	CSCE0	R/W			√	0000H
FFFFF680H	Timer 3	TM3	R			√	0000H
FFFFF682H	Capture/compare register 30	CC30	R/W			√	0000H
FFFFF684H	Capture/compare register 31	CC31	R/W			√	0000H
FFFFF686H	Timer control register 30	TMC30	R/W	√	√		00H
FFFFF688H	Timer control register 31	TMC31	R/W	√	√		20H
FFFFF689H	Valid edge selection register	SESC	R/W	√	√		00H
FFFFF690H	Timer 3 clock selection register	PRM03	R/W	√	√		00H
FFFFF698H	Timer 3 noise elimination time selection register	NRC3	R/W	√	√		00H
FFFFF6A0H	Timer 3 output control register	TO3C	R/W	√	√		00H
FFFFF800H	Peripheral command register	PHCMD	W		√		Undefined
FFFFF802H	Peripheral status register	PHS	R/W	√	√		00H
FFFFF810H	DMA trigger factor register 0	DTFR0	R/W	√	√		00H
FFFFF812H	DMA trigger factor register 1	DTFR1	R/W	√	√		00H
FFFFF814H	DMA trigger factor register 2	DTFR2	R/W	√	√		00H
FFFFF816H	DMA trigger factor register 3	DTFR3	R/W	√	√		00H
FFFFF820H	Power save mode register	PSMR	R/W	√	√		00H
FFFFF822H	Clock control register	CKC	R/W		√		00H
FFFFF824H	Lock register	LOCKR	R	√	√		0000000xB
FFFFF880H	External interrupt mode register 0	INTM0	R/W	√	√		00H

(9/10)

	Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
					1 Bit	8 Bits	16 Bits	
	FFFFF882H	External interrupt mode register 1	INTM1	R/W	√	√		00H
	FFFFF884H	External interrupt mode register 2	INTM2	R/W	√	√		00H
★	FFFFF8D4H	Flash programming mode control register	FLPMC	R/W	√	√		08H/0CH/00H[Note]
	FFFFF900H	Clocked serial interface mode register 0	CSIM0	R/W	√	√		00H
	FFFFF901H	Clocked serial interface clock selection register 0	CSIC0	R/W	√	√		00H
	FFFFF902H	Clocked serial interface receive buffer register 0	SIRB0	R			√	0000H
	FFFFF902H	Clocked serial interface receive buffer register L0	SIRBL0	R	√	√		00H
	FFFFF904H	Clocked serial interface transmit buffer register 0	SOTB0	R/W			√	0000H
	FFFFF904H	Clocked serial interface transmit buffer register L0	SOTBL0	R/W	√	√		00H
	FFFFF906H	Clocked serial interface read-only receive buffer register 0	SIRBE0	R			√	0000H
	FFFFF906H	Clocked serial interface read-only receive buffer register L0	SIRBEL0	R	√	√		00H
	FFFFF908H	Clocked serial interface initial transmit buffer register 0	SOTBF0	R/W			√	0000H
	FFFFF908H	Clocked serial interface initial transmit buffer register L0	SOTBFL0	R/W	√	√		00H
	FFFFF90AH	Serial I/O shift register 0	SIO0	R			√	0000H
	FFFFF90AH	Serial I/O shift register L0	SIOL0	R			√	0000H
	FFFFF910H	Clocked serial interface mode register 1	CSIM1	R/W	√	√		00H
	FFFFF911H	Clocked serial interface clock selection register 1	CSIC1	R/W	√	√		00H
	FFFFF912H	Clocked serial interface receive buffer register 1	SIRB1	R			√	0000H
	FFFFF912H	Clocked serial interface receive buffer register L1	SIRBL1	R			√	0000H
	FFFFF914H	Clocked serial interface transmit buffer register 1	SOTB1	R/W			√	0000H
	FFFFF914H	Clocked serial interface transmit buffer register L1	SOTBL1	R/W	√	√		00H
	FFFFF916H	Clocked serial interface read-only receive buffer register 1	SIRBE1	R			√	0000H
	FFFFF916H	Clocked serial interface read-only receive buffer register L1	SIRBEL1	R	√	√		00H
	FFFFF918H	Clocked serial interface initial transmit buffer register 1	SOTBF1	R/W			√	0000H
	FFFFF918H	Clocked serial interface initial transmit buffer register L1	SOTBFL1	R/W	√	√		00H
	FFFFF91AH	Serial I/O shift register 1	SIO1	R			√	0000H
	FFFFF91AH	Serial I/O shift register L1	SIOL1	R	√	√		00H

★ **Note** μPD703114: 00H

μPD70F3114: 08H or 0CH (For details, refer to **15.7.12 Flash programming mode control register (FLPMC)**.)

(10/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF920H	Prescaler mode register 3	PRSM3	R/W	√	√		00H
FFFFF922H	Prescaler compare register 3	PRSCM3	R/W		√		00H
FFFFFA00H	Asynchronous serial interface mode register 0	ASIM0	R/W	√	√		01H
FFFFFA02H	Receive buffer register 0	RXB0	R		√		FFH
FFFFFA03H	Asynchronous serial interface status register 0	ASIS0	R		√		00H
FFFFFA04H	Transmit buffer register 0	TXB0	R/W		√		FFH
FFFFFA05H	Asynchronous serial interface transmit status register 0	ASIF0	R	√	√		00H
FFFFFA06H	Clock select register 0	CKSR0	R/W		√		00H
FFFFFA07H	Baud rate generator control register 0	BRGC0	R/W		√		FFH
FFFFFA20H	2-frame continuous reception buffer register 1	RXB1	R			√	Undefined
FFFFFA22H	Receive buffer register L1	RXBL1	R		√		Undefined
FFFFFA24H	2-frame continuous transmission shift register 1	TXS1	W			√	Undefined
FFFFFA26H	Transmit shift register L1	TXSL1	W		√		Undefined
FFFFFA28H	Asynchronous serial interface mode register 10	ASIM10	R/W	√	√		81H
FFFFFA2AH	Asynchronous serial interface mode register 11	ASIM11	R/W	√	√		00H
FFFFFA2CH	Asynchronous serial interface status register 1	ASIS1	R	√	√		00H
FFFFFA2EH	Prescaler mode register 1	PRSM1	R/W	√	√		00H
FFFFFA30H	Prescaler compare register 1	PRSCM1	R/W		√		00H

3.4.9 Specific registers

Specific registers are registers that are protected from being written with illegal data due to inadvertent program loop (runaway), etc. The V850E/IA2 has three specific registers, the power save control register (PSC) (refer to **8.5.2 (3) Power save control register (PSC)**), clock control register (CKC) (refer to **8.3.4 Clock control register (CKC)**), ★ and flash programming mode control register (FLPMC) (refer to **15.7.12 Flash programming mode control register (FLPMC)**).

3.4.10 System wait control register (VSWC)

The system wait control register (VSWC) controls the wait cycles of a bus access to the on-chip peripheral I/O registers.

Set the following values to this register.

Set value of VSWC: 02H (when two wait clocks are set, with operating frequency (f_{XX}) = 40 MHz)

This register can be read/written in 8-bit units (address: FFFFF06EH, after reset: 77H).

Remark If the timing at which the flag or count value changes overlaps the register access timing when a register that includes a status flag indicating the status of on-chip peripheral functions (ASIF0, etc.) or a register that indicates a timer count value (TM0n, etc.) are accessed, a register access retry operation occurs. Therefore, it may take longer than normal to access an on-chip peripheral register.

3.4.11 Cautions

(1) Register to be set first

When using the V850E/IA2, the following registers must be set from the beginning.

- System wait control register (VSWC)
  (See **3.4.10 System wait control register (VSWC)**)
- Clock control register (CKC)
  (See **8.3.4 Clock control register (CKC)**)

After setting VSWC and CKC, set other registers as required.

★ **(2) Restriction on conflict between sld instruction and interrupt request**

(a) Description

If a conflict occurs between the decode operation of an instruction in <2> immediately before the sld instruction following an instruction in <1> and an interrupt request before the instruction in <1> is complete, the execution result of the instruction in <1> may not be stored in a register.

Instruction <1>

- ld instruction: ld.b, ld.h, ld.w, ld.bu, ld.hu
- sld instruction: sld.b, sld.h, sld.w, sld.bu, sld.hu
- Multiplication instruction: mul, mulh, mulhi, mulu

Instruction <2>

mov reg1, reg2	not reg1, reg2	satsubr reg1, reg2	satsub reg1, reg2
satadd reg1, reg2	satadd imm5, reg2	or reg1, reg2	xor reg1, reg2
and reg1, reg2	tst reg1, reg2	subr reg1, reg2	sub reg1, reg2
add reg1, reg2	add imm5, reg2	cmp reg1, reg2	cmp imm5, reg2
mulh reg1, reg2	shr imm5, reg2	sar imm5, reg2	shl imm5, reg2

<Example>

<i> ld.w [r11], r10
•
•
•
<ii> mov r10, r28
<iii> sld.w0x28, r10

If the decode operation of the mov instruction <ii> immediately before the sld instruction <iii> and an interrupt request conflict before execution of the ld instruction <i> is complete, the execution result of instruction <i> may not be stored in a register.

(b) Countermeasure

<1> When compiler (CA850) is used

Use CA850 Ver. 2.61 or later because generation of the corresponding instruction sequence can be automatically suppressed.

<2> For assembler

When executing the sld instruction immediately after instruction <ii>, avoid the above operation using either of the following methods.

- Insert a nop instruction immediately before the sld instruction.
- Do not use the same register as the sld instruction destination register in the above instruction <ii> executed immediately before the sld instruction.

CHAPTER 4 BUS CONTROL FUNCTION

The V850E/IA2 is provided with an external bus interface function by which external I/O and memories, such as ROM and RAM, can be connected.

4.1 Features

- 16-bit/8-bit data bus sizing function
- Wait function
  - Programmable wait function: up to 7 wait states can be inserted
  - External wait function via $\overline{\text{WAIT}}$ pin
- Idle state insertion function
- External device connection enabled via bus control/port alternate function pins

4.2 Bus Control Pins

The following pins are used for connection to external devices.

Bus Control Pin (Function When in Control Mode)	Function When in Port Mode	Register for Port/Control Mode Switching
Address/data bus (AD0 to AD15)	PDL0 to PDL15 (port DL)	PMCDL
Address bus (A16 to A21)	PDH0 to PDH5 (port DH)	PMCDH
Read/write control ($\overline{\text{LWR}}$/$\overline{\text{UWR}}$, $\overline{\text{RD}}$, ASTB)	PCT0, PCT1, PCT4, PCT6 (port CT)	PMCCT
External wait control ($\overline{\text{WAIT}}$)	PCM0 (port CM)	PMCCM
Internal system clock (CLKOUT)	PCM1 (port CM)	

Remark In the case of ROMless mode, when the system is reset, each bus control pin becomes valid unconditionally.

4.2.1 Pin status during internal ROM, internal RAM, and on-chip peripheral I/O access

When the internal ROM and RAM are accessed, both the address bus and address/data bus become undefined. The external bus control signal becomes inactive.

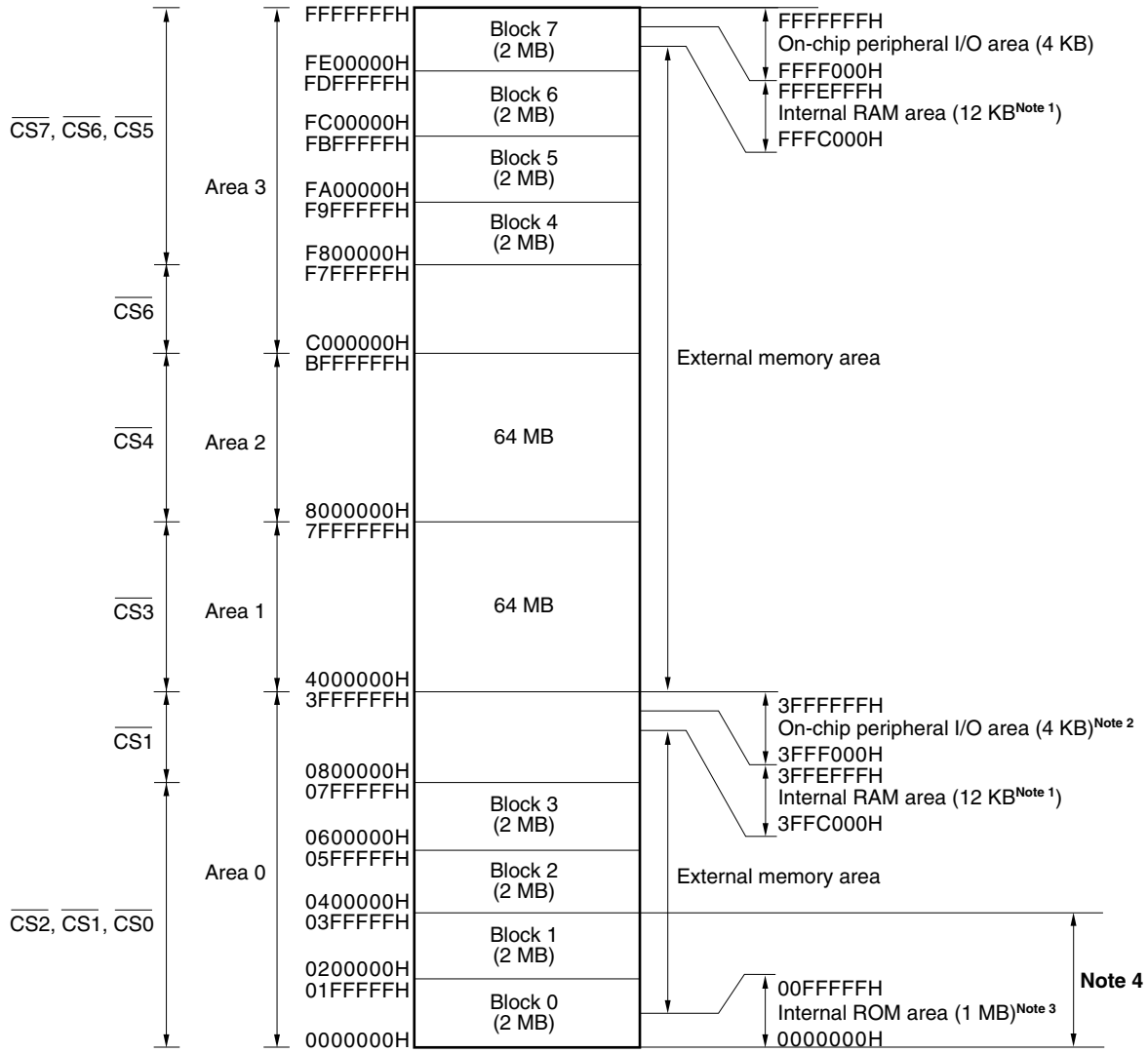
When on-chip peripheral I/O are accessed, both the address bus and address/data bus output the address of the on-chip peripheral I/O currently being accessed. No data is output. The external bus control signal becomes inactive.

4.3 Memory Block Function

In the V850E/IA1, the 256 MB memory space is divided into memory blocks of 2 MB and 64 MB units. The programmable wait function and bus cycle operation mode can be independently controlled for each block.

The area that can be used as program area is the 64 MB space of addresses 0000000H to 3FFFFFFH.

In the V850E/IA2, memory space is the 4 MB space of addresses 000000H to 3FFFFFH (n = 1 to 7) because the CSn pin has been deleted and the A0 to A21 pins have been specified as address pins.



Notes 1. Internal physical RAM: 6 KB

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

3. When in ROMless mode, this becomes an external memory area.

4. Memory space of the V850E/IA2

4.3.1 Chip select control function

Of the 256 MB memory area, the lower 8 MB (0000000H to 07FFFFFH) and the higher 8 MB (F800000H to FFFFFFFH) can be divided into 2 MB memory blocks by chip area selection control registers 0 and 1 (CSC0, CSC1) to control the chip select signal.

The memory area can be effectively used by dividing it into memory blocks using the chip select control function. The priority order is described below.

(1) Chip area selection control registers 0, 1 (CSC0, CSC1)

These registers can be read/written in 16-bit units and become valid by setting each bit to 1.

Only the CS01 and CS00 bits of the CSC0 register are valid in the V850E/IA2. These registers are not affected by other bit settings. In the V850E/IA2, set the CS01 and CS00 bits to 11B so that $\overline{CS0}$ is output to both block 0 and 1.

If different chip select signal outputs are set to the same block, the priority order is controlled as follows.

CSC0: $\overline{CS0}$ > $\overline{CS2}$ > $\overline{CS1}$
CSC1: $\overline{CS7}$ > $\overline{CS5}$ > $\overline{CS6}$

If both the CS0m and CS2m bits of the CSC0 register are set to 0, $\overline{CS1}$ is output to the corresponding block (m = 0 to 3).

Similarly, if both the CS5m and CS7m bits of the CSC1 register are set to 0, $\overline{CS6}$ is output to the corresponding block (m = 0 to 3).

Caution Write to the CSC0 and CSC1 registers after reset, and then do not change the set values.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CSC0	CS33	CS32	CS31	CS30	CS23	CS22	CS21	CS20	CS13	CS12	CS11	CS10	CS03	CS02	CS01	CS00	FFFFF060H	2C11H
CSC1	CS43	CS42	CS41	CS40	CS53	CS52	CS51	CS50	CS63	CS62	CS61	CS60	CS73	CS72	CS71	CS70	FFFFF062H	2C11H

Bit position	Bit name	Function
15 to 0	CSnm (n = 0 to 7) (m = 0 to 3)	Chip select enabled by setting CSnm bit to 1.

CSnm	CS operation
CS00	$\overline{CS0}$ output during block 0 access
CS01	$\overline{CS0}$ output during block 1 access.
CS02	$\overline{CS0}$ output during block 2 access.
CS03	$\overline{CS0}$ output during block 3 access.
CS10 to CS13	**Note 1**
CS20	$\overline{CS2}$ output during block 0 access.
CS21	$\overline{CS2}$ output during block 1 access.
CS22	$\overline{CS2}$ output during block 2 access.
CS23	$\overline{CS2}$ output during block 3 access.
CS30 to CS33	**Note 2**
CS40 to CS43	**Note 3**
CS50	$\overline{CS5}$ output during block 7 access.
CS51	$\overline{CS5}$ output during block 6 access.
CS52	$\overline{CS5}$ output during block 5 access.
CS53	$\overline{CS5}$ output during block 4 access.
CS60 to CS63	**Note 4**
CS70	$\overline{CS7}$ output during block 7 access.
CS71	$\overline{CS7}$ output during block 6 access.
CS72	$\overline{CS7}$ output during block 5 access.
CS73	$\overline{CS7}$ output during block 4 access.

Notes 1. If both the CS0m and CS2m bits have been set to 0, if area 0 is accessed, $\overline{CS1}$ will be output regardless of the setting of the CS1m bit.

2. When area 1 is accessed, $\overline{CS3}$ will be output regardless of the setting of the CS3m bit.

3. When area 2 is accessed, $\overline{CS4}$ will be output regardless of the setting of the CS4m bit.

4. If both the CS5m and CS7m bits have been set to 0, if area 3 is accessed, $\overline{CS6}$ will be output regardless of the setting of the CS6m bit.

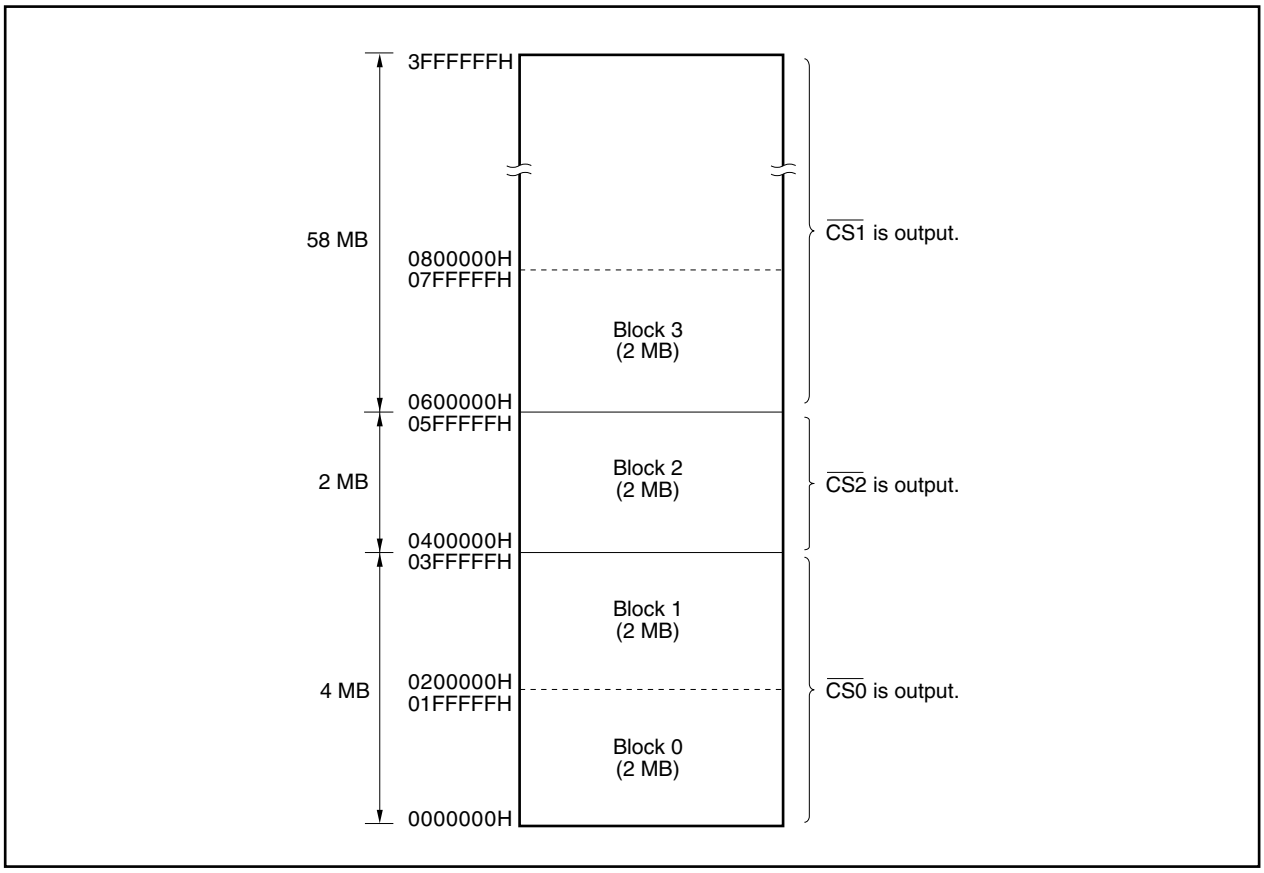
Caution In the V850E/IA2, set the CS01 and CS00 bits to 11B so that $\overline{CS0}$ is output to both block 0 and 1.

The following diagram shows the $\overline{CS}$ signal that is enabled for area 0 when the CSC0 register is set to 0703H.

When the CSC0 register is set to 0703H, $\overline{CS0}$ and $\overline{CS2}$ are output to block 0 and block 1, but since $\overline{CS0}$ has priority over $\overline{CS2}$, $\overline{CS0}$ is output if the addresses of block 0 and block 1 are accessed.

If the address of block 3 is accessed, both the CS03 and CS23 bits of the CSC0 register are 0, and $\overline{CS1}$ is output.

Figure 4-1. Example When CSC0 Register Is Set to 0703H



4.4 Bus Cycle Type Control Function

In the V850E/IA2, the following external devices can be connected directly to each memory block.

- SRAM, external ROM, external I/O

Connected external devices are specified by bus cycle type configuration registers 0 and 1 (BCT0 and BCT1).

(1) Bus cycle type configuration registers 0, 1 (BCT0, BCT1)

These registers can be read/written in 16-bit units.

Only the ME0 bit is valid in the V850E/IA2. These registers are not affected by other bit settings.

Caution Write to the BCT0 and BCT1 registers after reset, and then do not change the set values. Also, do not access an external memory area other than the one for this initialization routine until the initial setting of the BCT0 and BCT1 registers is complete. However, it is possible to access external memory areas whose initial settings are complete.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
BCT0	ME3	1	0	0	ME2	1	0	0	ME1	1	0	0	ME0	1	0	0	FFFFF480H	CCCCH
$\overline{\text{CSn}}$ signal	$\overline{\text{CS3}}$				$\overline{\text{CS2}}$				$\overline{\text{CS1}}$				$\overline{\text{CS0}}$					
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
BCT1	ME7	1	0	0	ME6	1	0	0	ME5	1	0	0	ME4	1	0	0	FFFFF482H	CCCCH
$\overline{\text{CSn}}$ signal	$\overline{\text{CS7}}$				$\overline{\text{CS6}}$				$\overline{\text{CS5}}$				$\overline{\text{CS4}}$					

Bit position	Bit name	Function
15, 11, 7, 3 (BCT0), 15, 11, 7, 3 (BCT1)	MEn (n = 0 to 7)	Sets memory controller operation enable for each chip select.

MEn	Memory controller operation enable
0	Operation disabled
1	Operation enabled

4.5 Bus Access

4.5.1 Number of access clocks

The number of basic clocks required to access each resource is shown below.

Bus Cycle Status / Resource (Bus Width)	Instruction Fetch	Operand Data Access
Internal ROM (32 bits)	1[Note 1]	5
Internal RAM (32 bits)	1[Note 2]	1
On-chip peripheral I/O (16 bits)	–	5[Note 3]
External memory (16 bits)	3[Note 3]	3[Note 3]

Notes 1. This value is 2 in the case of instruction branch.

2. This value is 2 if there is conflict with data access.

3. MIN. value

Remark Unit: Clock/access

4.5.2 Bus sizing function

The bus sizing function controls the data bus width for each CS space. The data bus width is specified by using the bus size configuration register (BSC).

(1) Bus size configuration register (BSC)

This register can be read/written in 16-bit units.

Only the BS00 bit is valid in the V850E/IA2. This register is not affected by other bit settings.

Cautions 1. Write to the BSC register after reset, and then do not change the set values. Also, do not access an external memory area other than the one for this initialization routine until the initial setting of the BSC register is complete. However, it is possible to access external memory areas whose initial settings are complete.

2. When the data bus width is specified as 8 bits, only the signals shown below become active.

$\overline{\text{LWR}}$: When accessing SRAM, external ROM, or external I/O (write cycle)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
BSC	0	BS70	0	BS60	0	BS50	0	BS40	0	BS30	0	BS20	0	BS10	0	BS00	FFFFF066H	5555H
$\overline{\text{CSn}}$ signal		$\overline{\text{CS7}}$		$\overline{\text{CS6}}$		$\overline{\text{CS5}}$		$\overline{\text{CS4}}$		$\overline{\text{CS3}}$		$\overline{\text{CS2}}$		$\overline{\text{CS1}}$		$\overline{\text{CS0}}$		

Bit position	Bit name	Function
14, 12, 10, 8, 6, 4, 2, 0	BSn0 (n = 0 to 7)	Sets the data bus width of CSn space.

BSn0	Data bus width of CSn space
0	8 bits
1	16 bits

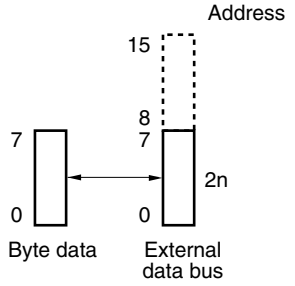
4.5.3 Bus width

The V850E/IA2 accesses on-chip peripheral I/O and external memory in 8-bit, 16-bit, or 32-bit units. The following shows the operation for each type of access. Access all data in order starting from the lower side.

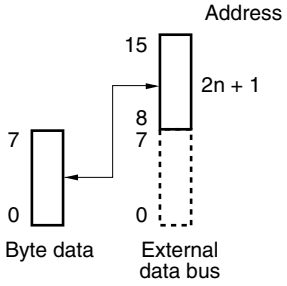
(1) Byte access (8 bits)

(a) When the data bus width is 16 bits (little endian)

<1> Access to even address (2n)

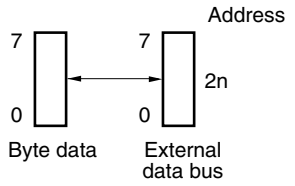


<2> Access to odd address (2n + 1)

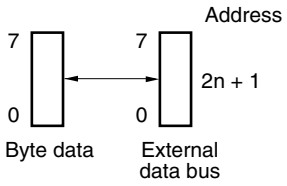


(b) When the data bus width is 8 bits (little endian)

<1> Access to even address (2n)

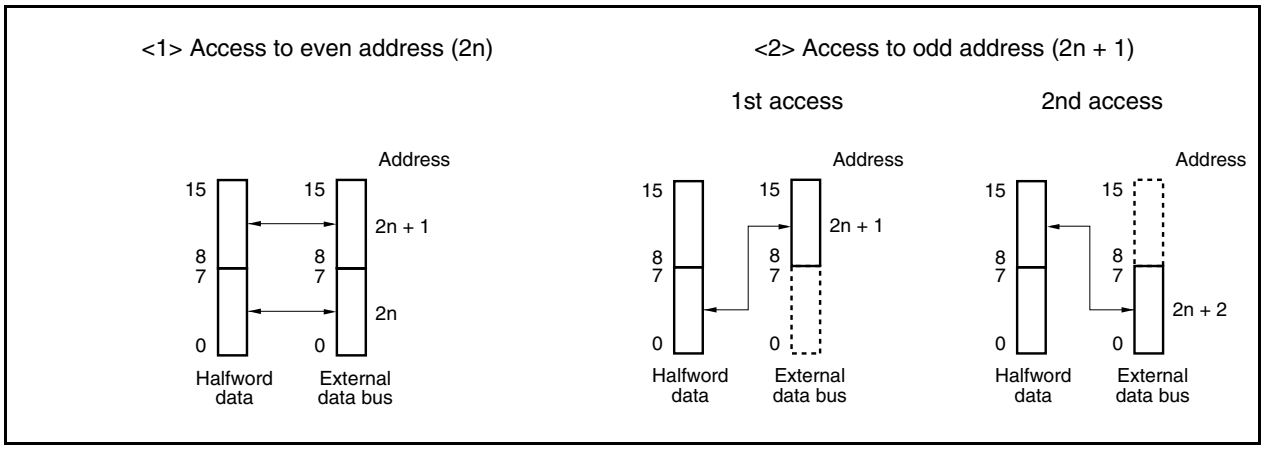


<2> Access to odd address (2n + 1)

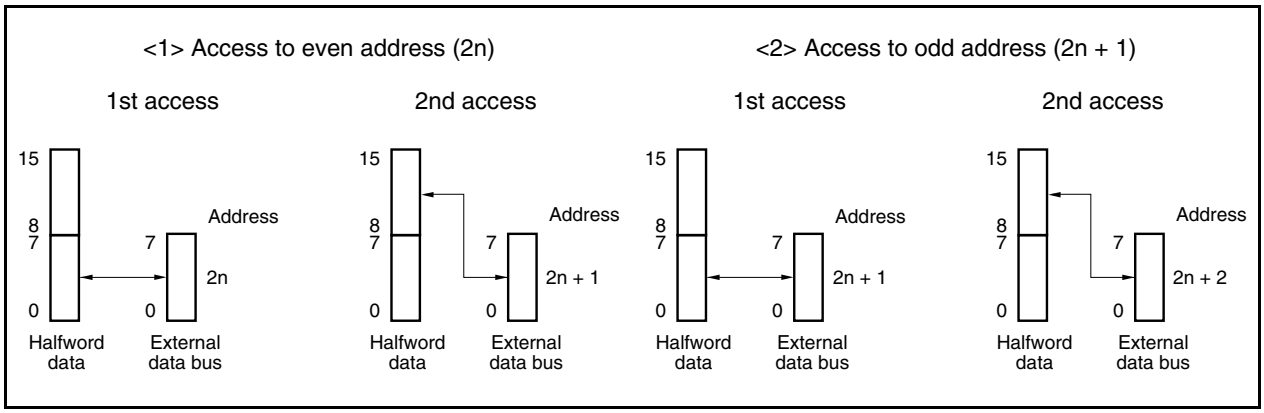


(2) Halfword access (16 bits)

(a) When the bus width is 16 bits (little endian)



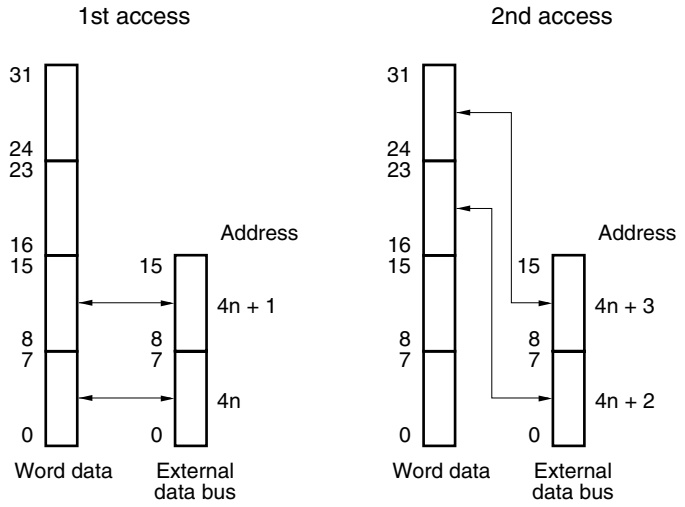
(b) When the data bus width is 8 bits (little endian)



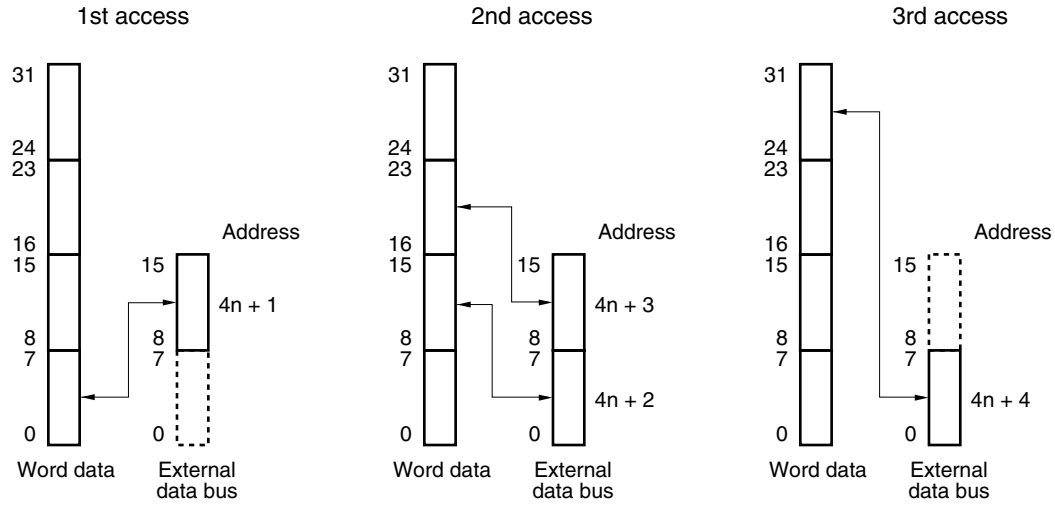
(3) Word access (32 bits)

(a) When the bus width is 16 bits (little endian) (1/2)

<1> Access to address 4n

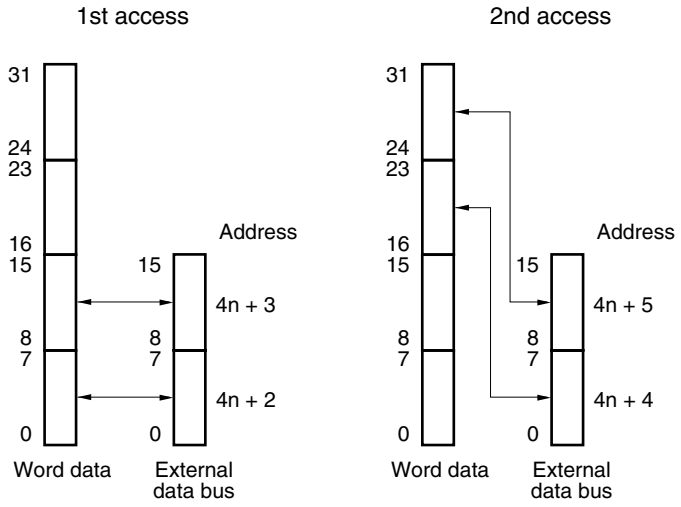


<2> Access to address 4n + 1

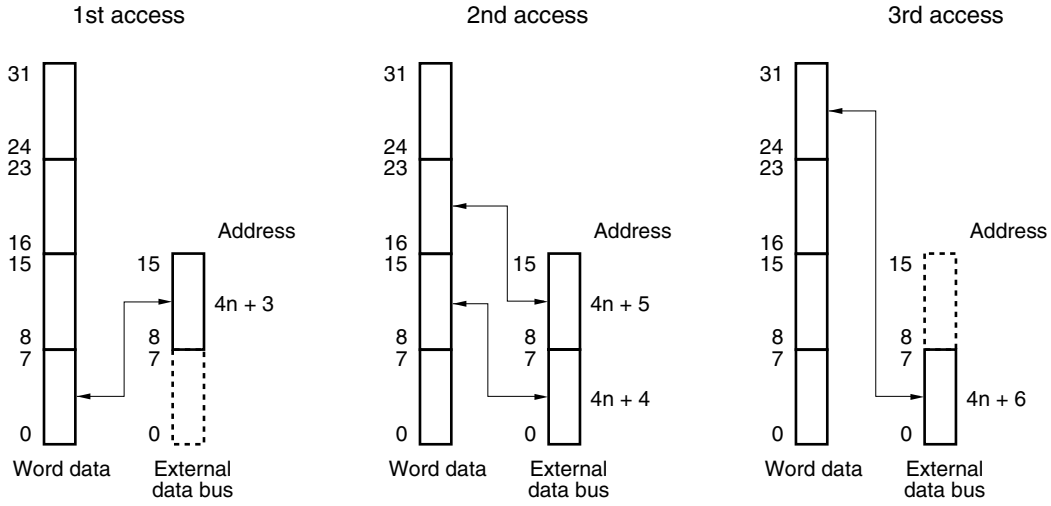


(a) When the bus width is 16 bits (little endian) (2/2)

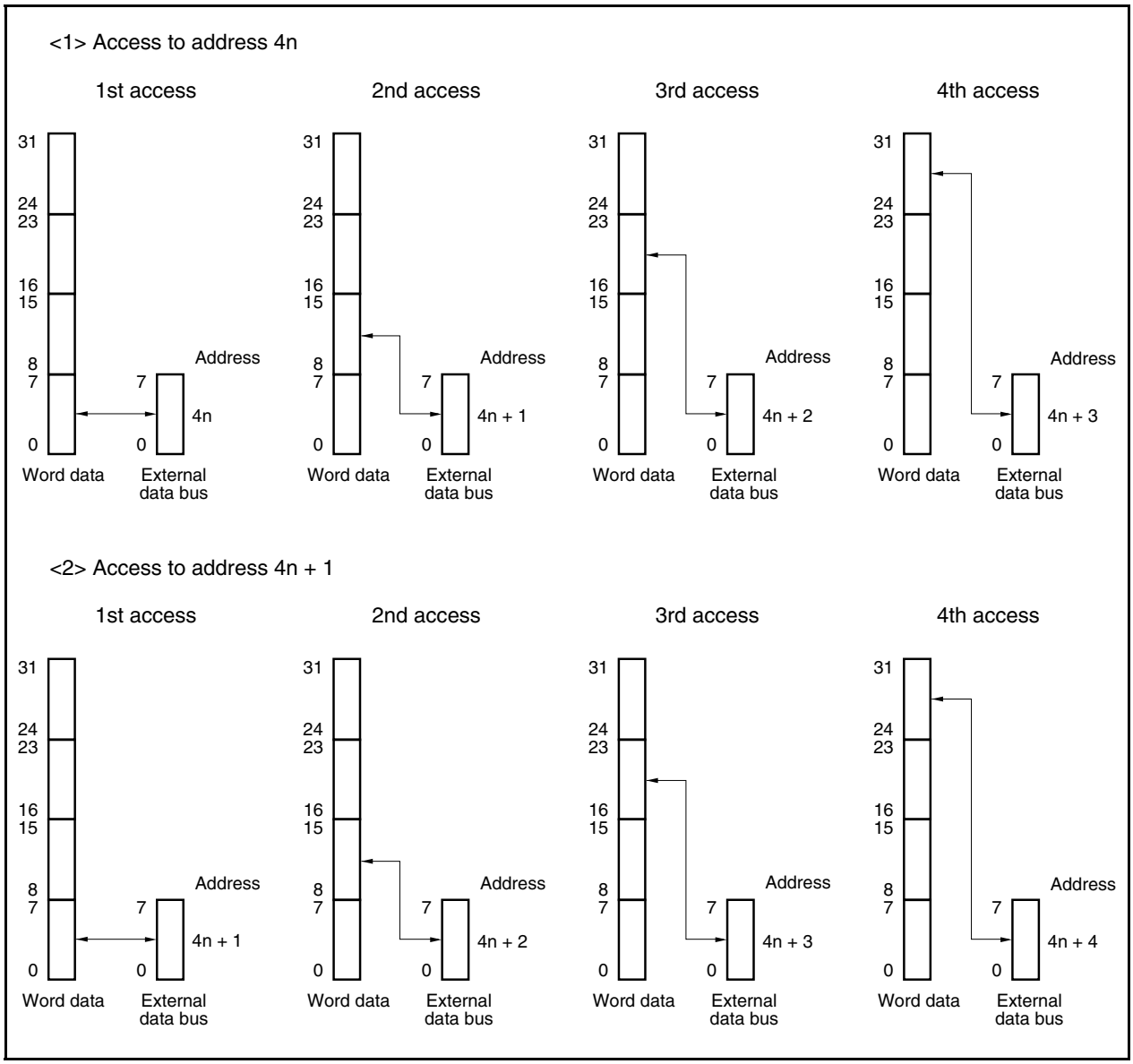
<3> Access to address 4n + 2



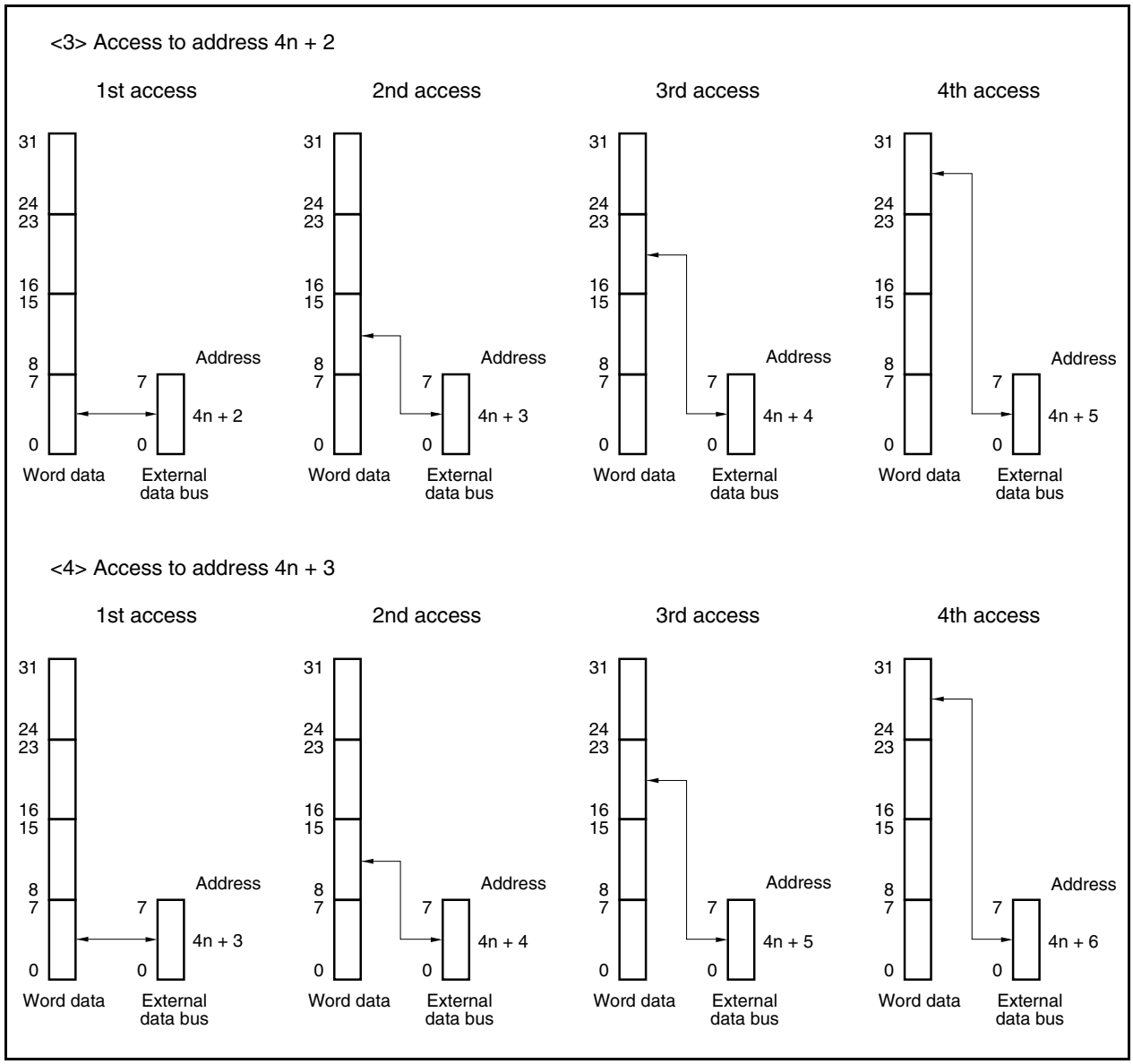
<4> Access to address 4n + 3



(b) When the data bus width is 8 bits (little endian) (1/2)



(b) When the data bus width is 8 bits (little endian) (2/2)



4.6 Wait Function

4.6.1 Programmable wait function

(1) Data wait control registers 0, 1 (DWC0, DWC1)

To facilitate interfacing with low-speed memory or with I/Os, it is possible to insert up to 7 data wait states in the bus cycle activated for each CS space.

The number of wait states can be specified by program using data wait control registers 0 and 1 (DWC0 and DWC1). Just after system reset, all blocks have 3 data wait states inserted.

These registers can be read/written in 16-bit units.

Only the DW02, DW01, and DW00 bits are valid in the V850E/IA2. These registers are not affected by other bit settings.

Cautions 1. The internal ROM area and internal RAM area are not subject to programmable waits and ordinarily no wait access is carried out. The on-chip peripheral I/O area is also not subject to programmable wait states, with wait control performed by each peripheral function only.

2. Write to the DWC0 and DWC1 registers after reset, and then do not change the set values. Also, do not access an external memory area other than the one for this initialization routine until the initial setting of the DWC0 and DWC1 registers is complete. However, it is possible to access external memory areas whose initial settings are complete.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DWC0	0	DW32	DW31	DW30	0	DW22	DW21	DW20	0	DW12	DW11	DW10	0	DW02	DW01	DW00	FFFFF484H	3333H
$\overline{CS}$n signal		$\overline{CS3}$				$\overline{CS2}$				$\overline{CS1}$				$\overline{CS0}$				

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DWC1	0	DW72	DW71	DW70	0	DW62	DW61	DW60	0	DW52	DW51	DW50	0	DW42	DW41	DW40	FFFFF486H	3333H
$\overline{CS}$n signal		$\overline{CS7}$				$\overline{CS6}$				$\overline{CS5}$				$\overline{CS4}$				

Bit position	Bit name	Function
14 to 12, 10 to 8, 6 to 4, 2 to 0	DWn2 to DWn0 (n = 0 to 7)	Specifies the number of wait states inserted in the CSn space.

DWn2	DWn1	DWn0	Number of wait states inserted in CSn space
0	0	0	Not inserted
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

(2) Address wait control register (AWC)

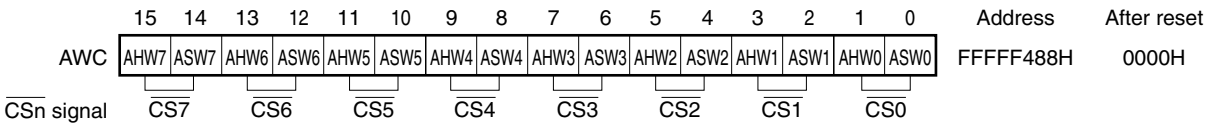
In the V850E/IA2, address setup wait and address hold wait states can be inserted before and after the T1 cycle, respectively.

These wait states can be set for each CS space via the AWC register.

This register can be read/written in 16-bit units.

Only the AHW0 and ASW0 bits are valid in the V850E/IA2. This register is not affected by other bit settings.

Caution Write to the AWC register after reset, and then do not change the set values.



Bit position	Bit name	Function
15, 13, 11, 9, 7, 5, 3, 1	AHWn (n = 0 to 7)	Sets the insertion of an address hold wait state in each CSn space after the T1 cycle. 0: Address hold wait state not inserted 1: Address hold wait state inserted
14, 12, 10, 8, 6, 4, 2, 0	ASWn (n = 0 to 7)	Sets the insertion of an address setup wait state in each CSn space before the T1 cycle. 0: Address setup wait state not inserted 1: Address setup wait state inserted

4.6.2 External wait function

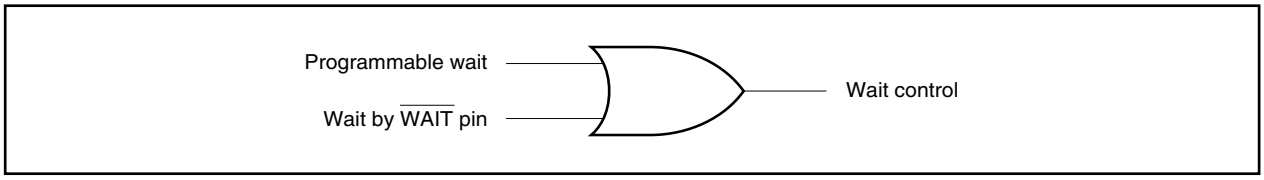
When an extremely slow device, an I/O, or an asynchronous system is connected, an arbitrary number of wait states can be inserted in the bus cycle by the external wait pin ($\overline{\text{WAIT}}$) for synchronization with the external device.

Just as with programmable waits, accessing internal ROM, internal RAM, and on-chip peripheral I/O areas cannot be controlled by external waits.

The external $\overline{\text{WAIT}}$ signal can be input asynchronously to CLKOUT and is sampled at the falling edge of the CLKOUT signal in the T2 and TW states of the bus cycle. If the setup/hold time is not satisfied within the sampling timing, a wait state may or may not be inserted in the next state.

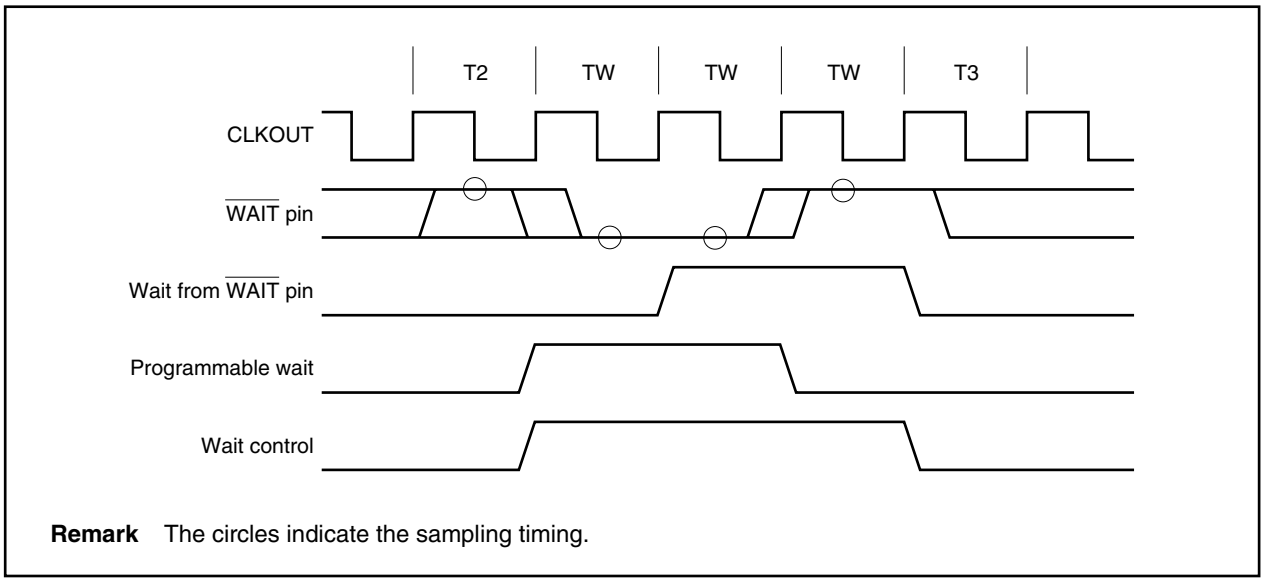
4.6.3 Relationship between programmable wait and external wait

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

Figure 4-2. Example of Wait Insertion



Remark The circles indicate the sampling timing.

4.7 Idle State Insertion Function

To facilitate interfacing with low-speed memory devices, a set number of idle states (T1) can be inserted into the bus cycle to be activated after the T3 state to secure the data output float delay time (t_{DF}) of the memory when each CS space is read-accessed. The bus cycle following the T3 state starts after the inserted idle state(s).

Idle states are inserted at the following timing.

- After the read cycle for SRAM, external I/O, or external ROM.

The idle state insertion setting can be specified using the bus cycle control register (BCC). Idle state insertion is automatically programmed for all memory blocks immediately after a system reset.

(1) Bus cycle control register (BCC)

This register can be read/written in 16-bit units.

Only the BC01 bit is valid in the V850E/IA2. This register is not affected by other bit settings.

Cautions 1. Idle states cannot be inserted in internal ROM, internal RAM, or on-chip peripheral I/O areas.

2. Write to the BCC register after reset, and then do not change the set values. Also, do not access an external memory area other than the one for this initialization routine until the initial setting for this register is complete. However, it is possible to access external memory areas whose initial settings are complete.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
BCC	BC71	0	BC61	0	BC51	0	BC41	0	BC31	0	BC21	0	BC11	0	BC01	0	FFFFF48AH	AAAAH
$\overline{\text{CS}}$n signal	$\overline{\text{CS7}}$		$\overline{\text{CS6}}$		$\overline{\text{CS5}}$		$\overline{\text{CS4}}$		$\overline{\text{CS3}}$		$\overline{\text{CS2}}$		$\overline{\text{CS1}}$		$\overline{\text{CS0}}$			

Bit position	Bit name	Function
15, 13, 11, 9, 7, 5, 3, 1	BCn1 (n = 0 to 7)	Specifies the insertion of idle states after the T3 state in each CSn space. 0: Idle state not inserted 1: Idle state inserted

4.8 Bus Priority Order

There are three external bus cycles: DMA cycle, operand data access, and instruction fetch.

In order of priority, DMA cycle is the highest, followed by operand data access and instruction fetch, in that order.

An instruction fetch may be inserted between a read access and write access during a read modify write access. Also, an instruction fetch may be inserted between bus accesses when the CPU bus is locked.

Table 4-1. Bus Priority Order

Priority Order	External Bus Cycle	Bus Master
High ↕ Low	DMA cycle	DMA controller
	Operand data access	CPU
	Instruction fetch	CPU

4.9 Boundary Operation Conditions

4.9.1 Program space

(1) Branching to the on-chip peripheral I/O area or successive fetches from the internal RAM area to the on-chip peripheral I/O area are prohibited. If the above is performed (branching or successive fetch), the data to be fetched is undefined and the operation is not guaranteed.

(2) If a branch instruction exists at the upper limit of the internal RAM area, a prefetch operation (invalid fetch) that straddles over the on-chip peripheral I/O area does not occur.

4.9.2 Data space

The V850E/IA2 is provided with an address misalign function.

Through this function, regardless of the data format (word data, halfword data, or byte data), data can be allocated to all addresses. However, in the case of word data and halfword data, if the data is not subject to boundary alignment, the bus cycle will be generated at least 2 times and bus efficiency will drop.

(1) In the case of halfword-length data access

When the address's LSB is 1, the byte-length bus cycle will be generated 2 times.

(2) In the case of word-length data access

(a) When the address's LSB is 1, bus cycles will be generated in the order of byte-length bus cycle, halfword-length bus cycle, and byte-length bus cycle.

(b) When the address's lowest 2 bits are 10, the halfword-length bus cycle will be generated 2 times.

CHAPTER 5 MEMORY ACCESS CONTROL FUNCTION

5.1 SRAM, External ROM, External I/O Interface

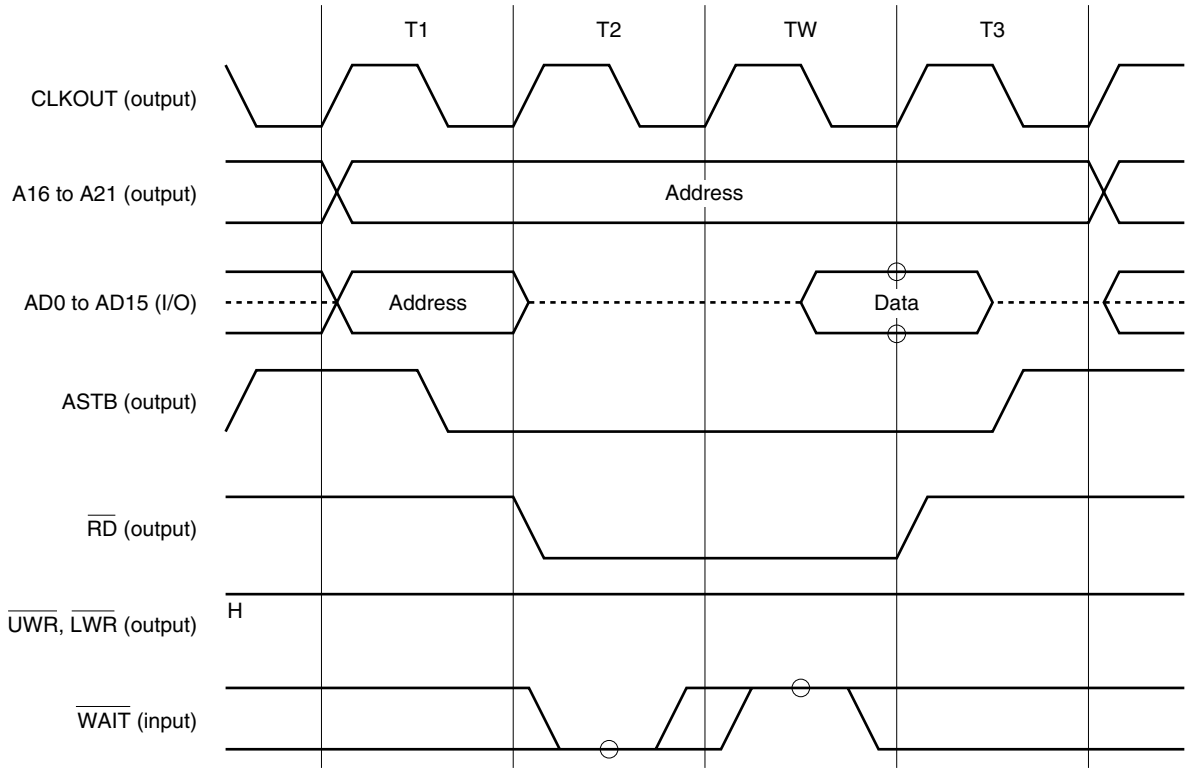
5.1.1 Features

- SRAM is accessed in a minimum of 3 states.
- A maximum of 7 programmable data wait states can be inserted according to DWC0 and DWC1 register settings.
- Data waits can be controlled by $\overline{\text{WAIT}}$ pin input.
- An idle state (1 state) can be inserted after a read/write cycle by setting the BCC register.
- An address hold wait state or address setup wait state can be inserted by setting the AWC register.

5.1.2 SRAM, external ROM, external I/O access

Figure 5-1. SRAM, External ROM, External I/O Access Timing (1/4)

(a) When reading (1 wait inserted)

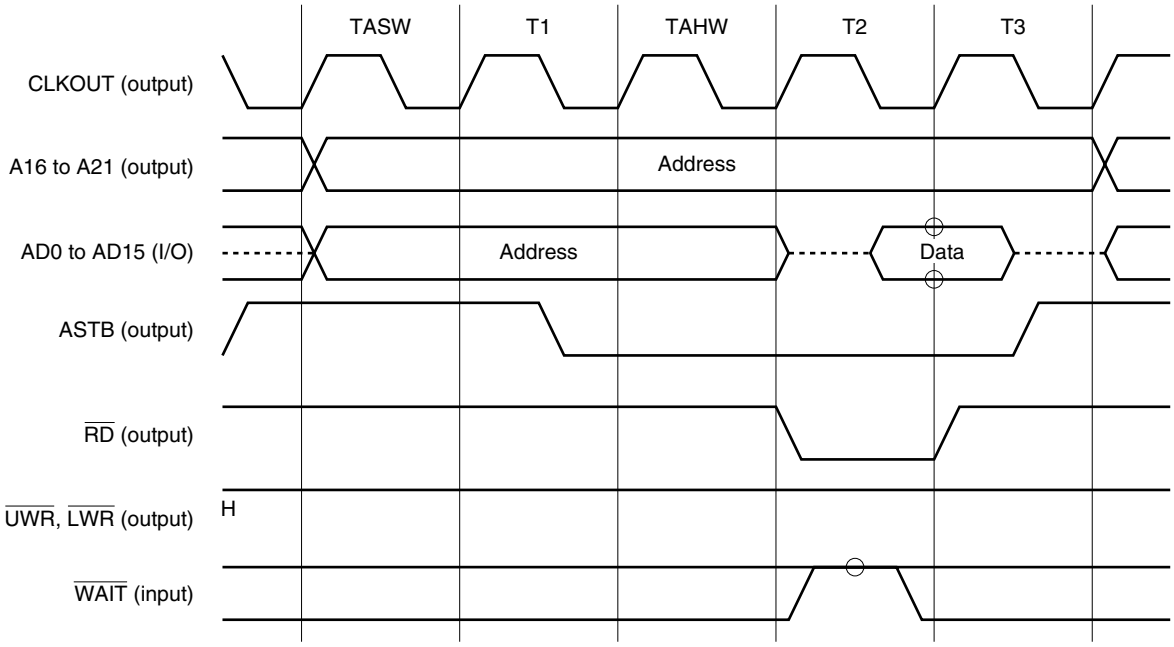


Remarks 1. The circles indicate the sampling timing.

2. Broken lines indicate high impedance.

Figure 5-1. SRAM, External ROM, External I/O Access Timing (2/4)

(b) When reading (0 waits, address setup waits, address hold wait states inserted)

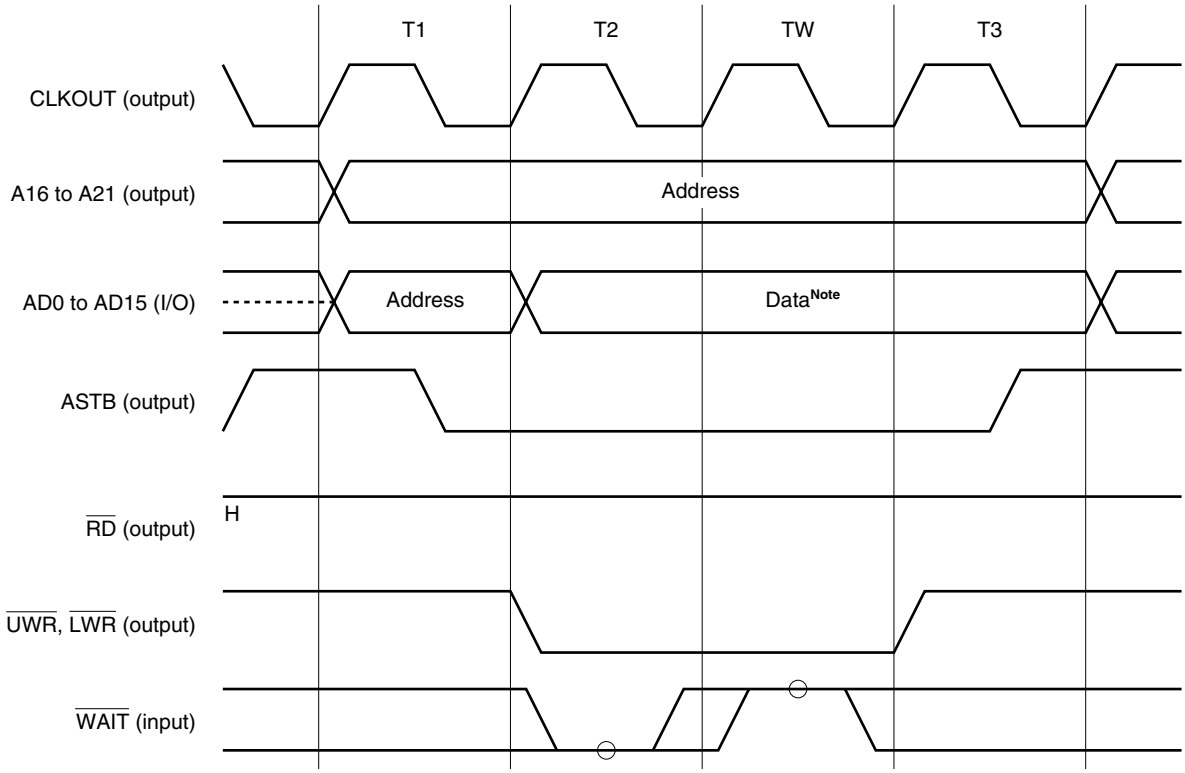


Remarks 1. The circles indicate the sampling timing.

2. Broken lines indicate high impedance.

Figure 5-1. SRAM, External ROM, External I/O Access Timing (3/4)

(c) When writing (1 wait inserted)

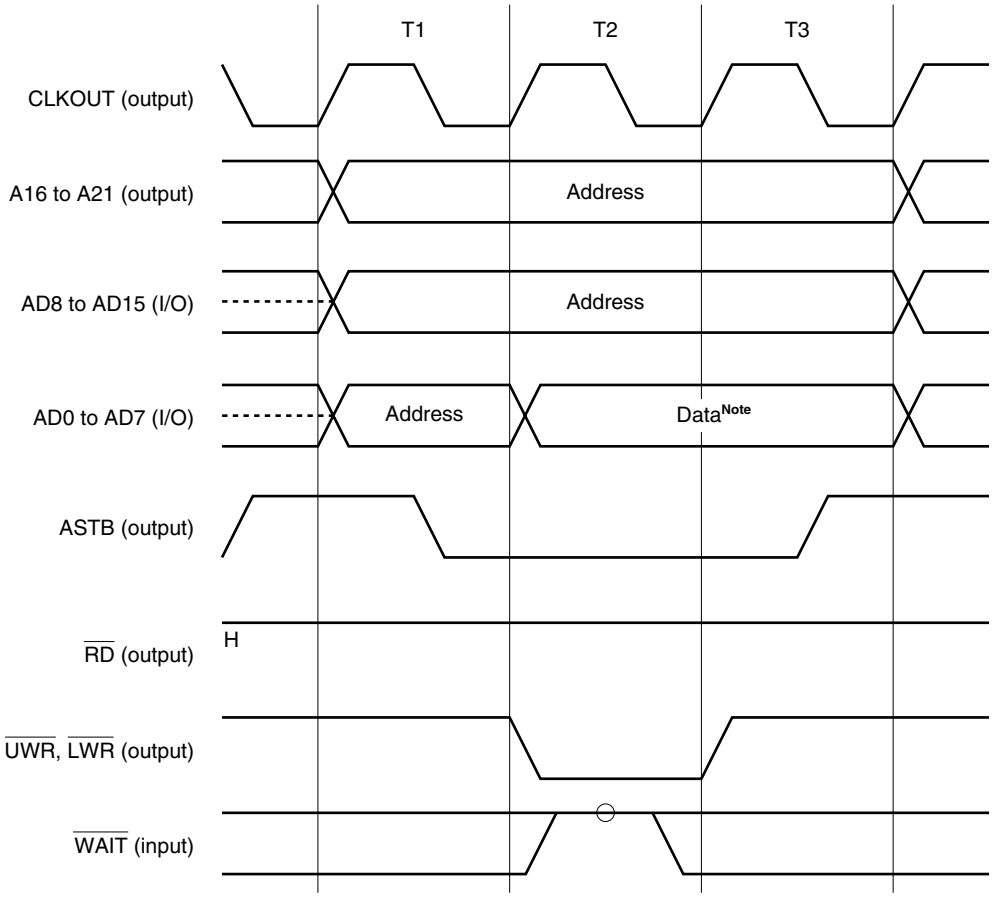


Note AD0 to AD7 output invalid data when odd-numbered address byte data is accessed.
AD8 to AD15 output invalid data when even-numbered address byte data is accessed.

Remarks 1. The circles indicate the sampling timing.
2. Broken lines indicate high impedance.

Figure 5-1. SRAM, External ROM, External I/O Access Timing (4/4)

(d) When writing (0 waits inserted, for 8-bit data bus)



Note AD0 to AD7 output invalid data when odd-numbered address byte data is accessed.

Remarks 1. The circles indicate the sampling timing.

2. Broken lines indicate high impedance.

CHAPTER 6 DMA FUNCTIONS (DMA CONTROLLER)

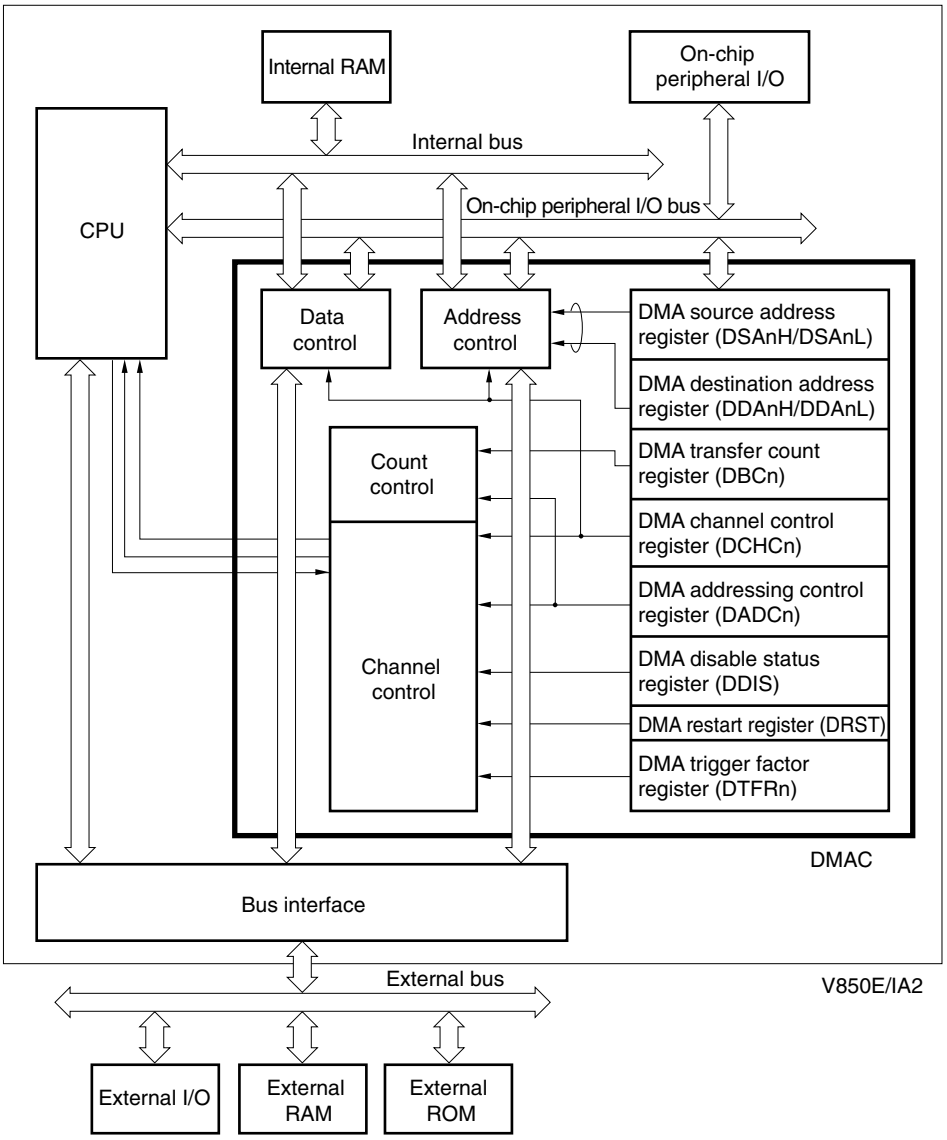
The V850E/IA2 includes a direct memory access (DMA) controller (DMAC) that executes and controls DMA transfer.

The DMAC controls data transfer between memory and peripheral I/O, between memories or between peripheral I/Os, based on DMA requests issued by the on-chip peripheral I/O (serial interface, timer/counter, and A/D converter), or software triggers (memory refers to internal RAM or external memory).

6.1 Features

- Four independent DMA channels
- Transfer unit: 8/16 bits
- Maximum transfer count: 65,536 (2^{16})
- Two-cycle transfer
- Three transfer modes
  - Single transfer mode
  - Single-step transfer mode
  - Block transfer mode
- Transfer requests
  - Request by interrupts from on-chip peripheral I/O (serial interface, timer/counter, A/D converter)
  - Requests by software trigger
- Transfer targets
  - Memory ↔ peripheral I/O
  - Memory ↔ memory
  - Peripheral I/O ↔ peripheral I/O
- Next address setting function

6.2 Configuration



Remark n = 0 to 3

6.3 Control Registers

6.3.1 DMA source address registers 0 to 3 (DSA0 to DSA3)

These registers are used to set the DMA source addresses (28 bits each) for DMA channel n (n = 0 to 3). They are divided into two 16-bit registers, DSAnH and DSAnL.

Since these registers are configured as 2-stage FIFO buffer registers, a new source address for DMA transfer can be specified during DMA transfer. (Refer to **6.8 Next Address Setting Function**.) In this case, if a new DSAn register is set, the value set will be transferred to the slave register and enabled only if DMA transfer ends normally, and the TCn bit of DMA channel control register n (DCHCn) has been set to 1 or the INITn bit of the DCHCn register has been set to 1 (n = 0 to 3).

(1) DMA source address registers 0H to 3H (DSA0H to DSA3H)

These registers can be read/written in 16-bit units.

Be sure to set bits 14 to 12 to 0. If they are set to 1, the operation is not guaranteed.

Cautions 1. When setting an address of an on-chip peripheral I/O register for the source address, be sure to specify an address between FFFF000H and FFFFFFFH. An address of the on-chip peripheral I/O register image (3FFF000H to 3FFFFFFH) must not be specified.

2. Do not set the DSAnH register while DMA is suspended.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DSA0H	IR	0	0	0	SA27	SA26	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16	FFFFF082H	Undefined
DSA1H	IR	0	0	0	SA27	SA26	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16	FFFFF08AH	Undefined
DSA2H	IR	0	0	0	SA27	SA26	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16	FFFFF092H	Undefined
DSA3H	IR	0	0	0	SA27	SA26	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16	FFFFF09AH	Undefined

Bit position	Bit name	Function
15	IR	Specifies the DMA source address. 0: External memory, on-chip peripheral I/O 1: Internal RAM
11 to 0	SA27 to SA16	Sets the DMA source addresses (A27 to A16). During DMA transfer, it stores the next DMA transfer source address.

(2) DMA source address registers 0L to 3L (DSA0L to DSA3L)

These registers can be read/written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DSA0L	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0	FFFFF080H	Undefined
DSA1L	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0	FFFFF088H	Undefined
DSA2L	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0	FFFFF090H	Undefined
DSA3L	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0	FFFFF098H	Undefined

Bit position	Bit name	Function
15 to 0	SA15 to SA0	Sets the DMA source address (A15 to A0). During DMA transfer, it stores the next DMA transfer source address.

6.3.2 DMA destination address registers 0 to 3 (DDA0 to DDA3)

These registers are used to set the DMA destination address (28 bits each) for DMA channel n (n = 0 to 3). They are divided into two 16-bit registers, DDAnH and DDAnL.

Since these registers are configured as 2-stage FIFO buffer registers, a new destination address for DMA transfer can be specified during DMA transfer. (Refer to **6.8 Next Address Setting Function**.) In this case, if a new DDAn register is set, the value set will be transferred to the slave register and enabled only if DMA transfer ends normally, and the TCn bit of DMA channel control register n (DCHCn) has been set to 1 or the INITn bit of the DCHCn register has been set to 1 (n = 0 to 3).

(1) DMA destination address registers 0H to 3H (DDA0H to DDA3H)

These registers can be read/written in 16-bit units.

Be sure to set bits 14 to 12 to 0. If they are set to 1, the operation is not guaranteed.

Cautions 1. When setting an address of an on-chip peripheral I/O register for the destination address, be sure to specify an address between FFFF000H and FFFFFFFH. An address of the on-chip peripheral I/O register image (3FFF000H to 3FFFFFFH) must not be specified.

2. Do not set the DDAnH register while DMA is suspended.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
DDA0H	IR	0	0	0	DA27	DA26	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	Address FFFFF086H	After reset Undefined
DDA1H	IR	0	0	0	DA27	DA26	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	Address FFFFF08EH	After reset Undefined
DDA2H	IR	0	0	0	DA27	DA26	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	Address FFFFF096H	After reset Undefined
DDA3H	IR	0	0	0	DA27	DA26	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	Address FFFFF09EH	After reset Undefined

Bit position	Bit name	Function
15	IR	Specifies the DMA destination address. 0: External memory, on-chip peripheral I/O 1: Internal RAM
11 to 0	DA27 to DA16	Sets the DMA destination addresses (A27 to A16). During DMA transfer, it stores the next DMA transfer destination address.

(2) DMA destination address registers 0L to 3L (DDA0L to DDA3L)

These registers can be read/written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
DDA0L	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	Address FFFFF084H	After reset Undefined
DDA1L	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	Address FFFFF08CH	After reset Undefined
DDA2L	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	Address FFFFF094H	After reset Undefined
DDA3L	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	Address FFFFF09CH	After reset Undefined

Bit position	Bit name	Function
15 to 0	DA15 to DA0	Sets the DMA destination address (A15 to A0). During DMA transfer, it stores the next DMA transfer destination address.

6.3.3 DMA transfer count registers 0 to 3 (DBC0 to DBC3)

These 16-bit registers are used to set the byte transfer counts for DMA channels n (n = 0 to 3). They store the remaining transfer counts during DMA transfer.

Since these registers are configured as 2-stage FIFO buffer registers, a new DMA byte transfer count for DMA transfer can be specified during DMA transfer. (Refer to **6.8 Next Address Setting Function**.) In this case, if a new DBCn register is set, the value set will be transferred to the slave register and enabled only if DMA transfer ends normally, and the TCn bit of DMA channel control register n (DCHCn) has been set to 1 or the INITn bit of the DCHCn register has been set to 1 (n = 0 to 3).

These registers are decremented by 1 per transfer. Transfer is terminated if a borrow occurs.

These registers can be read/written in 16-bit units.

Cautions 1. During 2-cycle transfer when the transfer source is the internal RAM, do not set the transfer count to 2 (the set value of the DBCn register is 0001H).
If DMA transfer is required twice, perform DMA transfer with the transfer count set to one (the set value of the DBCn register is 0000H) twice.
2. Do not set the DBCn register while DMA is suspended.

Remark If the DBCn register is read after a terminal count has occurred during DMA transfer without the value of the DBCn register rewritten, the value set immediately before DMA transfer is read (0000H is not read even after completion of transfer).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
DBC0	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	Address FFFFF0C0H	After reset Undefined
DBC1	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	Address FFFFF0C2H	After reset Undefined
DBC2	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	Address FFFFF0C4H	After reset Undefined
DBC3	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	Address FFFFF0C6H	After reset Undefined

Bit position	Bit name	Function
15 to 0	BC15 to BC0	Sets the byte transfer count. It stores the remaining byte transfer count during DMA transfer.

DBCn (n = 0 to 3)	States
0000H	Byte transfer count 1 or remaining byte transfer count
0001H	Byte transfer count 2 or remaining byte transfer count
:	:
FFFFH	Byte transfer count 65,536 (2^{16}) or remaining byte transfer count

6.3.4 DMA addressing control registers 0 to 3 (DADC0 to DADC3)

These 16-bit registers are used to control the DMA transfer modes for DMA channel n (n = 0 to 3). These registers cannot be accessed during DMA operation.

They can be read/written in 16-bit units.

Be sure to set bits 13 to 8, 1, and 0 to 0. If they are set to 1, the operation is not guaranteed.

Cautions 1. The DS1 and DS0 bits are used to set how many bits of data are transferred.
When 8-bit data (DS1, DS0 bits = 00) is set, the lower data bus (AD0 to AD7) is not necessarily used.
When the transfer data size is set to 16 bits, the transfer must start from an address with bit 1 of the lower address aligned to "0". In this case, the transfer cannot start from an odd address.

2. Set the DADCn register when the corresponding channel is in one of the following periods (the operation is not guaranteed if set at another timing).

- **Time from system reset to the generation of the first DMA transfer**
- **Time from DMA transfer end (after terminal count) to the generation of the next DMA transfer request**
- **Time from the forcible termination of DMA transfer (after the INITn bit of DMA channel control register n (DCHCn) has been set to 1) to the generation of the next DMA transfer request**

(1/2)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
DADC0	DS1	DS0	0	0	0	0	0	0	SAD1	SAD0	DAD1	DAD0	TM1	TM0	0	0	Address FFFFF0D0H	After reset 0000H
DADC1	DS1	DS0	0	0	0	0	0	0	SAD1	SAD0	DAD1	DAD0	TM1	TM0	0	0	Address FFFFF0D2H	After reset 0000H
DADC2	DS1	DS0	0	0	0	0	0	0	SAD1	SAD0	DAD1	DAD0	TM1	TM0	0	0	Address FFFFF0D4H	After reset 0000H
DADC3	DS1	DS0	0	0	0	0	0	0	SAD1	SAD0	DAD1	DAD0	TM1	TM0	0	0	Address FFFFF0D6H	After reset 0000H

Bit position	Bit name	Function
15, 14	DS1, DS0	Sets the transfer data size for DMA transfer.

DS1	DS0	Transfer data size
0	0	8 bits
0	1	16 bits
1	0	Setting prohibited
1	1	Setting prohibited

For the on-chip peripheral I/O registers, ensure the transfer size matches the access size.

(2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>SAD1,
SAD0</td><td>Sets the count direction of the source address for DMA channel n (n = 0 to 3).
<table>
<tr><th>SAD1</th><th>SAD0</th><th>Count direction</th></tr>
<tr><td>0</td><td>0</td><td>Increment</td></tr>
<tr><td>0</td><td>1</td><td>Decrement</td></tr>
<tr><td>1</td><td>0</td><td>Fixed</td></tr>
<tr><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>5, 4</td><td>DAD1,
DAD0</td><td>Sets the count direction of the destination address for DMA channel n (n = 0 to 3).
<table>
<tr><th>DAD1</th><th>DAD0</th><th>Count direction</th></tr>
<tr><td>0</td><td>0</td><td>Increment</td></tr>
<tr><td>0</td><td>1</td><td>Decrement</td></tr>
<tr><td>1</td><td>0</td><td>Fixed</td></tr>
<tr><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>3, 2</td><td>TM1, TM0</td><td>Sets the transfer mode during DMA transfer.
<table>
<tr><th>TM1</th><th>TM0</th><th>Transfer mode</th></tr>
<tr><td>0</td><td>0</td><td>Single transfer mode</td></tr>
<tr><td>0</td><td>1</td><td>Single-step transfer mode</td></tr>
<tr><td>1</td><td>0</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>Block transfer mode</td></tr>
</table></td></tr>
</table>

6.3.5 DMA channel control registers 0 to 3 (DCHC0 to DCHC3)

These 8-bit registers are used to control the DMA transfer operating mode for DMA channel n (n = 0 to 3).

These registers can be read/written in 8-bit or 1-bit units. (However, bit 7 is read only and bits 2 and 1 are write only. If bits 2 and 1 are read, the read value is always 0.)

Be sure to set bits 6 to 4 to 0. If they are set to 1, the operation is not guaranteed.

Cautions 1. If transfer is completed with the MLEn bit set to 1, and the next transfer request is executed with the DMA transfer (hardware DMA) started by an interrupt from the on-chip peripheral I/O, the next transfer will be executed if the TCn bit is set to 1 (will not be automatically cleared to 0).

2. Set the MLEn bit when the corresponding channel is in one of the following periods (the operation is not guaranteed if set at another timing).

- **Time from system reset to the generation of the first DMA transfer request**
- **Time from DMA transfer end (after terminal count) to the generation of the next DMA transfer request**
- **Time from the forcible termination of DMA transfer (after the INITn bit has been set to 1) to the generation of the next DMA transfer request**

3. If DMA transfer is forcibly terminated in the last transfer cycle with the MLEn bit set to 1, the same operations as transfer completion (setting of the TCn bit to 1) are performed (the Enn bit will be cleared to 0 in forcible termination regardless of the value of the MLEn bit).
In this case, at the next DMA transfer request, the Enn bit must be set to 1 and the TCn bit must be read (cleared to 0).

4. During DMA transfer completion (terminal count), each bit is updated in the order of clearing the Enn bit to 0 and setting the TCn bit to 1. For this reason, if the TCn bit and Enn bit are in the polling mode, the value indicating "transfer not completed, and transfer prohibited" (TCn bit = 0, and Enn bit = 0) may be read in some cases if the DCHCn register is read while each of the above bits is being updated (this is not an error).

5. Do not set the Enn and STGn bits while DMA is suspended. The operation is not guaranteed if set while DMA is suspended.

	<7>	6	5	4	<3>	<2>	<1>	<0>		
DCHC0	TC0	0	0	0	MLE0	INIT0	STG0	E00	Address FFFFF0E0H	After reset 00H
DCHC1	TC1	0	0	0	MLE1	INIT1	STG1	E11	Address FFFFF0E2H	After reset 00H
DCHC2	TC2	0	0	0	MLE2	INIT2	STG2	E22	Address FFFFF0E4H	After reset 00H
DCHC3	TC3	0	0	0	MLE3	INIT3	STG3	E33	Address FFFFF0E6H	After reset 00H

Bit position	Bit name	Function
7	TCn	This status bit indicates whether DMA transfer through DMA channel n has completed or not. This bit is read-only. It is set to 1 during the last DMA transfer and cleared (to 0) when it is read. 0: DMA transfer had not completed. 1: DMA transfer had completed.
3	MLEn	When this bit is set to 1 when DMA transfer is complete (at terminal count output), the Enn bit is not cleared to 0 and the DMA transfer enable state is retained. When the next DMA transfer start factor is an interrupt from the on-chip peripheral I/O (hardware DMA), the DMA transfer request can be acknowledged even when the TCn bit is not read. When the next DMA transfer start factor is the setting of the STGn bit to 1 (software DMA), the DMA transfer start factor can be acknowledged by reading and clearing the TCn bit to 0. When this bit is cleared to 0 when DMA transfer is complete (at terminal count output), the Enn bit is cleared to 0 and the DMA transfer disable state is entered. At the next DMA transfer request, the setting of the Enn bit to 1 and the reading of the TCn bit are required.
2	INITn	When this bit is set to 1 during DMA transfer or while DMA is suspended, DMA transfer is forcibly terminated (refer to **6.12.1 Restrictions on forcible termination of DMA transfer**).
1	STGn	If this bit is set to 1 in the DMA transfer enable state (TCn bit = 0, Enn bit = 1), DMA transfer is started.
0	Enn	Specifies whether DMA transfer through DMA channel n is to be enabled or disabled. This bit is cleared to 0 when DMA transfer ends. It is also cleared to 0 when DMA transfer is forcibly suspended or terminated by means of setting the INITn bit to 1 or by NMI input. 0: DMA transfer disabled 1: DMA transfer enabled **Caution Once the Enn bit is set to 1, do not set the bit again until the number of DMA transfers set in the DBCn register is complete or DMA transfer has been forcibly terminated by setting the INITn bit.**

Remark n = 0 to 3

6.3.6 DMA disable status register (DDIS)

This register holds the contents of the Enn bit of the DCHCn register when DMA is forcibly suspended (during NMI input) (n = 0 to 3).

This register is read-only, in 8-bit units.

Be sure to set bits 7 to 4 to 0. If they are set to 1, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
DDIS	0	0	0	0	CH3	CH2	CH1	CH0	FFFFF0F0H	00H

Bit position	Bit name	Function
3 to 0	CH3 to CH0	Reflects the value of the Enn bit of the DCHCn register when DMA is forcibly suspended (during NMI input). The contents of this register are held until the next forcible suspension (NMI input) or until the system is reset.

6.3.7 DMA restart register (DRST)

★ The ENn bit of the DRST register and the Enn bit of the DCHCn register are linked to each other, the Enn bit can also be used to set the enabling or disabling of DMA transfer independently for four channels, and the DRST register can be used to set the enabling or disabling of DMA transfer for four channels at the same time (n = 0 to 3).

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

Be sure to set bits 7 to 4 to 0. If they are set to 1, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
DRST	0	0	0	0	EN3	EN2	EN1	EN0	FFFFF0F2H	00H

Bit position	Bit name	Function
3 to 0	EN3 to EN0	Specifies whether DMA transfer via DMA channel n is to be enabled or disabled. This bit is cleared to 0 when DMA transfer is completed in accordance with the terminal count output (n = 0 to 3). It is also cleared to 0 when DMA transfer is forcibly terminated by setting the INITn bit of the DCHCn register to 1 or by NMI input. 0: DMA transfer disabled 1: DMA transfer enabled

6.3.8 DMA trigger factor registers 0 to 3 (DTFR0 to DTFR3)

These 8-bit registers are used to control the DMA transfer start trigger via interrupt requests from on-chip peripheral I/O.

The interrupt requests set with these registers serve as DMA transfer start factors.

★ These registers can be read/written in 8-bit units. Only bit 7 (DFn) can be read/written in 1-bit units, and bits 5 to 0 (IFCn5 to IFCn0) can be read/written in 8-bit units. (n = 0 to 3).

Be sure to set bit 6 to 0. If it is set to 1, the operation is not guaranteed.

Cautions 1. Be sure to stop the DMA operation before making changes to DTFRn register settings.

2. Except INTP0 to INPT4 and INTP20 to INTP25 (when noise elimination by an analog filter is selected), an interrupt request input in standby mode (IDLE or software STOP mode) does not trigger DMA transfer.

3. INTCM004 and INTCM005 cannot be used as DMA trigger sources.

★ **4. If the start factor for DMA transfer is changed using the IFCn5 to IFCn0 bits, be sure to clear (0) the DFn bit with the instruction immediately after the change.**

(1/3)

	<7>	6	5	4	3	2	1	0	Address	After reset
DTFR0	DF0	0	IFC05	IFC04	IFC03	IFC02	IFC01	IFC00	FFFFF810H	00H
	<7>	6	5	4	3	2	1	0	Address	After reset
DTFR1	DF1	0	IFC15	IFC14	IFC13	IFC12	IFC11	IFC10	FFFFF812H	00H
	<7>	6	5	4	3	2	1	0	Address	After reset
DTFR2	DF2	0	IFC25	IFC24	IFC23	IFC22	IFC21	IFC20	FFFFF814H	00H
	<7>	6	5	4	3	2	1	0	Address	After reset
DTFR3	DF3	0	IFC35	IFC34	IFC33	IFC32	IFC31	IFC30	FFFFF816H	00H

Bit position	Bit name	Function
7	DFn	This is a DMA transfer request flag. Only 0 can be written to this bit. 0: No DMA transfer request 1: DMA transfer request If the interrupt specified as the DMA transfer start factor occurs and it is necessary to clear the DMA transfer request while DMA transfer is disabled (including when it is aborted by NMI or forcibly stopped by software), stop the operation that has caused the interrupt (e.g., if serial reception is in progress, by disabling reception) and then clear the DFn bit. If it is clearly known that the interrupt will not occur until the next DMA transfer is started, it is not necessary to stop the operation that has caused the interrupt.

Remark n = 0 to 3

(2/3)

Bit position	Bit name	Function
5 to 0	IFCn5 to IFCn0	Sets the interrupt source that serves as the DMA transfer start factor.

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt source
0	0	0	0	0	0	DMA request from on-chip peripheral I/O disabled
0	0	0	0	0	1	INTP0
0	0	0	0	1	0	INTP1
0	0	0	0	1	1	INTP2
0	0	0	1	0	0	INTP3
0	0	0	1	0	1	INTP4
0	0	1	0	0	0	INTDET0
0	0	1	0	0	1	INTDET1
0	0	1	0	1	0	INTTM00
0	0	1	0	1	1	INTCM003
0	0	1	1	0	0	INTTM01
0	0	1	1	0	1	INTCM013
0	0	1	1	1	0	INTP100/INTCC100
0	0	1	1	1	1	INTP101/INTCC101
0	1	0	0	0	0	INTCM100
0	1	0	0	0	1	INTCM101
0	1	0	1	1	0	INTTM20
0	1	0	1	1	1	INTTM21
0	1	1	0	0	0	INTP20/INTCC20
0	1	1	0	0	1	INTP21/INTCC21
0	1	1	0	1	0	INTP22/INTCC22
0	1	1	0	1	1	INTP23/INTCC23
0	1	1	1	0	0	INTP24/INTCC24
0	1	1	1	0	1	INTP25/INTCC25
0	1	1	1	1	0	INTTM3
0	1	1	1	1	1	INTP30/INTCC30
1	0	0	0	0	0	INTP31/INTCC31
1	0	0	0	0	1	INTCM4
1	0	0	0	1	0	INTDMA0
1	0	0	0	1	1	INTDMA1
1	0	0	1	0	0	INTDMA2

Remark n = 0 to 3

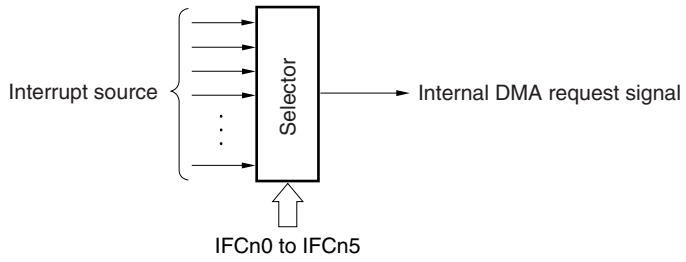
(3/3)

Bit position	Bit name	Function
5 to 0	IFCn5 to IFCn0	

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt source
1	0	0	1	0	1	INTDMA3
1	0	1	0	1	0	INTCSI0
1	0	1	0	1	1	INTCSI1
1	0	1	1	0	0	INTSR0
1	0	1	1	0	1	INTST0
1	0	1	1	1	0	INTSER0
1	0	1	1	1	1	INTSR1
1	1	0	0	0	0	INTST1
1	1	0	0	1	1	INTAD0
1	1	0	1	0	0	INTAD1
1	1	1	0	1	0	INTCM010
1	1	1	0	1	1	INTCM011
1	1	1	1	0	0	INTCM012
1	1	1	1	0	1	INTCM014
1	1	1	1	1	0	INTCM015
Other than above						Setting prohibited

Remark n = 0 to 3

The relationship between the interrupt source and the DMA transfer trigger is as follows (n = 0 to 3).



★ **Caution An interrupt request will be generated when DMA transfer starts. To prevent an interrupt from being generated, mask the interrupt by setting the interrupt request control register. DMA transfer starts even if an interrupt is masked.**

6.4 Transfer Modes

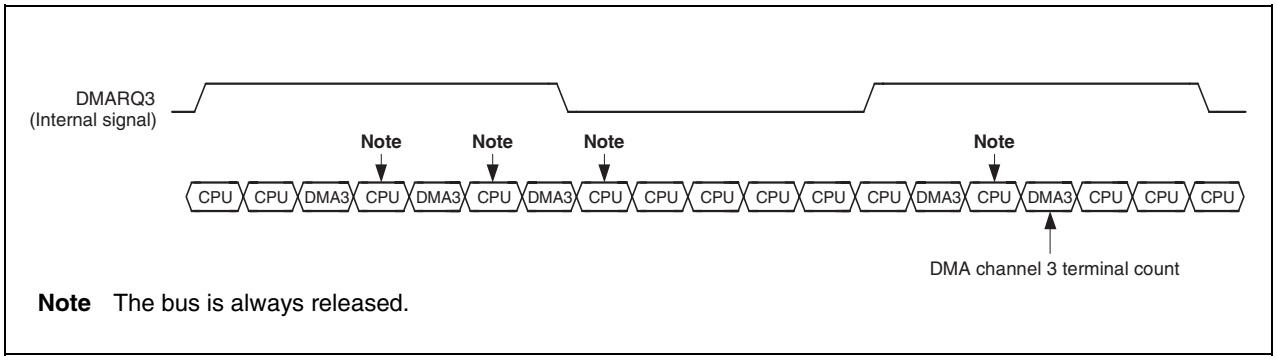
6.4.1 Single transfer mode

In single transfer mode, the DMAC releases the bus at each byte/halfword transfer. If there is a subsequent DMA transfer request, transfer is performed again once. This operation continues until a terminal count occurs.

When the DMAC has released the bus, if another higher priority DMA transfer request is issued, the higher priority DMA request always takes precedence. However, if a lower priority DMA transfer request is generated within one clock after the end of a single transfer, even if the previous higher priority DMA transfer request signal stays active, this request is not prioritized, and the next DMA transfer after the bus is released for the CPU is a transfer based on the newly generated, lower priority DMA transfer request.

Figures 6-1 to 6-4 show examples of single transfer.

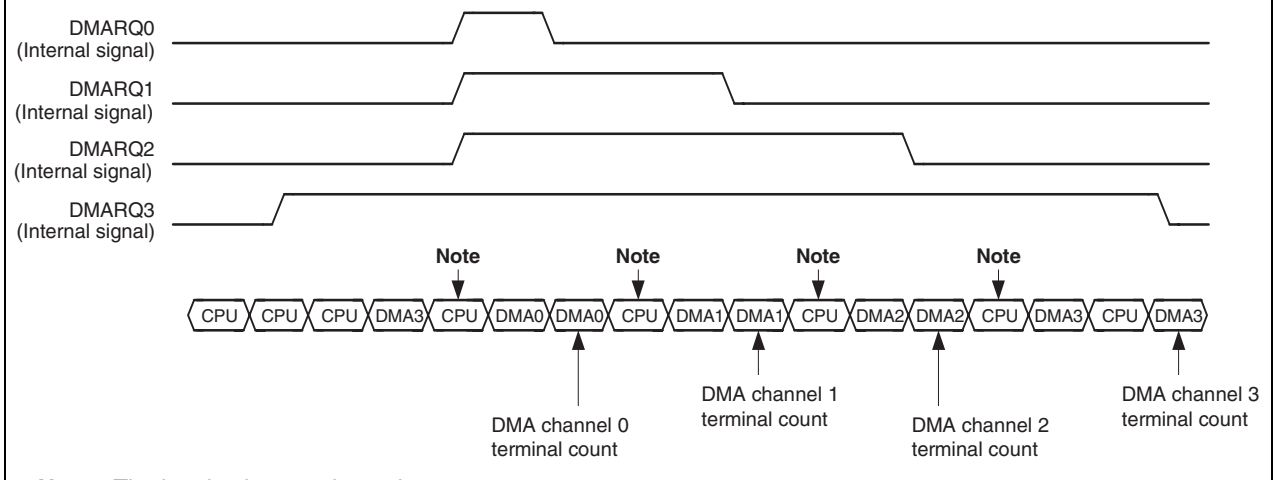
Figure 6-1. Single Transfer Example 1



Note The bus is always released.

Figure 6-2 shows a single transfer mode example in which a higher priority DMA transfer request is generated. DMA channels 0 to 2 are used for a block transfer, and channel 3 is used for a single transfer.

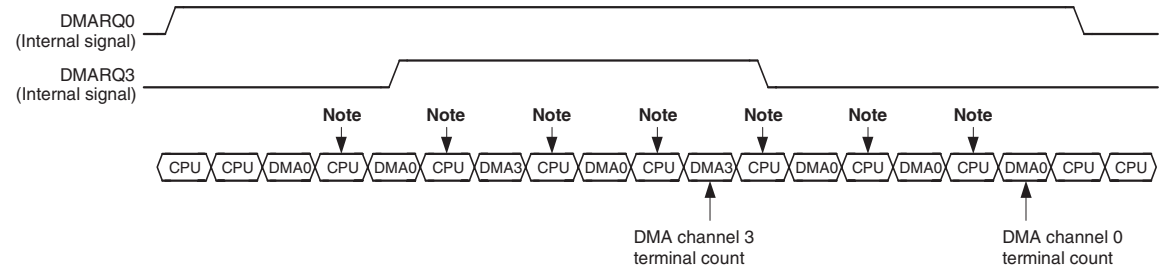
Figure 6-2. Single Transfer Example 2



Note The bus is always released.

Figure 6-3 shows a single transfer mode example in which a lower priority DMA transfer request is generated within one clock after the end of a single transfer. DMA channels 0 and 3 are used for a single transfer. When two DMA transfer request signals are activated at the same time, the two DMA transfers are performed alternately.

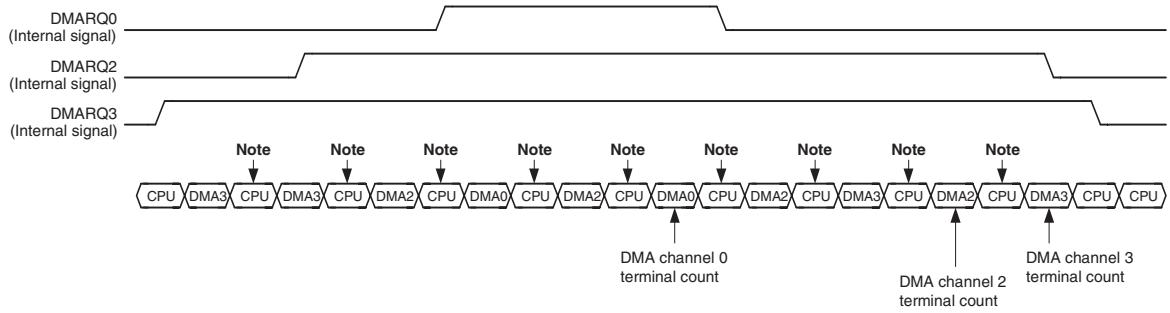
Figure 6-3. Single Transfer Example 3



Note The bus is always released.

Figure 6-4 shows a single transfer mode example in which two or more lower priority DMA transfer requests are generated within one clock after the end of a single transfer. DMA channels 0, 2, and 3 are used for a single transfer. When three or more DMA transfer request signals are activated at the same time, the two highest priority DMA transfers are performed alternately.

Figure 6-4. Single Transfer Example 4



Note The bus is always released.

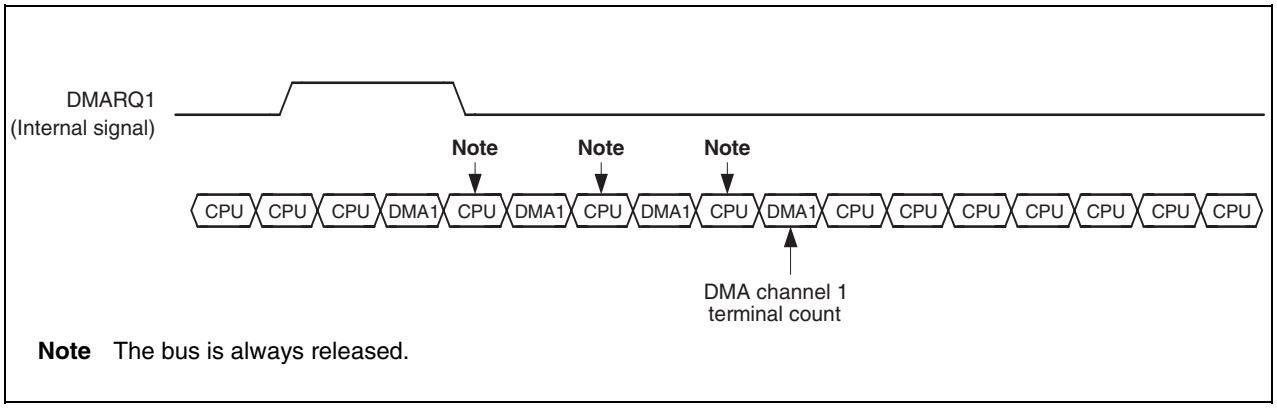
6.4.2 Single-step transfer mode

In single-step transfer mode, the DMAC releases the bus at each byte/halfword transfer. Once a DMA transfer request signal has been received, transfer continues until a terminal count occurs.

When the DMAC has released the bus, if another higher priority DMA transfer request is issued, the higher priority DMA request always takes precedence.

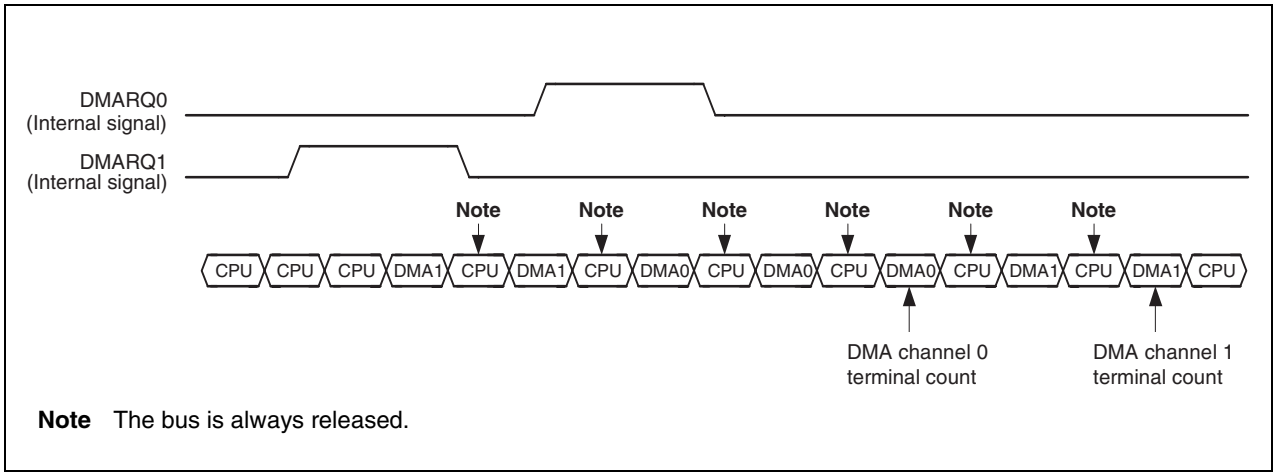
The following shows examples of single-step transfer. Figure 6-6 shows a single-step transfer mode example in which a higher priority DMA transfer request is generated and DMA channels 0 and 1 are set to the single-step transfer mode.

Figure 6-5. Single-Step Transfer Example 1



Note The bus is always released.

Figure 6-6. Single-Step Transfer Example 2



Note The bus is always released.

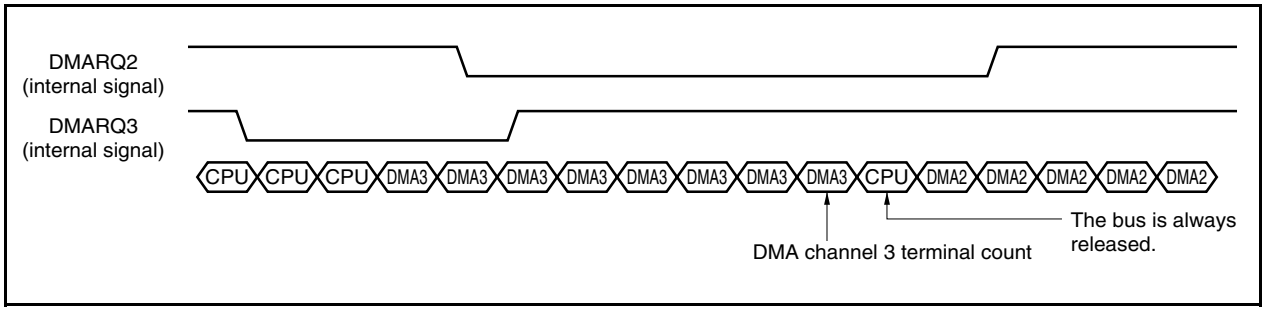
6.4.3 Block transfer mode

In the block transfer mode, once transfer starts, the DMAC continues the transfer operation without releasing the bus until a terminal count occurs. No other DMA requests are acknowledged during block transfer.

After the block transfer ends and the DMAC releases the bus, another DMA transfer can be acknowledged.

The following shows an example of block transfer in which a higher priority DMA request is issued. DMA channels 2 and 3 are in the block transfer mode.

★ **Figure 6-7. Block Transfer Example**



6.5 Transfer Types

6.5.1 Two-cycle transfer

In two-cycle transfer, data transfer is performed in two cycles, a read cycle (source to DMAC) and a write cycle (DMAC to destination).

In the first cycle, the source address is output and reading is performed from the source to the DMAC. In the second cycle, the destination address is output and writing is performed from the DMAC to the destination.

★ **Caution An idle cycle of 1 to 2 clocks is always inserted between the read cycle and write cycle.**

6.6 Transfer Target

6.6.1 Transfer type and transfer target

Table 6-1 lists the relationship between the transfer type and transfer target (√: Transfer enabled, ×: Transfer disabled).

Table 6-1. Relationship Between Transfer Type and Transfer Target

		Destination			
		Internal ROM	On-Chip Peripheral I/O[Note]	Internal RAM	External Memory, External I/O
Source	On-chip peripheral I/O[Note]	×	√	√	√
	External I/O	×	√	√	√
	Internal RAM	×	√	×	√
	External memory	×	√	√	√
	Internal ROM	×	×	×	×

★ **Note** If the transfer target is the on-chip peripheral I/O, only the single transfer mode can be used.

Cautions 1. The operation is not guaranteed for combinations of transfer destination and source marked with "×" in Table 6-1.

2. Addresses between 3FFF000H and 3FFFFFFH cannot be specified for the source and destination address of DMA transfer. Be sure to specify an address between FFFF000H and FFFFFFFH.

Remark If the target of the DMA transfer is an on-chip peripheral I/O register (transfer source/transfer destination), be sure to specify the same transfer size as the register size. For example, in the case of DMA transfer to an 8-bit register, be sure to specify byte (8-bit) transfer.

<16-bit transfer>

- Transfer from a 16-bit bus to an 8-bit bus
  A read cycle (16 bits) is generated and then a write cycle (8 bits) is generated twice successively.
- Transfer from an 8-bit bus to a 16-bit bus
  A read cycle (8 bits) is generated twice successively and then a write cycle (16 bits) is generated. The data is written to the transfer target with the lower bits first then higher bits in little endian and the higher bits then the lower bits in big endian.

<8-bit transfer>

- Transfer from a 16-bit bus to an 8-bit bus
  A read cycle (the higher 8 bits go into a high-impedance state) is generated and then a write cycle (8 bits) is generated.
- Transfer from an 8-bit bus to a 16-bit bus
  A read cycle (8 bits) is generated and then a write cycle (the higher 8 bits go into a high-impedance state) is generated. The data is written to the transfer target with the lower bits first then higher bits in little endian and the higher bits then the lower bits in big endian.

6.6.2 External bus cycles during DMA transfer (two-cycle transfer)

The external bus cycles during DMA transfer (two-cycle transfer) are shown below.

Table 6-2. External Bus Cycles During DMA Transfer (Two-Cycle Transfer)

Transfer Target	External Bus Cycle	
On-chip peripheral I/O, internal RAM	None	–
External memory, external I/O	Yes	SRAM, external ROM, external I/O access cycle

6.7 DMA Channel Priorities

The DMA channel priorities are fixed as follows.

DMA channel 0 > DMA channel 1 > DMA channel 2 > DMA channel 3

★ In the block transfer mode, the channel used for transfer is never switched.

In the single-step transfer mode, if a higher priority DMA transfer request is issued while the bus is released, the higher priority DMA transfer request is acknowledged.

Caution Be sure not to activate multiple DMA channels using the same start factor. If multiple channels are activated in this way, a lower priority DMA channel may be acknowledged prior to a higher priority DMA channel.

6.8 Next Address Setting Function

The DMA source address registers (DSAnH, DSAnL), DMA destination address registers (DDAnH, DDAnL), and DMA transfer count register (DBCn) are 2-stage FIFO buffer registers configured with a master register and slave register (n = 0 to 3).

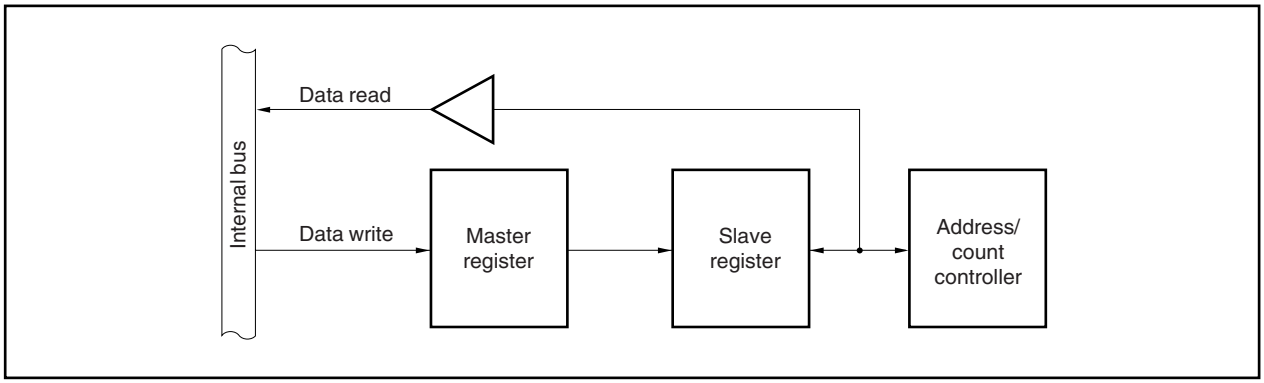
When the terminal count is issued, these registers are automatically rewritten with the value that was set immediately before.

★ If new DMA transfer setting is made to these registers during DMA transfer, therefore, the values of the registers are automatically updated to the new value after completion of transfer[Note].

★ **Note** Before making another DMA transfer setting, confirm that DMA transfer has started. If new settings are made before DMA transfer starts, the set values are overwritten to both the master and slave registers, preventing the DMA transfer based on the set value immediately before from being correctly performed.

Figure 6-8 shows the configuration of the buffer register.

Figure 6-8. Buffer Register Configuration



The actual DMA transfer is performed based on the settings of the slave register.

The settings incorporated in the master and slave registers differ as follows according to the timing (time) at which the settings were made.

(1) Time from system reset to the generation of the first DMA transfer request

The settings made are incorporated in both the master and slave registers.

(2) During DMA transfer (time from the generation to end of DMA transfer request)

The settings made are incorporated in only the master register, and not in the slave register (the slave register maintains the value set for the next DMA transfer).

However, the contents of the master register are automatically overwritten in the slave register after DMA transfer ends.

If the value of each register is read during this period, the value of the slave register is read.

★ To check that DMA transfer has been started, confirm that the first transfer has been executed by reading the DBCn register (n = 0 to 3).

(3) Time from DMA transfer end to the start of the next DMA transfer

The settings made are incorporated in both the master and slave registers.

Remark “DMA transfer end” means one of the following.

- Completion of DMA transfer (terminal count)
- Forcible termination of DMA transfer (the INITn bit of the DCHCn register is set to 1)

6.9 DMA Transfer Start Factors

There are two types of DMA transfer start factors, as shown below.

Cautions 1. Do not use both start factors ((1) and (2)) in combination for the same channel (if these two start factors are generated at the same time, only one of them is valid, but the valid start factor cannot be identified).

The operation is not guaranteed if two start factors are used in combination.

2. If DMA transfer is started via software and if the software does not correctly detect whether the expected DMA transfer operation has been completed through manipulation (setting to 1) of the STGn bit of the DCHCn register, it cannot be guaranteed whether the next (second) manipulation of the STGn bit corresponds to the start of "the next DMA transfer expected by software" (n = 0 to 3).

For example, suppose single transfer is started by manipulating the STGn bit. Even if the STGn bit is manipulated next (the second time) without checking by software whether the single transfer has actually been executed, the next (second) DMA transfer is not always executed. This is because the STGn bit may be manipulated the second time before the first DMA transfer is started or completed because, for example, DMA transfer with a higher priority had already been started when the STGn bit was manipulated for the first time.

It is therefore necessary to manipulate the STGn bit next time (the second time) after checking whether DMA transfer started by the first manipulation of the STGn bit has been completed.

Completion of DMA transfer can be checked by confirming the contents of the DBCn register.

(1) Request from software

If the STGn, Enn, and TCn bits of the DCHCn register are set as follows, DMA transfer starts (n = 0 to 3).

- STGn bit = 1
- Enn bit = 1
- TCn bit = 0

(2) Request from on-chip peripheral I/O

If, when the Enn and TCn bits of the DCHCn register are set as shown below, an interrupt request is issued from the on-chip peripheral I/O that is set in the DTFRn register, DMA transfer starts (n = 0 to 3).

- Enn bit = 1
- TCn bit = 0

6.10 Forcible Suspension

DMA transfer can be forcibly suspended by NMI input during DMA transfer.

At such a time, the DMAC resets the Enn bit of the DCHCn register of all channels to 0 and the DMA transfer disabled state is entered. An NMI request can then be acknowledged after the DMA transfer executed during NMI input is terminated (n = 0 to 3).

Initialize the DMA transfer that has been forcibly suspended by setting the INITn bit of the DCHCn register to 1 to forcibly terminate DMA transfer.

6.11 DMA Transfer End

When DMA transfer ends and the TCn bit of the DCHCn register is set to 1, a DMA transfer end interrupt (INTDMAn) is issued to the interrupt controller (INTC) (n = 0 to 3).

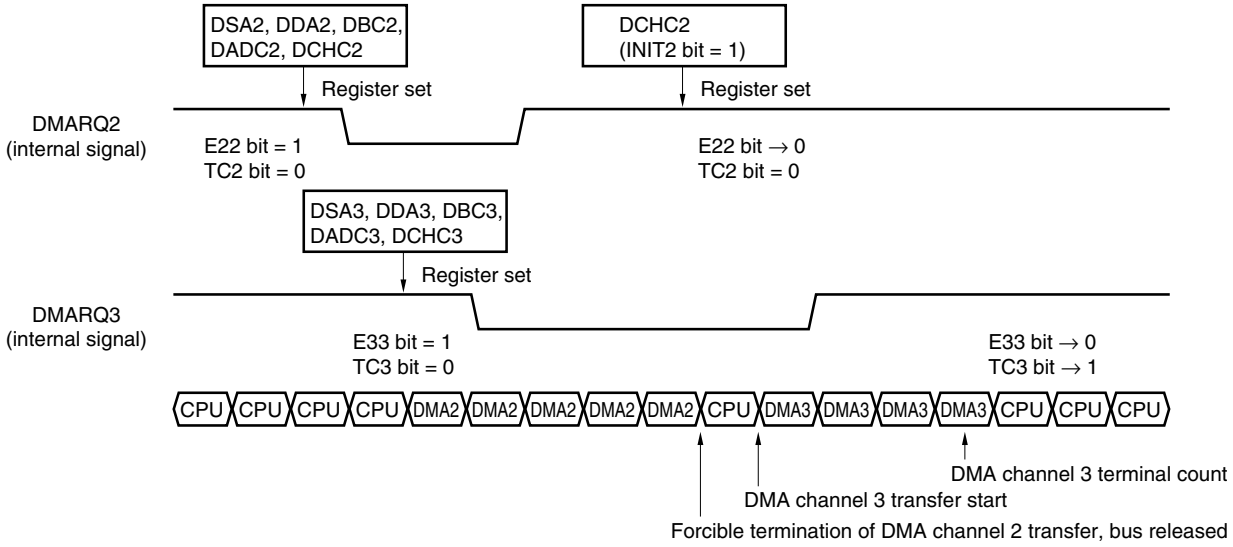
6.12 Forcible Termination

In addition to the forcible interruption operation by means of NMI input, DMA transfer can be forcibly terminated by the INITn bit of the DCHCn register (n = 0 to 3).

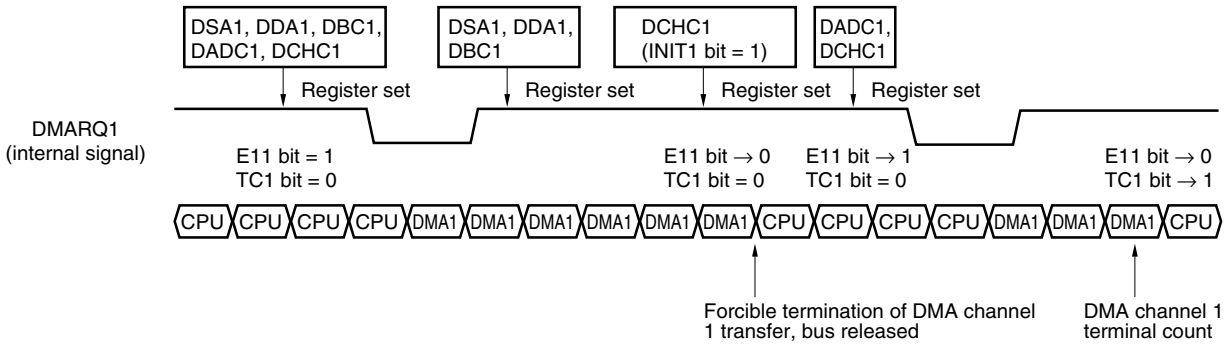
An example of forcible termination by the INITn bit of the DCHCn register is illustrated below (n = 0 to 3).

★ **Figure 6-9. Example of Forcible Termination of DMA Transfer**

(a) Block transfer via DMA channel 3 is started during block transfer via DMA channel 2



(b) When transfer is suspended during DMA channel 1 block transfer, and transfer under another condition is executed



Remark The values of the DSAn, DDAn, and DBCn registers (n = 0 to 3) are retained even when DMA transfer is forcibly terminated, because these registers are FIFO-format buffer registers. The next transfer condition can be set to these registers even while DMA transfer is in progress. On the other hand, the setting of the DADCn and DCHCn registers is invalid during DMA transfer because these registers are not buffer registers (see **6.8 Next Address Setting Function**, **6.3.4 DMA addressing control registers 0 to 3 (DADC0 to DADC3)**, and **6.3.5 DMA channel control registers 0 to 3 (DCHC0 to DCHC3)**).

6.12.1 Restrictions on forcible termination of DMA transfer

During the procedure to forcibly terminate DMA transfer using the INITn bit of the DCHCn register, the transfer may not be terminated and suspended instead even if the INITn bit has been set to 1. Consequently, when the DMA transfer of the channel that should have been forcibly terminated is resumed, DMA transfer may end after completion of an unexpected transfer count, generating a DMA transfer end interrupt (INTDMAn) (n = 0 to 3).

[Preventive measures]

The above can be prevented by software using either of the following.

(1) Temporarily stopping transfers of all DMA channels

These restrictions can be prevented if the program configuration is such that the TCn bit of the DCHCn register is expected to be 1 only during the preventive processing shown below. (The TCn bit of the DCHCn register is cleared to 0 after a read. That is, the TCn bit is cleared to 0 when preventive processing routine (ii) in step <5> of the preventive processing is executed.)

<1> Disable interrupts (DI).
<2> Read the DMA restart register (DRST) and transfer the value in the ENn bit of each channel to general-purpose registers (value A).
<3> Write 00H to the DRST register (write twice[Note]). Writing twice ensures that DMA transfer is stopped before the processing in step <4>.
<4> Set the INITn bit of the DCHCn register of the channel to be forcibly terminated to 1.
<5> Manipulate value A read in step <2> as follows (value B).
  (i) Clear the bit corresponding to the channel to be forcibly terminated to 0.
  (ii) If both the TCn bit of the DCHCn register and the ENn bit of the DRST register of the channel that is not to be forcibly terminated are 1 (the ANDed value is 1), clear the bit corresponding to the channel to 0.
<6> Write value B manipulated in step <5> to the DRST register.
<7> Enable interrupts (EI).

Note Write three times if the transfer target (transfer source or destination) is the internal RAM.

Caution Step <5> must be performed to prevent the ENn bit of the DRST register of the channel for which transfer was successfully complete during steps <2> and <3> from being illegally set to 1.

Remark n = 0 to 3

(2) Repetitively setting the INITn bit of the DCHCn register until the transfer is forcibly terminated successfully

The preventive processing steps are shown below.

<1> Copy the initial transfer count of the channel to be forcibly terminated to a general-purpose register.
<2> Set the INITn bit of the DCHCn register of the channel to be forcibly terminated to 1.
<3> Read the value of DMA transfer count register n (DBCn) of the channel to be forcibly terminated and compare it with the value copied in step <1>. If the values do not match, repeat steps <2> and <3>.

Cautions 1. When the DBCn register was read in step <3>, if DMA stops due to this restriction, the remaining number of the transfer count is read. If the forcible termination is successful, the initial transfer count is read.
2. Note that this preventive method takes longer until the forcible termination in applications in which DMA transfers of DMA channels other than those subject to forcible termination are frequently performed.

Remark n = 0 to 3

6.13 Time Required for DMA Transfer

The overhead before and after DMA transfer and minimum execution clock for DMA transfer are shown below.

Table 6-3. Minimum Number of Execution Clocks in DMA Cycle

DMA Cycle		Minimum Number of Execution Clocks
<1> Response time to DMA request		4 clocks[Note 1]
<2> Memory access	Internal RAM access	2 clocks[Note 2]
	On-chip peripheral I/O register access	4 clocks + number of waits by VSWC register

Notes 1. If the external interrupt (INTPn) is specified as a start factor of DMA transfer, the time for noise elimination is added to this value (n = 0 to 4, 100, 101, 20 to 25, 30, 31).
2. Two clocks are required for the DMA cycle.

The following shows the minimum number of execution clocks in a DMA cycle in each transfer mode.

Single transfer: DMA response time (<1>) + transfer source memory access (<2>) + 1[Note] + transfer destination memory access (<2>)
Block transfer: DMA response time (<1>) + (transfer source memory access (<2>) + 1[Note] + transfer destination memory access (<2>)) × number of transfers

Note One clock is inserted between the read and write cycles of any DMA transfer.

6.14 Cautions

(1) Memory boundary

The transfer operation is not guaranteed if the source or the destination address exceeds the area of DMA targets (external memory, internal RAM, or on-chip peripheral I/O) during DMA transfer.

(2) Transfer of misaligned data

DMA transfer of 16-bit bus width misaligned data is not supported.

★ If the source or the destination address is set to an odd address, the LSB of the address is forcibly handled as “0”.

(3) Bus arbitration for CPU

The CPU can access external memory, on-chip peripheral I/O, and internal RAM not undergoing DMA transfer.

While data transfer between external memories or to and from I/O is being performed, the CPU can access internal RAM.

While data transfer is being executed between internal RAMs, the CPU can access external memory and on-chip peripheral I/O.

★ **(4) DMA start factors**

Do not start two or more DMA channels with the same factor. If two or more DMA channels are started with the same factor, the DMA channel with a lower priority may be acknowledged before the DMA channel with a higher priority. Operation is not guaranteed in this case.

★ **(5) Program execution and DMA transfer with internal RAM**

Do not execute DMA transfer to/from the internal RAM and an instruction in the internal RAM simultaneously.

(6) Restrictions related to automatic clearing of TCn bit of DCHCn register

The TCn bit of the DCHCn register is automatically cleared to 0 when it is read. When DMA transfer is executed to transfer data to or from the internal RAM when two or more DMA transfer channels are simultaneously used, the TCn bit may not be cleared even if it is read after completion of DMA transfer (n = 0 to 3).

Caution This restriction does not apply if one of the following conditions is satisfied.

- **Only one channel of DMA transfer is used.**
- **DMA is not executed to transfer data to or from the internal RAM.**

[Preventive measures]

To read the TCn bit of the DCHCn register of the DMA channel that is used to transfer data to or from the internal RAM, be sure to read the TCn bit three times in a row. This can accurately clear the TCn bit to 0.

(7) Read values of DSAn and DDAn registers

If the values of the DSAn and DDAn registers are read during DMA transfer, the values in the middle of being updated may be read (n = 0 to 3).

For example, if the DSAnH register and the DSAnL register are read in that order when the value of the DMA transfer source address (DSAn register) is “0000FFFFH” and the counting direction is incremental (when the SADn1 and SADn0 bits of the DADCn register = 00), the value of the DSAnL register differs as follows depending on whether DMA transfer is executed immediately after the DSAnH register has been read.

(a) If DMA transfer does not occur while the DSAn register is being read

<1> Reading DSAnH register: DSAnH = 0000H
<2> Reading DSAnL register: DSAnL = FFFFH

(b) If DMA transfer occurs while the DSAn register is being read

<1> Reading DSAnH register: DSAnH = 0000H
<2> Occurrence of DMA transfer
<3> Incrementing DSAn register : DSAn = 00010000H
<4> Reading DSAnL register: DSAnL = 0000H

CHAPTER 7 INTERRUPT/EXCEPTION PROCESSING FUNCTION

The V850E/IA2 is provided with an interrupt controller (INTC) that can process a total of 48 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 V850E/IA2 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).

Eight levels of software-programmable priorities can be specified for each interrupt request. Interrupt servicing starts after at least 4 system clocks (100 ns (@ 40 MHz)) following the generation of an interrupt request.

7.1 Features

○ Interrupts
- Non-maskable interrupts: 1 source

★ **Caution P00 alternately functions as NMI, and is fixed to input. P00 and NMI cannot be switched. If the P00 bit of the P0 register is read, the level of the P00/NMI pin is read.**
Set the valid edge of the NMI pin using the ESN0 bit of the INTM0 register (default value: falling edge detection).

- Maskable interrupts: 47 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[Note], edge detection, and valid edge specification for external interrupt request signals.

Note For details of the noise eliminator, refer to **12.4 Noise Eliminator.**

○ Exceptions
- Software exceptions: 32 sources
- Exception traps: 2 sources (illegal opcode exception and debug trap)

Interrupt/exception sources are listed in Table 7-1.

Table 7-1. Interrupt/Exception Source List (1/2)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Controlling Register	Generating Source	Generating Unit				
Reset	Interrupt	RESET	–	$\overline{\text{RESET}}$ input	Pin	–	0000H	00000000H	Undefined
Non-maskable	Interrupt	NMI0	–	NMI input	Pin	–	0010H	00000010H	nextPC
Software exception	Exception	TRAP0n[Note 1]	–	TRAP instruction	–	–	004nH[Note 1]	00000040H	nextPC
	Exception	TRAP1n[Note 1]	–	TRAP instruction	–	–	005nH[Note 1]	00000050H	nextPC
Exception trap	Exception	ILGOP/DBG0	–	Illegal opcode/ DBTRAP instruction	–	–	0060H	00000060H	nextPC
Maskable	Interrupt	INTP0	P0IC0	INTP0 pin	Pin	0	0080H	00000080H	nextPC
	Interrupt	INTP1	P0IC1	INTP1 pin	Pin	1	0090H	00000090H	nextPC
	Interrupt	INTP2	P0IC2	INTP2 pin	Pin	2	00A0H	000000A0H	nextPC
	Interrupt	INTP3	P0IC3	INTP3 pin	Pin	3	00B0H	000000B0H	nextPC
	Interrupt	INTP4	P0IC4	INTP4 pin	Pin	4	00C0H	000000C0H	nextPC
	Interrupt	–	–	Not used[Note 2]	–	–	–	000000D0H	–
	Interrupt	–	–	Not used[Note 2]	–	–	–	000000E0H	–
	Interrupt	INTDET0	DETIC0	AD0 voltage detection	ADC0	5	00F0H	000000F0H	nextPC
	Interrupt	INTDET1	DETIC1	AD1 voltage detection	ADC1	6	0100H	00000100H	nextPC
	Interrupt	INTTM00	TM0IC0	TM00 underflow	TM00	7	0110H	00000110H	nextPC
	Interrupt	INTCM003	CM03IC0	CM003 match	TM00	8	0120H	00000120H	nextPC
	Interrupt	INTTM01	TM0IC1	TM01 underflow	TM01	9	0130H	00000130H	nextPC
	Interrupt	INTCM013	CM03IC1	CM013 match	TM01	10	0140H	00000140H	nextPC
	Interrupt	INTP100/ INTCC100	CC10IC0	INTP100 pin/ CC100 match	Pin/TM10	11	0150H	00000150H	nextPC
★	Interrupt	INTP101/ INTCC101	CC10IC1	INTP101/INTP100 pin[Note 3]/ CC101 match	Pin/TM10	12	0160H	00000160H	nextPC
	Interrupt	INTCM100	CM10IC0	CM100 match	TM10	13	0170H	00000170H	nextPC
	Interrupt	INTCM101	CM10IC1	CM101 match	TM10	14	0180H	00000180H	nextPC
	Interrupt	–	–	Not used[Note 2]	–	–	–	00000190H	–
	Interrupt	–	–	Not used[Note 2]	–	–	–	000001A0H	–
	Interrupt	–	–	Not used[Note 2]	–	–	–	000001B0H	–
	Interrupt	–	–	Not used[Note 2]	–	–	–	000001C0H	–
	Interrupt	INTTM20	TM2IC0	TM20 overflow	TM20	15	01D0H	000001D0H	nextPC
	Interrupt	INTTM21	TM2IC1	TM20 overflow	TM21	16	01E0H	000001E0H	nextPC
	Interrupt	INTP20/INTCC20	CC2IC0	INTP20 pin/CC20 match	Pin/TM20	17	01F0H	000001F0H	nextPC
	Interrupt	INTP21/INTCC21	CC2IC1	INTP21 pin/ CC21 match	Pin/ TM20/TM21	18	0200H	00000200H	nextPC
	Interrupt	INTP22/INTCC22	CC2IC2	INTP22 pin/ CC22 match	Pin/ TM20/TM21	19	0210H	00000210H	nextPC
	Interrupt	INTP23/INTCC23	CC2IC3	INTP23 pin/ CC23 match	Pin/ TM20/TM21	20	0220H	00000220H	nextPC
	Interrupt	INTP24/INTCC24	CC2IC4	INTP24 pin/ CC24 match	Pin/ TM20/TM21	21	0230H	00000230H	nextPC
	Interrupt	INTP25/INTCC25	CC2IC5	INTP25 pin/ CC25 match	Pin/TM21	22	0240H	00000240H	nextPC
	Interrupt	INTTM3	TM3IC0	TM3 overflow	TM3	23	0250H	00000250H	nextPC

Notes 1. n = 0 to FH

2. Reserved for expansion to the V850E/IA1.

★ **3.** Select using the CSL10 register.

Table 7-1. Interrupt/Exception Source List (2/2)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Controlling Register	Generating Source	Generating Unit				
Maskable	Interrupt	INTP30/INTCC30	CC3IC0	INTP30 pin/ CC30 match	Pin/TM3	24	0260H	00000260H	nextPC
	Interrupt	INTP31/INTCC31	CC3IC1	INTP31 pin/ CC31 match	Pin/TM3	25	0270H	00000270H	nextPC
	Interrupt	INTCM4	CM4IC0	CM4 match signal	TM4	26	0280H	00000280H	nextPC
	Interrupt	INTDMA0	DMAIC0	End of DMA0 transfer	DMA	27	0290H	00000290H	nextPC
	Interrupt	INTDMA1	DMAIC1	End of DMA1 transfer	DMA	28	02A0H	000002A0H	nextPC
	Interrupt	INTDMA2	DMAIC2	End of DMA2 transfer	DMA	29	02B0H	000002B0H	nextPC
	Interrupt	INTDMA3	DMAIC3	End of DMA3 transfer	DMA	30	02C0H	000002C0H	nextPC
	Interrupt	–	–	Not used[Note]	–	–	–	000002D0H	–
	Interrupt	–	–	Not used[Note]	–	–	–	000002E0H	–
	Interrupt	–	–	Not used[Note]	–	–	–	000002F0H	–
	Interrupt	–	–	Not used[Note]	–	–	–	00000300H	–
	Interrupt	INTCSI0	CSIIC0	CSI0 transmission complete	CSI0	31	0310H	00000310H	nextPC
	Interrupt	INTCSI1	CSIIC1	CSI1 reception complete	CSI1	32	0320H	00000320H	nextPC
	Interrupt	INTSR0	SRIC0	UART0 reception complete	UART0	33	0330H	00000330H	nextPC
	Interrupt	INTST0	STIC0	UART0 transmission complete	UART0	34	0340H	00000340H	nextPC
	Interrupt	INTSER0	SEIC0	UART0 receiver error	UART0	35	0350H	00000350H	nextPC
	Interrupt	INTSR1	SRIC1	UART1 reception complete	UART1	36	0360H	00000360H	nextPC
	Interrupt	INTST1	STIC1	UART1 transmission complete	UART1	37	0370H	00000370H	nextPC
	Interrupt	–	–	Not used[Note]	–	–	–	00000380H	–
	Interrupt	–	–	Not used[Note]	–	–	–	00000390H	–
	Interrupt	INTAD0	ADIC0	End of AD0 conversion	ADC0	38	03A0H	000003A0H	nextPC
	Interrupt	INTAD1	ADIC1	End of AD0 conversion	ADC1	39	03B0H	000003B0H	nextPC
	Interrupt	–	–	Not used[Note]	–	–	–	000003C0H	–
	Interrupt	–	–	Not used[Note]	–	–	–	000003D0H	–
	Interrupt	–	–	Not used[Note]	–	–	–	000003E0H	–
	Interrupt	INTCM010	CM00IC1	CM010 match	TM01	40	03F0H	000003F0H	nextPC
	Interrupt	INTCM011	CM01IC1	CM011 match	TM01	41	0400H	00000400H	nextPC
	Interrupt	INTCM012	CM02IC1	CM012 match	TM01	42	0410H	00000410H	nextPC
	Interrupt	INTCM014	CM04IC1	CM014 match	TM01	43	0420H	00000420H	nextPC
	Interrupt	INTCM015	CM05IC1	CM015 match	TM01	44	0430H	00000430H	nextPC
	Interrupt	INTCM004	CM04IC0	CM004 match	TM00	45	0440H	00000440H	nextPC
	Interrupt	INTCM005	CM05IC0	CM005 match	TM00	46	0450H	00000450H	nextPC

Note Reserved for expansion to the V850E/IA1.

★ **Remarks 1.** Default priority: The priority order when two or more maskable interrupt requests are generated at the same time. The highest priority is 0.

Restored PC: The value of the program counter (PC) saved to EIPC, FEPC, or DBPC of CPU 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. In this case, the address of the aborted instruction is the restore PC.)

- 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).

7.2 Non-Maskable Interrupt

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.

A non-maskable interrupt request is input from the NMI pin. When the valid edge specified by bit 0 (ESN0) of the external interrupt mode register 0 (INTM0) is detected on the NMI pin, the interrupt occurs.

While the service program of the non-maskable interrupt is being executed, the acknowledgment of another non-maskable interrupt request is held pending. The pending NMI is acknowledged after the original service program of the non-maskable interrupt under execution has been terminated (by the RETI instruction). Note that if two or more NMI requests are input during the execution of the service program for an NMI, the number of NMIs that will be acknowledged after the RETI instruction has been executed is only one.

7.2.1 Operation

If a non-maskable interrupt is generated, 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 NP and ID bits of the PSW and clears the EP bit.
(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 7-1.

Figure 7-1. Servicing Configuration of Non-Maskable Interrupt

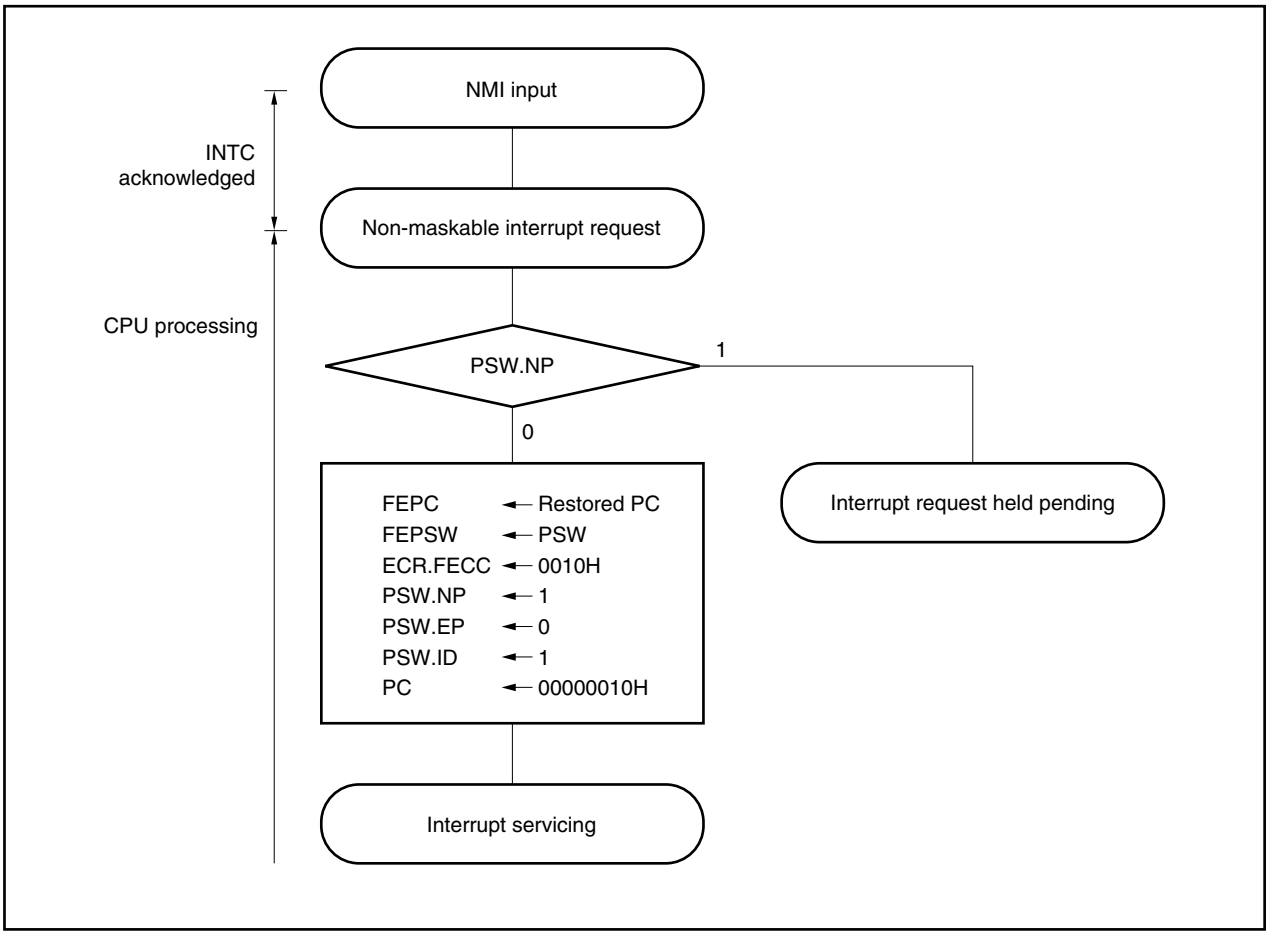
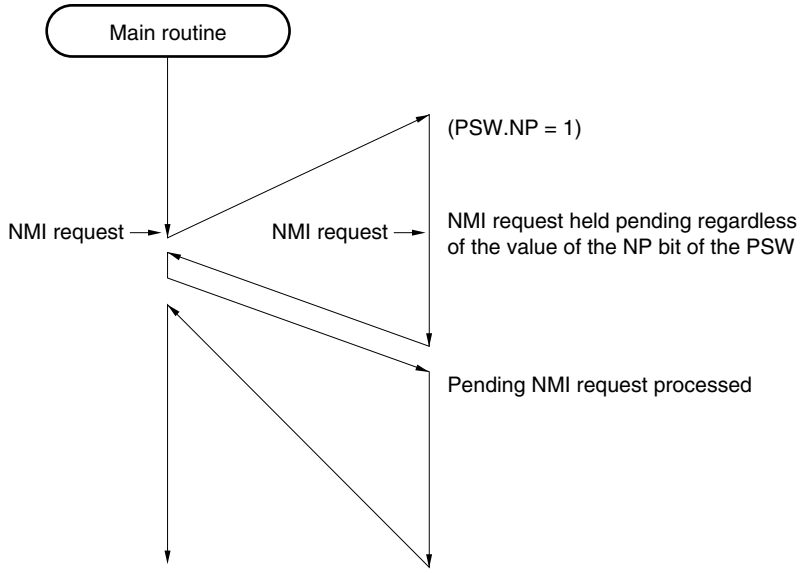
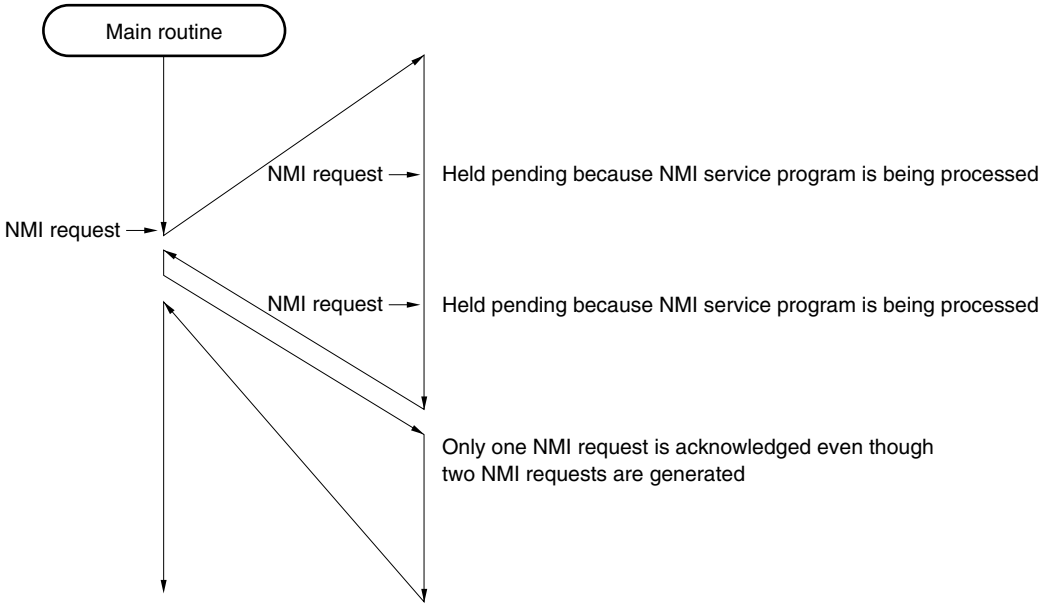


Figure 7-2. Acknowledging Non-Maskable Interrupt Request

(a) If a new NMI request is generated while an NMI service program is being executed



(b) If a new NMI request is generated twice while an NMI service program is being executed



7.2.2 Restore

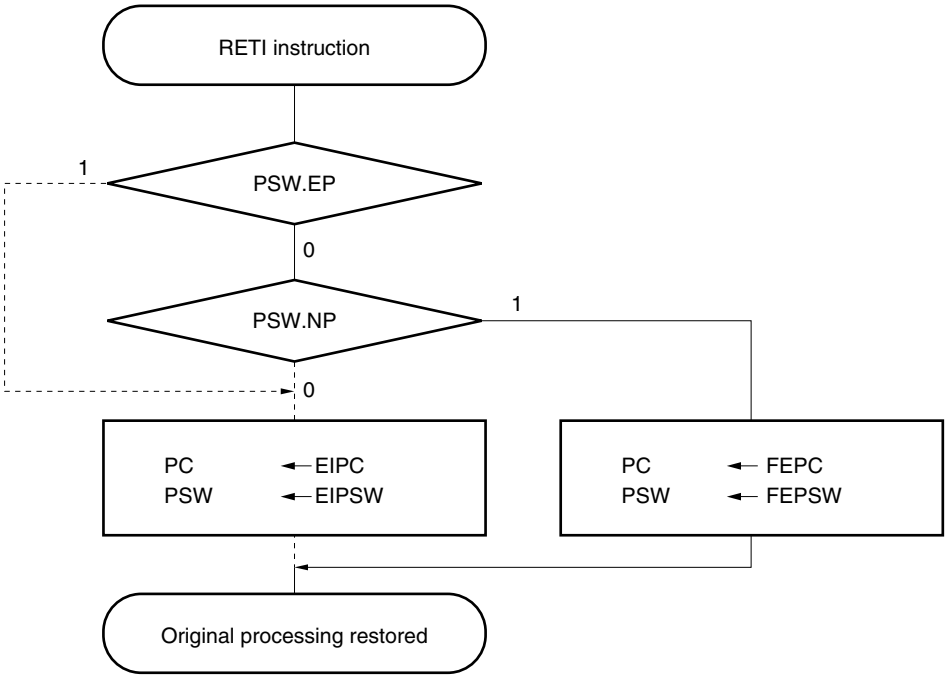
Execution is restored from the non-maskable interrupt servicing by the RETI instruction.

When the RETI instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

(1) Restores the values of the PC and the PSW from FEPC and FEPSW, respectively, because the EP bit of the PSW is 0 and the NP bit of the PSW is 1.
(2) Transfers control back to the address of the restored PC and PSW.

Figure 7-3 illustrates how the RETI instruction is processed.

Figure 7-3. 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 lines show the CPU processing flow.

7.2.3 Non-maskable interrupt status flag (NP)

The NP flag is a status flag that indicates that non-maskable interrupt (NMI) servicing is under execution.

This flag is set when an NMI interrupt has been acknowledged, and masks all interrupt requests and exceptions to prohibit multiple interrupts from being acknowledged.

	31																								8	7	6	5	4	3	2	1	0	
PSW	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NP	EP	ID	SAT	CY	OV	S	Z	After reset 00000020H

Bit position	Bit name	Function
7	NP	Indicates whether NMI interrupt servicing is in progress. 0: No NMI interrupt servicing 1: NMI interrupt currently being serviced

7.2.4 Edge detection function

(1) External interrupt mode register 0 (INTM0)

External interrupt mode register 0 (INTM0) is a register that specifies the valid edge of a non-maskable interrupt (NMI). The NMI valid edge can be specified to be either the rising edge or the falling edge by the ESN0 bit.

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

	7	6	5	4	3	2	1	<0>	
INTM0	0	0	0	0	0	0	0	ESN0	Address FFFFF880H, After reset 00H

Bit position	Bit name	Function
0	ESN0	Specifies the NMI pin's valid edge. 0: Falling edge 1: Rising edge

7.3 Maskable Interrupts

Maskable interrupt requests can be masked by interrupt control registers. The V850E/IA2 has 47 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 servicing 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.

However, if multiple interrupts are executed, the following processing is necessary.

<1> Save EIPC and EIPSW in memory or a general-purpose register before executing the EI instruction.
<2> Execute the DI instruction before executing the RETI instruction, then reset EIPC and EIPSW with the values saved in <1>.

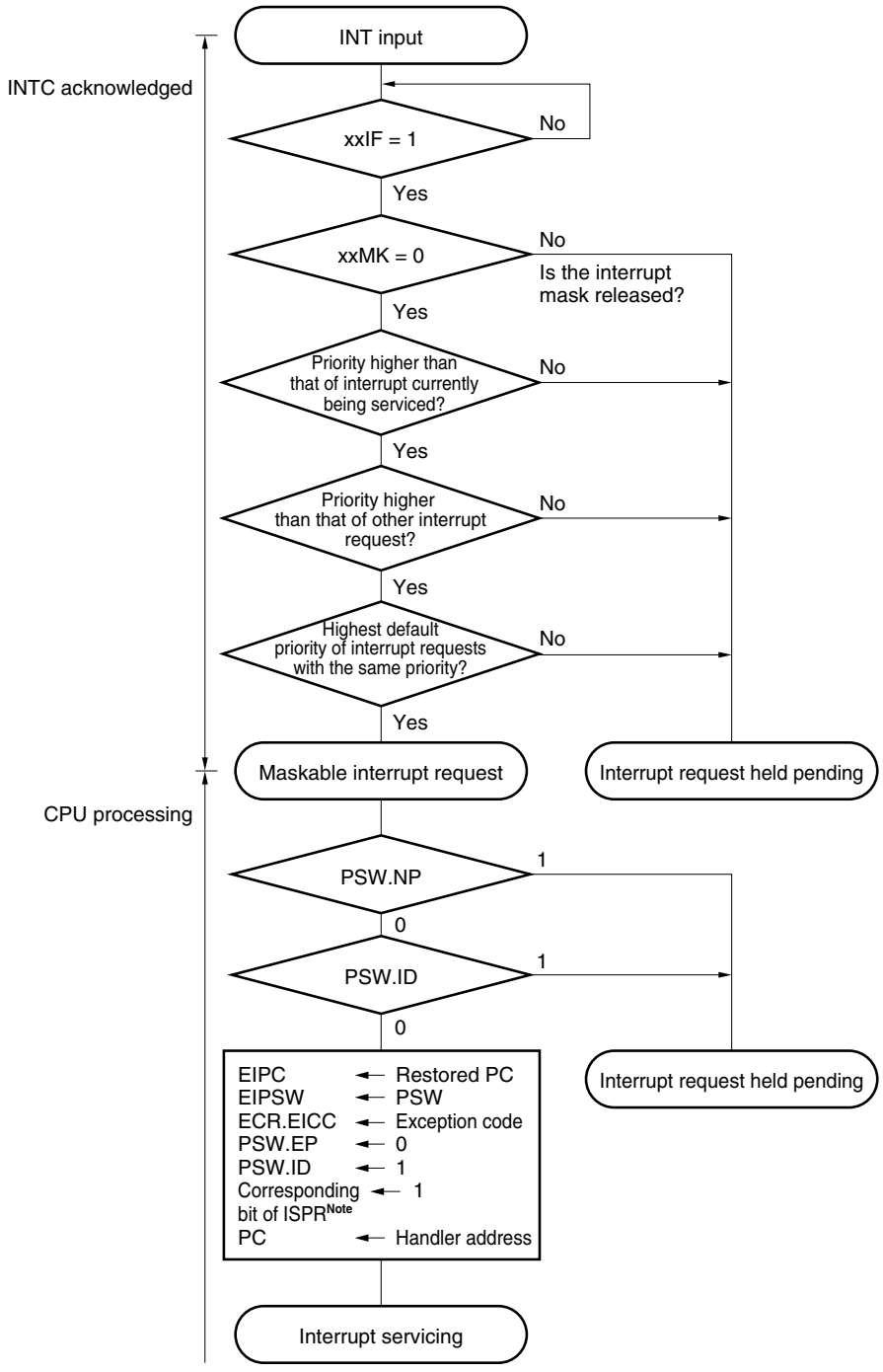
7.3.1 Operation

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 ID bit of the PSW and clears the EP bit.
(5) Sets the handler address corresponding to each interrupt to the PC, and transfers control.

The servicing configuration of a maskable interrupt is shown in Figure 7-4.

Figure 7-4. Maskable Interrupt Servicing



Note For details of the ISPR register, see **7.3.6 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.

7.3.2 Restore

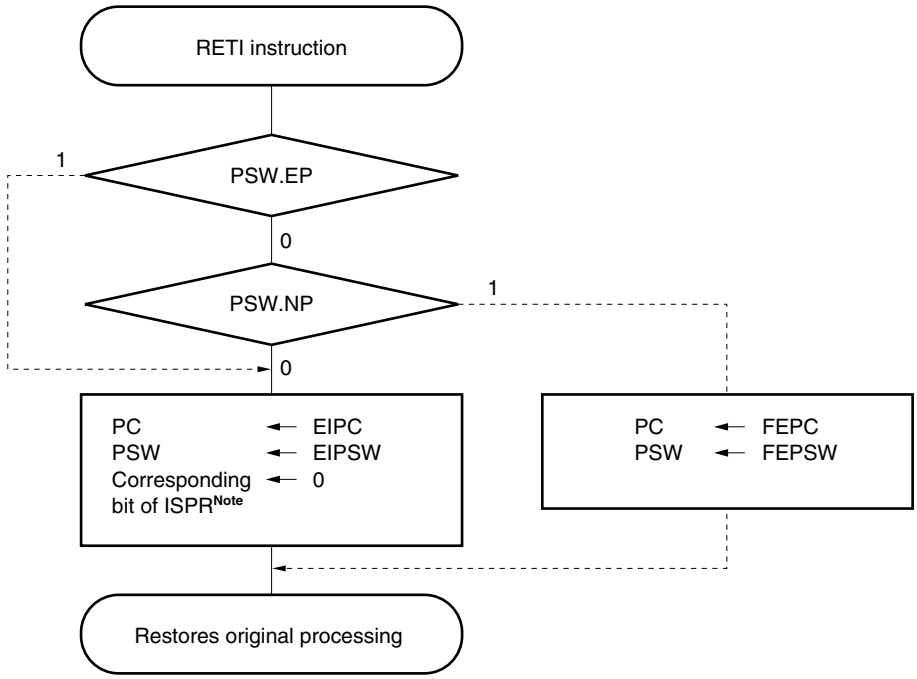
Recovery from maskable interrupt servicing is carried out by the RETI instruction.

When the RETI instruction is executed, the CPU performs the following steps, and transfers control to the address of the restored PC.

(1) Restores the values of the PC and the PSW from EIPC and EIPSW because the EP bit of the PSW is 0 and the NP bit of the PSW is 0.
(2) Transfers control to the address of the restored PC and PSW.

Figure 7-5 illustrates the processing of the RETI instruction.

Figure 7-5. RETI Instruction Processing



Note For details of the ISPR register, see **7.3.6 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 lines show the CPU processing flow.

7.3.3 Priorities of maskable interrupts

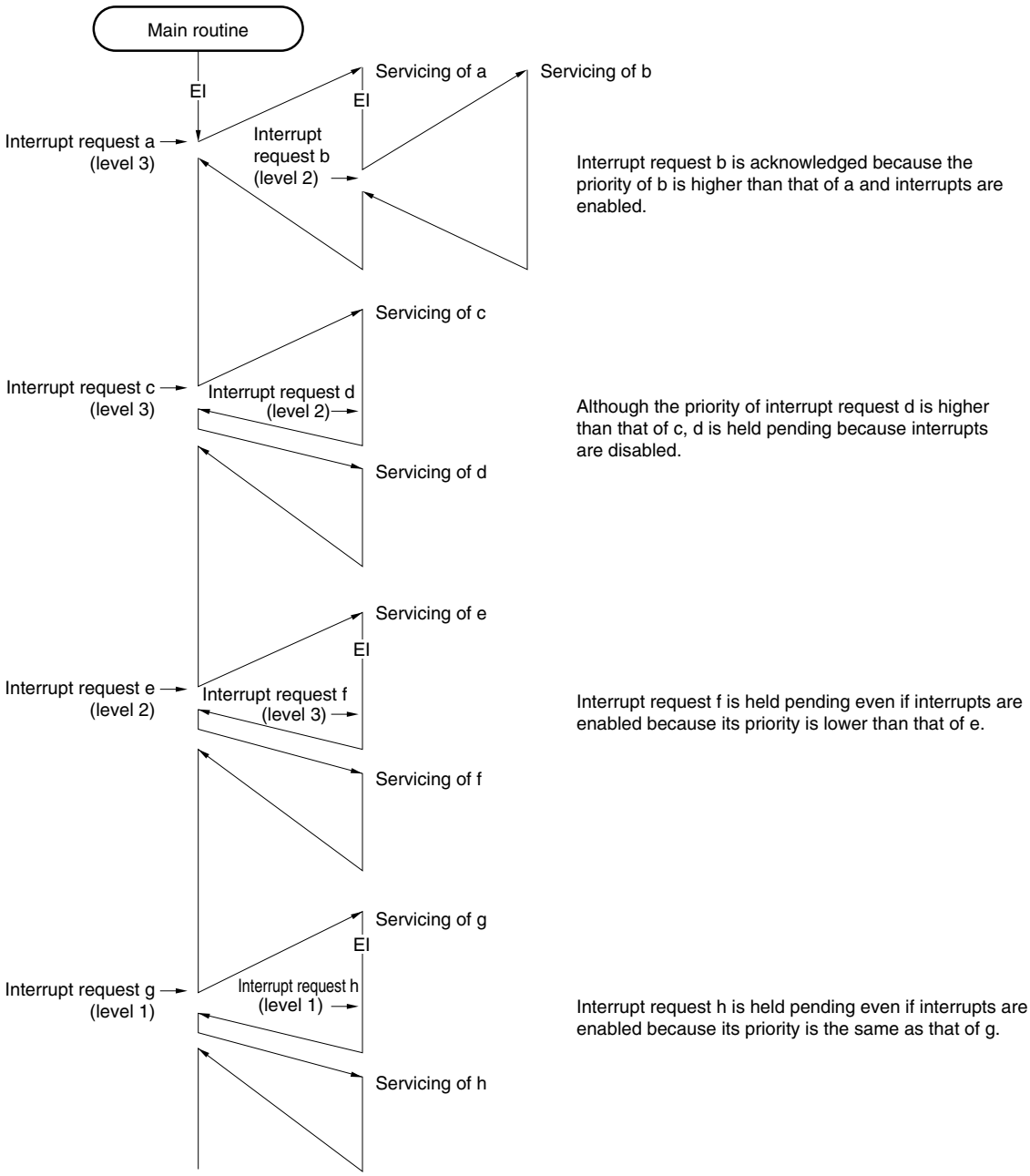
The V850E/IA2 provides multiple interrupt servicing in which an interrupt is acknowledged while another interrupt is being serviced. Multiple interrupts can be controlled by priority levels.

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 7-1 Interrupt/Exception 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 7-2**)
n: Peripheral unit number (refer to **Table 7-2**)

Figure 7-6. Example of Servicing in Which Another Interrupt Request Is Issued While an Interrupt Is Being Serviced (1/2)



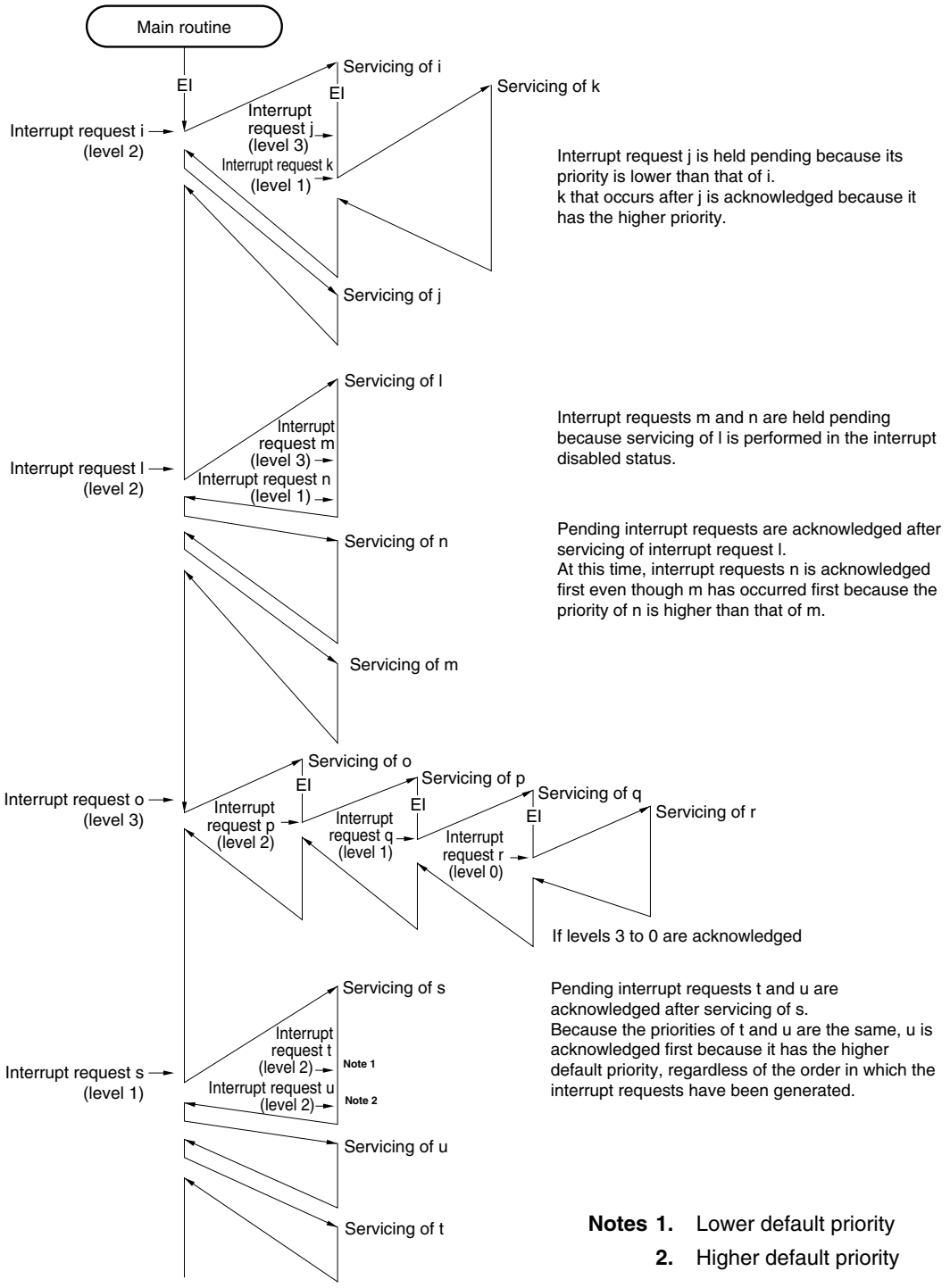
Caution The values of the EIPC and EIPSW registers must be saved before executing multiple interrupts.

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.

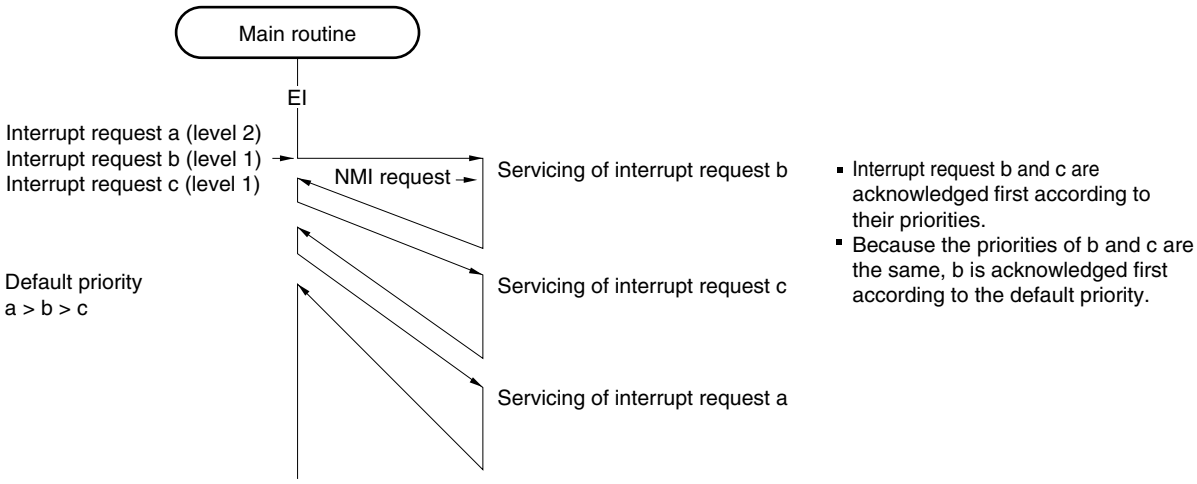
2. The default priority in the figure indicates the relative priority between two interrupt requests.

Figure 7-6. Example of Servicing in Which Another Interrupt Request Is Issued While an Interrupt Is Being Serviced (2/2)



Caution The values of the EIPC and EIPSW registers must be saved before executing multiple interrupts. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Figure 7-7. Example of Servicing Interrupt Requests Generated Simultaneously



Caution The values of the EIPC and EIPSW registers must be saved before executing multiple interrupts. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Remark a to c in the figure are pseudo names given to interrupt requests for the sake of explanation.

7.3.4 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/written in 8-bit or 1-bit units.

Caution Read the xxIFn bit of the xxICn register in the interrupt disabled (DI) state. Otherwise if the timing of interrupt acknowledgment and bit reading conflict, normal values may not be read.

	<7>	<6>	5	4	3	<2>	<1>	<0>	Address	After reset
xxICn	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0	FFFFF110H to FFFF18AH	47H

Bit position	Bit name	Function
7	xxIFn	This is an interrupt request flag. 0: Interrupt request not issued 1: Interrupt request issued The flag xxIFn is reset automatically by the hardware if an interrupt request is acknowledged.
6	xxMKn	This is an interrupt mask flag. 0: Enables interrupt servicing 1: Disables interrupt servicing (pending)
2 to 0	xxPRn2 to xxPRn0	8 levels of priority order are specified for each interrupt.

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).

Remark xx: Identification name of each peripheral unit (refer to **Table 7-2**)
n: Peripheral unit number (refer to **Table 7-2**).

The address and bit of each interrupt control register are as follows.

Table 7-2. Addresses and Bits of Interrupt Control Registers (1/2)

Address	Register	Bit							
		<7>	<6>	5	4	3	<2>	<1>	<0>
FFFFF110H	P0IC0	P0IF0	P0MK0	0	0	0	P0PR02	P0PR01	P0PR00
FFFFF112H	P0IC1	P0IF1	P0MK1	0	0	0	P0PR12	P0PR11	P0PR10
FFFFF114H	P0IC2	P0IF2	P0MK2	0	0	0	P0PR22	P0PR21	P0PR20
FFFFF116H	P0IC3	P0IF3	P0MK3	0	0	0	P0PR32	P0PR31	P0PR30
FFFFF118H	P0IC4	P0IF4	P0MK4	0	0	0	P0PR42	P0PR41	P0PR40
FFFFF11AH	Not used[Note]	–	–	–	–	–	–	–	–
FFFFF11CH	Not used[Note]	–	–	–	–	–	–	–	–
FFFFF11EH	DETIC0	DETIF0	DETMK0	0	0	0	DETPR02	DETPR01	DETPR00
FFFFF120H	DETIC1	DETIF1	DETMK1	0	0	0	DETPR12	DETPR11	DETPR10
FFFFF122H	TM0IC0	TM0IF0	TM0MK0	0	0	0	TM0PR02	TM0PR01	TM0PR00
FFFFF124H	CM3IC0	CM03IF0	CM03MK0	0	0	0	CM03PR02	CM03PR01	CM03PRC0
FFFFF126H	TM0IC1	TM0IF1	TM0MK1	0	0	0	TM0PR12	TM0PR11	TM0PR10
FFFFF128H	CM03IC1	CM03IF1	CM03MK1	0	0	0	CM03PR12	CM03PR11	CM03PR10
FFFFF12AH	CC10IC0	CC10IF0	CC10MK0	0	0	0	CC10PR02	CC10PR01	CC10PR00
FFFFF12CH	CC1CIC1	CC10IF1	CC10MK1	0	0	0	CC10PR12	CC10PR11	CC10PR10
FFFFF12EH	CM10IC0	CM10IF0	CM10MK0	0	0	0	CM10PR02	CM10PR01	CM10PR00
FFFFF130H	CM10IC1	CM10IF1	CM10MK1	0	0	0	CM10PR12	CM10PR11	CM10PR10
FFFFF132H	Not used[Note]	–	–	–	–	–	–	–	–
FFFFF134H	Not used[Note]	–	–	–	–	–	–	–	–
FFFFF136H	Not used[Note]	–	–	–	–	–	–	–	–
FFFFF138H	Not used[Note]	–	–	–	–	–	–	–	–
FFFFF13AH	TM2IC0	TM2IF0	TM2MK0	0	0	0	TM2PR02	TM2PR01	TM2PR00
FFFFF13CH	TM2IC1	TM2IF1	TM2MK1	0	0	0	TM2PR12	TM2PR11	TM2PR10
FFFFF13EH	CC2IC0	CC2IF0	CC2MK0	0	0	0	CC2PR02	CC2PR01	CC2PR00
FFFFF140H	CC2IC1	CC2IF1	CC2MK1	0	0	0	CC2PR12	CC2PR11	CC2PR10
FFFFF142H	CC2IC2	CC2IF2	CC2MK2	0	0	0	CC2PR22	CC2PR21	CC2PR20
FFFFF144H	CC2IC3	CC2IF3	CC2MK3	0	0	0	CC2PR32	CC2PR31	CC2PR30
FFFFF146H	CC2IC4	CC2IF4	CC2MK4	0	0	0	CC2PR42	CC2PR41	CC2PR40
FFFFF148H	CC2IC5	CC2IF5	CC2MK5	0	0	0	CC2PR52	CC2PR51	CC2PR50
FFFFF14AH	TM3IC0	TM3IF0	TM3MK0	0	0	0	TM3PR02	TM3PR01	TM3PR00
FFFFF14CH	CC3IC0	CC3IF0	CC3MK0	0	0	0	CC3PR02	CC3PR01	CC3PR00
FFFFF14EH	CC3IC1	CC3IF1	CC3MK1	0	0	0	CC3PR12	CC3PR11	CC3PR10

Note Reserved for expansion to V850E/IA1.

Table 7-2. Addresses and Bits of Interrupt Control Registers (2/2)

Address	Register	Bit							
		<7>	<6>	5	4	3	<2>	<1>	<0>
FFFFF150H	CM4IC0	CM4IF0	CM4MK0	0	0	0	CM4PR02	CM4PR01	CM4PR00
FFFFF152H	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00
FFFFF154H	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10
FFFFF156H	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20
FFFFF158H	DMAIC3	DMAIF3	DMAMK3	0	0	0	DMAPR32	DMAPR31	DMAPR30
FFFFF15AH	Not used[Note]	–	–	–	–	–	–	–	–
FFFFF15CH	Not used[Note]	–	–	–	–	–	–	–	–
FFFFF15EH	Not used[Note]	–	–	–	–	–	–	–	–
FFFFF160H	Not used[Note]	–	–	–	–	–	–	–	–
FFFFF162H	CSIIC0	CSIIF0	CSIMK0	0	0	0	CSIPR02	CSIPR01	CSIPR00
FFFFF164H	CSIIC1	CSIIF1	CSIMK1	0	0	0	CSIPR12	CSIPR11	CSIPR10
FFFFF166H	SRIC0	SRIF0	SRMK0	0	0	0	SRPR02	SRPR01	SRPR00
FFFFF168H	STIC0	STIF0	STMK0	0	0	0	STPR02	STPR01	STPR00
FFFFF16AH	SEIC0	SEIF0	SEMK0	0	0	0	SEPR02	SEPR01	SEPR00
FFFFF16CH	SRIC1	SRIF1	SRMK1	0	0	0	SRPR12	SRPR11	SRPR10
FFFFF16EH	STIC1	STIF1	STMK1	0	0	0	STPR12	STPR11	STPR10
FFFFF170H	Not used[Note]	–	–	–	–	–	–	–	–
FFFFF172H	Not used[Note]	–	–	–	–	–	–	–	–
FFFFF174H	ADIC0	ADIF0	ADMK0	0	0	0	ADPR02	ADPR01	ADPR00
FFFFF176H	ADIC1	ADIF1	ADMK1	0	0	0	ADPR12	ADPR11	ADPR10
FFFFF178H	Not used[Note]	–	–	–	–	–	–	–	–
FFFFF17AH	Not used[Note]	–	–	–	–	–	–	–	–
FFFFF17CH	Not used[Note]	–	–	–	–	–	–	–	–
FFFFF17EH	CM00IC1	CM00IF1	CM00MK1	0	0	0	CM00PR12	CM00PR11	CM00PR10
FFFFF180H	CM01IC1	CM01IF1	CM01MK1	0	0	0	CM01PR12	CM01PR11	CM01PR10
FFFFF182H	CM02IC1	CM02IF1	CM02MK1	0	0	0	CM02PR12	CM02PR11	CM02PR10
FFFFF184H	CM04IC1	CM04IF1	CM04MK1	0	0	0	CM04PR12	CM04PR11	CM04PR10
FFFFF186H	CM05IC1	CM05IF1	CM05MK1	0	0	0	CM05PR12	CM05PR11	CM05PR10
FFFFF188H	CM04IC0	CM04IF0	CM04MK0	0	0	0	CM04PR02	CM04PR01	CM04PR00
FFFFF18AH	CM05IC0	CM05IF0	CM05MK0	0	0	0	CM05PR02	CM05PR01	CM05PR00

Note Reserved for expansion to V850E/IA1.

7.3.5 Interrupt mask registers 0 to 3 (IMR0 to IMR3)

These registers set the interrupt mask state for the maskable interrupts.

The xxMKn bit of the IMR0 to IMR3 registers is equivalent to the xxMKn bit of the xxICn register.

IMRm can be read/written in 16-bit units (m = 0 to 3).

When the IMRm register is divided into two registers: higher 8 bits (IMRmH register) and lower 8 bits (IMRmL register), these registers can be read/written in 8-bit or 1-bit units.

Caution The device file defines the xxMKn bit of the xxICn register as a reserved word. If a bit is manipulated with the name xxMKn, therefore, the xxICn register, rather than the IMRm register, is rewritten (as a result, the IMRm register is also rewritten).

	<15>	<14>	<13>	<12>	<11>	<10>	<9>	<8>	Address	After reset
IMR0	CM10MK0	CC10MK1	CC10MK0	CM03MK1	TM0MK1	CM03MK0	TM0MK0	DETMK1	FFFFF100H	FFFFH
	<7>	6	5	<4>	<3>	<2>	<1>	<0>		
	DETMK0	1	1	P0MK4	P0MK3	P0MK2	P0MK1	P0MK0		

	<15>	<14>	<13>	<12>	<11>	<10>	<9>	<8>	Address	After reset
IMR1	CC3MK1	CC3MK0	TM3MK0	CC2MK5	CC2MK4	CC2MK3	CC2MK2	CC2MK1	FFFFF102H	FFFFH
	<7>	<6>	<5>	4	3	2	1	<0>		
	CC2MK0	TM2MK1	TM2MK0	1	1	1	1	CM10MK1		

	<15>	<14>	<13>	<12>	<11>	<10>	<9>	8	Address	After reset
IMR2	STMK1	SRMK1	SEMK0	STMK0	SRMK0	CSIMK1	CSIMK0	1	FFFFF104H	FFFFH
	7	6	5	<4>	<3>	<2>	<1>	<0>		
	1	1	1	DMAMK3	DMAMK2	DMAMK1	DMAMK0	CM4MK0		

	15	14	<13>	<12>	<11>	<10>	<9>	<8>	Address	After reset
IMR3	1	1	CM05MK0	CM04MK0	CM05MK1	CM04MK1	CM02MK1	CM01MK1	FFFFF106H	FFFFH
	<7>	6	5	4	<3>	<2>	1	0		
	CM00MK1	1	1	1	ADMK1	ADMK0	1	1		

Bit position	Bit name	Function
15 to 7, 4 to 0 (IMR0) 15 to 5, 0 (IMR1) 15 to 9, 4 to 0 (IMR2) 13 to 7, 3, 2 (IMR3)	xxMKn	Interrupt mask flag 0: Interrupt servicing enabled 1: Interrupt servicing disabled (pending)

Remark xx: Identification name of each peripheral unit (refer to **Table 7-2**).
n: Peripheral unit number (refer **Table 7-2**)

7.3.6 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 cleared to 0 by hardware. However, it is not cleared 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.

Caution In the interrupt enabled (EI) state, if an interrupt is acknowledged during the reading of the ISPR register, the value of the ISPR register may be read after the bit is set (1) by this interrupt acknowledgment. To read the value of the ISPR register properly before interrupt acknowledgment, read it in the interrupt disabled (DI) state.

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0	FFFFF1FAH	00H

Bit position	Bit name	Function
7 to 0	ISPR7 to ISPR0	Indicates 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)

7.3.7 Maskable interrupt status flag (ID)

The ID flag is bit 5 of the PSW and this controls the maskable interrupt's operating state, and stores control information regarding enabling or disabling of interrupt requests.

PSW

31																								8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NP	EP	ID	SAT	CY	OV	S	Z

After reset
00000020H

Bit position	Bit name	Function
5	ID	Indicates whether maskable interrupt servicing is enabled or disabled. 0: Maskable interrupt request acknowledgment enabled 1: Maskable interrupt request acknowledgment disabled (pending) 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. The interrupt request generated during the acknowledgment disabled period (ID = 1) is acknowledged when the xxIFn bit of xxICn register is set to 1, and the ID flag is reset to 0.

7.3.8 Interrupt trigger mode selection

The valid edge of the INTPn, ADTRG0, ADTRG1, TIUD10, TCUD10, TCLR10, TCLR3, and TI3 pins can be selected by program. The edge that can be selected as the valid edge is one of the following (n = 0 to 4, 20 to 25, 30, 31, 100, 101).

- Rising edge
- Falling edge
- Both the rising and falling edges

When the INTPn, ADTRG0, ADTRG1, TIUD10, TCUD10, TCLR10, TCLR3, and TI3 signals are edge-detected, they become an interrupt source or capture trigger.

The valid edge is specified by external interrupt mode registers 1 and 2 (INTM1 and INTM2), signal edge selection register 10 (SESA10), the valid edge selection register (SESC), and TM2 input filter mode registers 0 to 5 (FEM0 to FEM5).

(1) External interrupt mode registers 1, 2 (INTM1, INTM2)

These registers specify the valid edge for external interrupt requests (INTP0 to INTP4), input via external pins.

The correspondence between each register and the external interrupt requests that register controls is shown below.

- INTM1: INTP0, INTP1, INTP2/ADTRG0, INTP3/ADTRG1
- INTM2: INTP4

INTP2 and INTP3 function alternately as ADTRG0 and ADTRG1 (A/D converter external trigger input). Therefore, if the external trigger mode has been set by the TRG0 to TRG2 bits of A/D converter mode register n0 (ADSCMn0), setting the ES20 and ES21, and ES30 and ES31 bits of INTM1 also specifies the valid edge of the external trigger input (ADTRG0 and ADTRG1) (n = 0, 1).

The valid edge can be specified independently for each pin (rising edge, falling edge, or both rising and falling edges).

These registers can be read/written in 8-bit or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
INTM1	ES31	ES30	ES21	ES20	ES11	ES10	ES01	ES00	FFFFF882H	00H
	INTP3/ADTRG1		INTP2/ADTRG0		INTP1		INTP0			

	7	6	5	4	3	2	1	0	Address	After reset
INTM2	0	0	0	0	0	0	ES41	ES40	FFFFF884H	00H
							INTP4			

Bit position	Bit name	Function
7 to 0 (INTM1), 1, 0 (INTM2)	ESn1, ESn2 (n = 0 to 4)	Specifies the valid edge of the INTPn, ADTRG0 and ADTRG1 pins.

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

(2) Signal edge selection register 10 (SESA10)

These registers specify the valid edge of external interrupt requests (INTP100, INTP101, TIUD10, TCUD10, and TCLR10), input via external pins.

The valid edge can be specified independently for each pin (rising edge, falling edge, or both rising and falling edges).

These registers can be read/written in 8-bit or 1-bit units.

Cautions 1. The bits of the SESA10 register cannot be changed during TM10 operation (TM1CE0 bit of timer control register 10 (TMC10) = 1).

2. TM1CE0 bit must be set (1) before using the TCUD10/INTP100 and TCLR10/INTP101 pins as INTP100 and INTP101, even if not using timer 1.

3. Setting the trigger mode of the INTP100, INTP101, TIUD10, TCUD10, or TCLR10 pin should be performed after setting the PMC1 register.

If the PMC1 register is set after setting the SESA10 register, an invalid interrupt may occur when the PMC1 register is set.

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
SESA10	TESUD01	TESUD00	CESUD01	CESUD00	IES1011	IES1010	IES1001	IES1000	FFFFF5EDH	00H
	TIUD10, TCUD10		TCLR10		INTP101		INTP100			

Bit position	Bit name	Function
7, 6	TESUD01, TESUD00	Specifies the valid edge of the TIUD10 and TCUD10 pins.

TESUD01	TESUD00	Valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Cautions 1. The values set to the TESUD01 and TESUD00 bits are valid only in UDC mode A[Note 1] and UDC mode B[Note 1].

2. If TM10 operation has been specified in mode 4[Note 2], the valid edge specification (TESUD01 and TESUD00 bits) for the TIUD10 and TCUD10 pins is invalid.

Notes 1. See **9.2.4 (2) Timer unit mode register 0 (TUM0)**.

2. See **9.2.4 (6) Prescaler mode register 10 (PRM10)**.

(2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>5, 4</td><td>CESUD01,
CESUD00</td><td>Specifies the valid edge of the TLCR10 pin

<table><tr><th>CESUD01</th><th>CESUD00</th><th>Valid edge</th></tr><tr><td>0</td><td>0</td><td>Falling edge</td></tr><tr><td>0</td><td>1</td><td>Rising edge</td></tr><tr><td>1</td><td>0</td><td>Low level</td></tr><tr><td>1</td><td>1</td><td>High level</td></tr></table>The setting values of the CESUD01 and CESUD00 bits and the operation of TM10 are as follows.
00: TM10 cleared after detection of TCLR10 rising edge
01: TM10 cleared after detection of TCLR10 falling edge
10: TM10 holds cleared status while TCLR10 input is low level
11: TM10 holds cleared status while TCLR10 input is high level

Caution The values set to the CESUD01 and CESUD00 bits are valid only in UDC mode A^{Note}.</td></tr>
<tr><td>3, 2</td><td>IES1011,
IES1010</td><td>Specifies the valid edge of the pin selected using the CSL0 bit of the CSL10 register (INTP101/INTP100)

<table><tr><th>IES1011</th><th>IES1010</th><th>Valid edge</th></tr><tr><td>0</td><td>0</td><td>Falling edge</td></tr><tr><td>0</td><td>1</td><td>Rising edge</td></tr><tr><td>1</td><td>0</td><td>Setting prohibited</td></tr><tr><td>1</td><td>1</td><td>Both rising and falling edges</td></tr></table></td></tr>
<tr><td>1, 0</td><td>IES1001,
IES1000</td><td>Specifies the valid edge of the INTP100 pin

<table><tr><th>IES1001</th><th>IES1000</th><th>Valid edge</th></tr><tr><td>0</td><td>0</td><td>Falling edge</td></tr><tr><td>0</td><td>1</td><td>Rising edge</td></tr><tr><td>1</td><td>0</td><td>Setting prohibited</td></tr><tr><td>1</td><td>1</td><td>Both rising and falling edges</td></tr></table></td></tr>
</table>

Note See **9.2.4 (2) Timer unit mode register 0 (TUM0)**.

(3) Valid edge selection register (SESC)

This register specifies the valid edge for external interrupt requests (INTP30, INTP31, TCLR3, TI3), input via external pins.

The valid edge can be specified independently for each pin (rising edge, falling edge, or both rising and falling edges).

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

Cautions 1. The TM3CAE and TM3CE bits of timer control register 30 (TMC30) must be set (1) before using the TI3/TCLR3/INTP30 and TO3/INTP31 pins as INTP30 and INTP31, even if not using timer 3.

2. Setting the trigger mode of the INTP30, INTP31, TCLR3, or TI3 pin should be performed after setting the PMC2 register.

If the PMC2 register is set after setting the SESC register, an invalid interrupt may occur when the PMC2 register is set.

	7	6	5	4	3	2	1	0	Address	After reset
SESC	TES31	TES30	CES31	CES30	IES311	IES310	IES301	IES300	FFFFF689H	00H
	TI3		TCLR3		INTP31		INTP30			

Bit position	Bit name	Function
7, 6	TES31, TES30	Specifies the valid edge of the INTP30, INTP31, TCLR3, or TI3 pins.
5, 4	CES31, CES30	
3, 2	IES311, IES310	
1, 0	TES301, TES300	

xESn1	xESn0	Valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Remark n = 3, 30, 31

(4) Timer 2 input filter mode registers 0 to 5 (FEM0 to FEM5)

These registers specify the valid edge for external interrupts input to timer 2 (INTP20 to INTP25). The correspondence between each register and the external interrupt request that register controls is shown below.

- FEM0: INTP20
- FEM1: INTP21
- FEM2: INTP22
- FEM3: INTP23
- FEM4: INTP24
- FEM5: INTP25

The valid edge can be specified independently for each pin (rising edge, falling edge, or both rising and falling edges).

These registers can be read/written in 8-bit or 1-bit units.

Cautions 1. Be sure to clear (0) the STFTE bit of timer 2 clock stop register 0 (STOPTE0) even when using the TI2/INTP20, TO21/INTP21, TO22/INTP22, TO23/INTP23, TO24/INTP24, and TCLR2/INTP25 pins as INTP20, INTP21, INTP22, INTP23, INTP24, and INTP25, respectively, even if not using timer 2.

2. Setting the trigger mode of the INTP2n pin should be performed after setting the PMC2 register.

If the PMC2 register is set after setting the FEMn register, an invalid interrupt may occur when the PMC2 register is set (n = 0 to 5).

★ **3. The noise elimination function starts operating by setting the CEEn bit of the TCRE0 register to 1 (enabling count operations).**

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
FEM0	DFEN00	0	0	0	EDGE010	EDGE000	TMS010	TMS000	FFFFF630H	00H
					INTP20					
FEM1	DFEN01	0	0	0	EDGE011	EDGE001	TMS011	TMS001	FFFFF631H	00H
					INTP21					
FEM2	DFEN02	0	0	0	EDGE012	EDGE002	TMS012	TMS002	FFFFF632H	00H
					INTP22					
FEM3	DFEN03	0	0	0	EDGE013	EDGE003	TMS013	TMS003	FFFFF633H	00H
					INTP23					
FEM4	DFEN04	0	0	0	EDGE014	EDGE004	TMS014	TMS004	FFFFF634H	00H
					INTP24					
FEM5	DFEN05	0	0	0	EDGE015	EDGE005	TMS015	TMS005	FFFFF635H	00H
					INTP25					

Bit position	Bit name	Function
7	DFEN0n	Specifies the filter of the INTP2n pin. 0: Analog filter 1: Digital filter **Caution When the DFEN0n bit = 1, the sampling clock of the digital filter is f_{XTM2} (clock selected by the PRM02 register).**
3, 2	EDGE01n, EDGE00n	Specifies the valid edge of the INTP2n pin.

EDGE01n	EDGE00n	Operation
0	0	Interrupt by INTCC2n[Note]
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Note Set when INTCC2n is selected by a match between TM20, TM21 and the subchannel compare register (specified by the TMS01n, TMS00n bits) (n = 0 to 5).

Remark n = 0 to 5

(2/2)

Bit position	Bit name	Function
1, 0	TMS01n, TMS00n	Selects the capture input[Note].

TMS01n	TMS00n	Operation
0	0	Used as a pin
0	1	Digital filter (noise eliminator specification)
1	0	Timer-based capture to subchannel 1
1	1	Timer-based capture to subchannel 2

Note Selection of capture input based on INTCM100 and INTCM101 is valid only for the FEM1 and FEM2 registers. Set the TMS01m and TMS00m bits of the FEMm register to 00B or 01B. All other settings are prohibited (m = 1, 3 to 5).

Subchannels 1 and 2 of timer 2 can be captured by INTP21, INTP22, and INTCM100, INTCM101.

An example is given below.

(a) When subchannel 1 is captured by INTCM101

FEM1 register = xxxxxx10B

TMIC0 register = 00000010B

(b) When subchannel 2 is captured by INTCM101

FEM2 register = xxxxxx11B

TMIC0 register = 00001000B

Remark n = 0 to 5

7.4 Software Exception

A software exception is generated when the CPU executes the TRAP instruction, and can be always acknowledged.

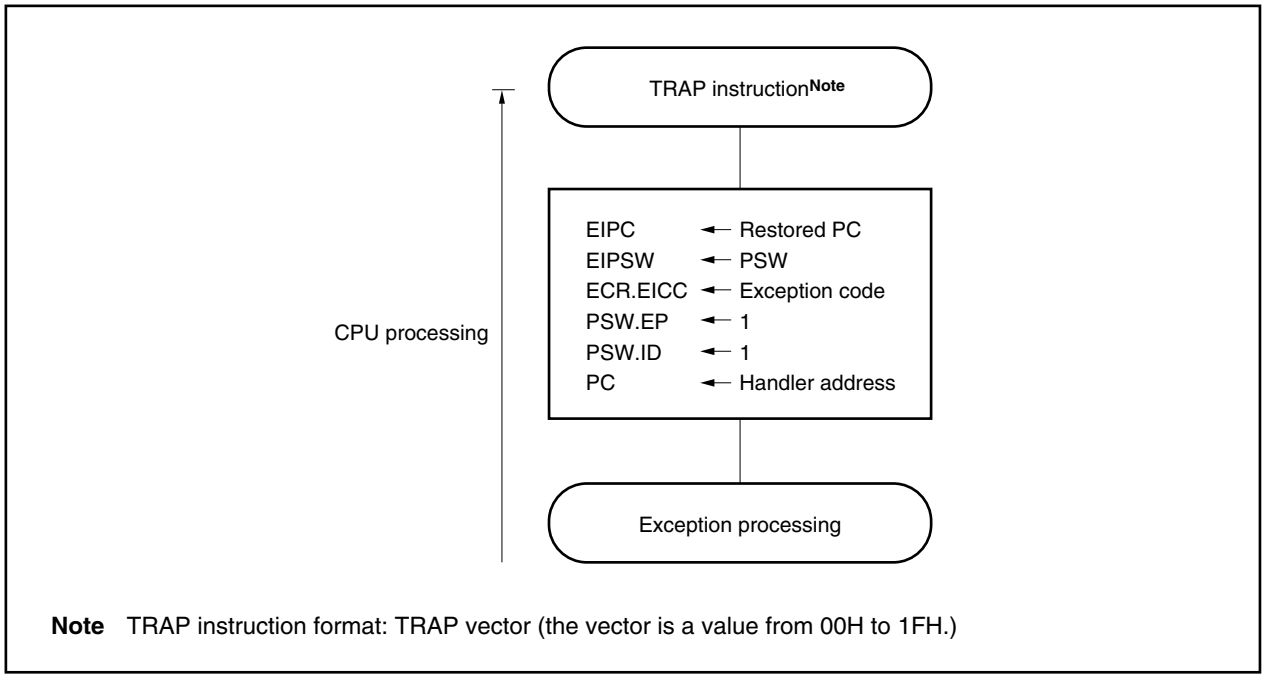
7.4.1 Operation

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 EP and ID bits of the PSW.
(5) Sets the handler address (00000040H or 00000050H) corresponding to the software exception to the PC, and transfers control.

Figure 7-8 illustrates the processing of a software exception.

Figure 7-8. 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.

7.4.2 Restore

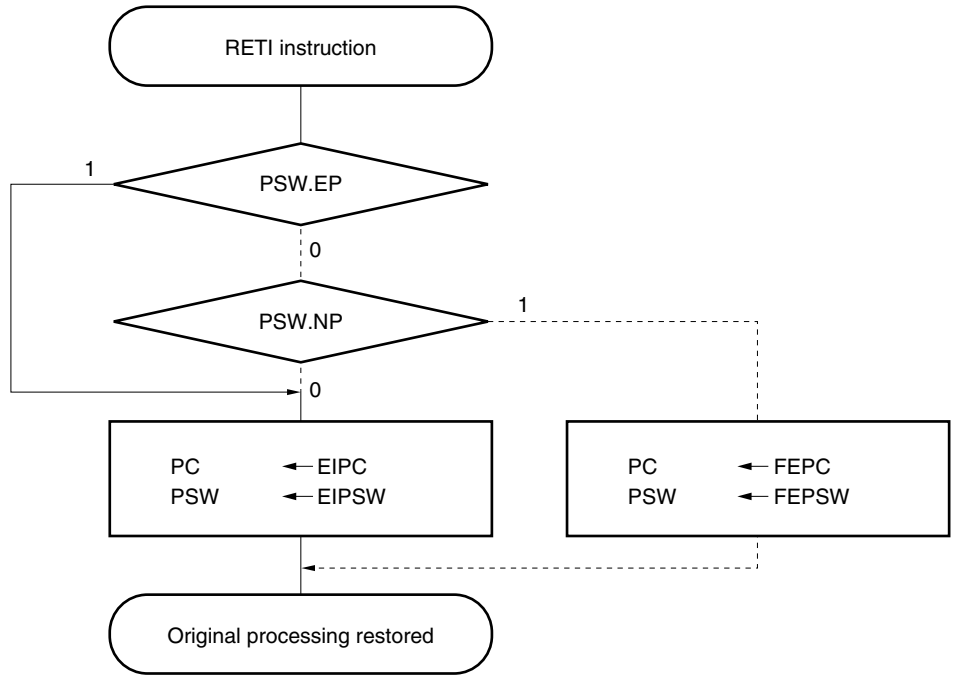
Returning from software exception processing is carried out by the RETI instruction.

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 EP bit of the PSW is 1.
(2) Transfers control to the address of the restored PC and PSW.

Figure 7-9 illustrates the processing of the RETI instruction.

Figure 7-9. 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 lines show the CPU processing flow.

7.4.3 Exception status flag (EP)

The EP flag is bit 6 of PSW, and is a status flag used to indicate that exception processing is in progress. It is set when an exception occurs.

PSW

31																								8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NP	EP	ID	SAT	CY	OV	S	Z

After reset
00000020H

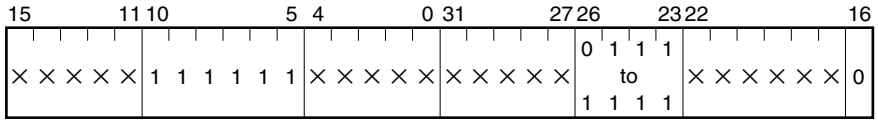
Bit position	Bit name	Function
6	EP	Shows that exception processing is in progress. 0: Exception processing not in progress. 1: Exception processing in progress.

7.5 Exception Trap

An exception trap is an interrupt that is requested when an illegal execution of an instruction takes place. In the V850E/IA2, an illegal opcode exception (ILGOP: Illegal Opcode Trap) is considered as an exception trap.

7.5.1 Illegal opcode definition

The illegal instruction has an opcode (bits 10 to 5) of 111111B, sub-opcodes of 0111B to 1111B (bits 26 to 23), and 0B (bit 16). An exception trap is generated when an instruction applicable to this illegal instruction is executed.



×: Arbitrary

Caution Since it is possible that this instruction will be assigned to an illegal opcode in the future, it is recommended that it not be used.

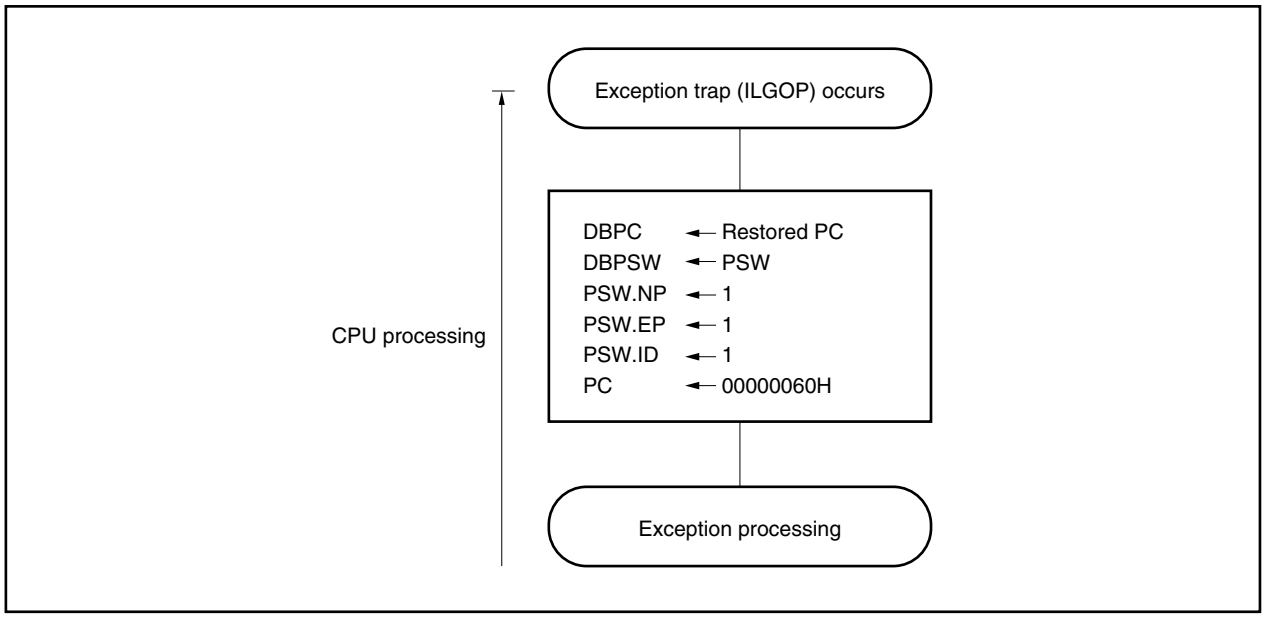
(1) Operation

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 NP, EP, and ID bits of the PSW.
(4) Sets the handler address (00000060H) corresponding to the exception trap to the PC, and transfers control.

Figure 7-10 illustrates the processing of the exception trap.

Figure 7-10. Exception Trap Processing



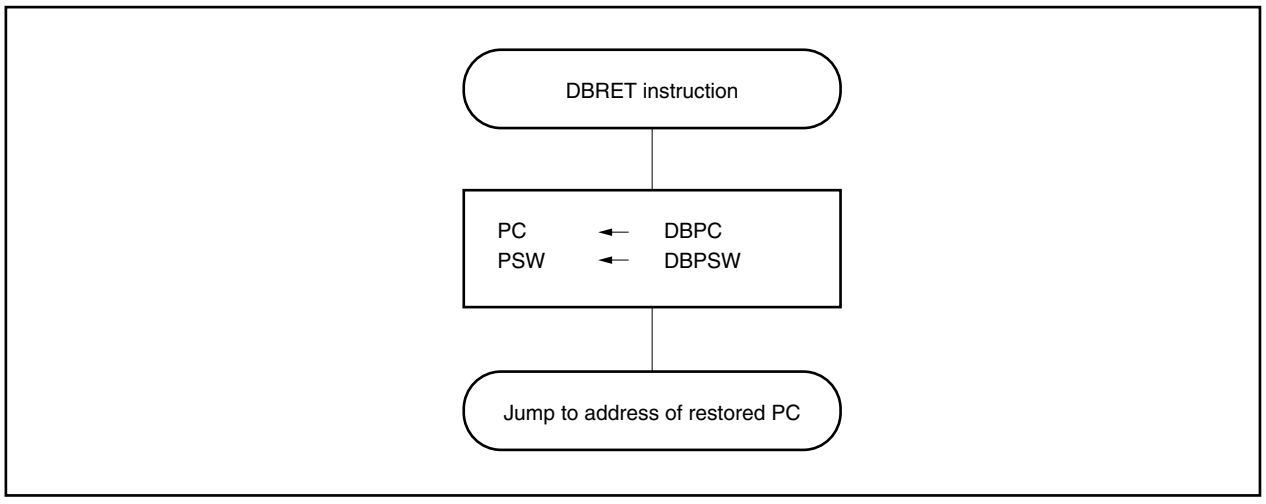
(2) Restore

Returning from exception trap processing 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 7-11 illustrates the processing for restoring from an exception trap.

Figure 7-11. Processing for Restoring from Exception Trap



7.5.2 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.

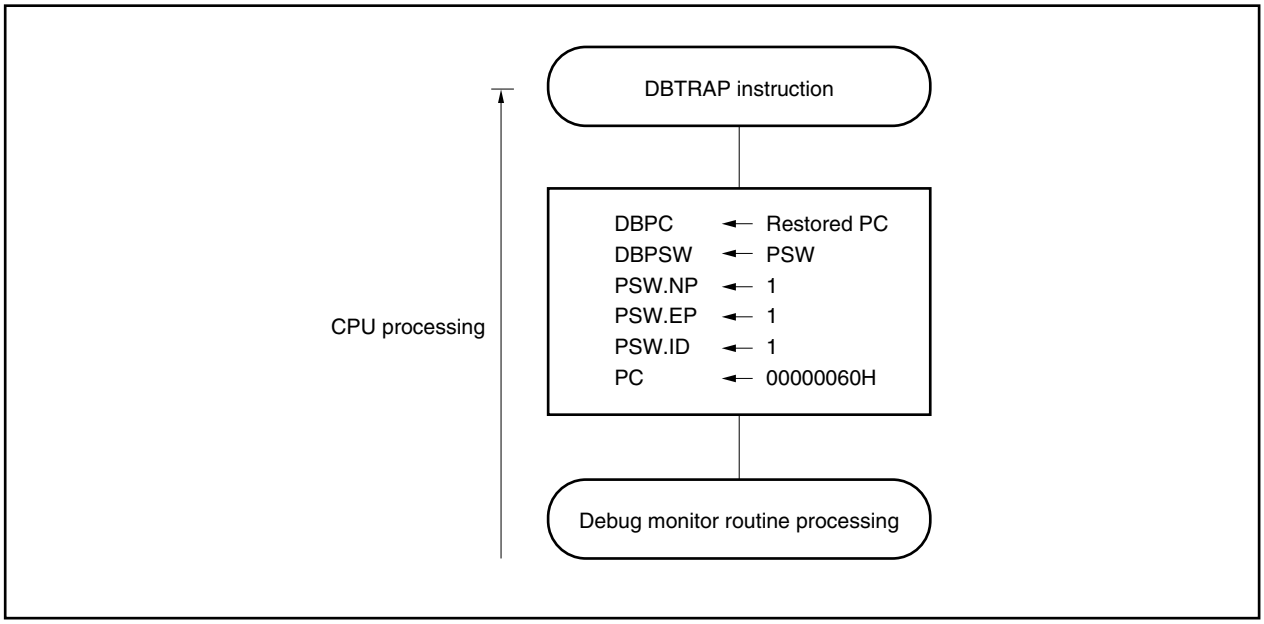
(1) Operation

When the debug trap is generated, the CPU performs the following processing, transfers control to the debug monitor routine, and shifts to debug mode.

(1) Saves the restored PC to DBPC.
(2) Saves the current PSW to DBPSW.
(3) Sets the NP, EP and ID bits of the PSW.
(4) Sets the handler address (00000060H) corresponding to the debug trap to the PC and transfers control.

Figure 7-12 illustrates the processing of the debug trap.

Figure 7-12. Debug Trap Processing



(2) Restore

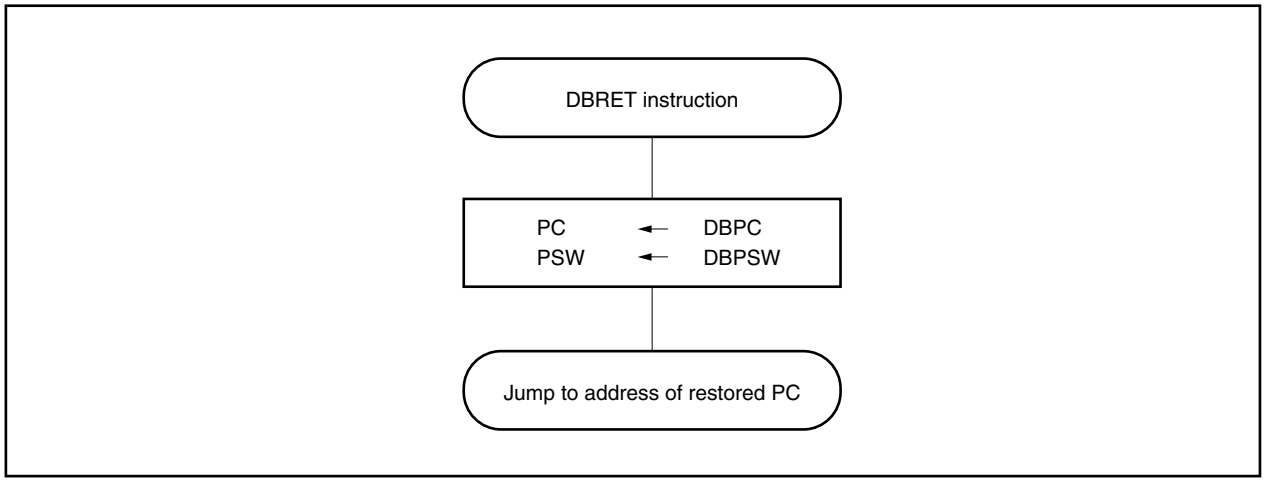
Returning from debug trap processing is carried out by the DBRET instruction. By executing the DBRET instruction, the CPU carries out the following processing and controls the address of the restored PC.

(1) Loads the restored PC and PSW from DBPC and DBPSW.
(2) Transfers control to the address indicated by the restored PC and PSW.

★ **Caution DBPC and DBPSW can be accessed during the period between when the DBTRAP is executed and when the DBRET instruction is executed.**

Figure 7-13 illustrates the processing for restoring from a debug trap.

Figure 7-13. Processing for Restoring from Debug Trap



7.6 Multiple Interrupt Servicing Control

Multiple interrupt servicing control is a process by which an interrupt request that is currently being processed can be interrupted during processing if there is an interrupt request with a higher priority level, and the higher priority interrupt request is received and processed first.

If there is an interrupt request with a lower priority level than the interrupt request currently being processed, that interrupt request is held pending.

Maskable interrupt multiple processing control is executed when interrupts are enabled (ID = 0). Thus, if multiple interrupts are executed, it is necessary for interrupts to be enabled (ID = 0) even during an interrupt servicing routine.

If a maskable interrupt or a software exception is generated in a maskable interrupt or software exception service program, it is necessary to save EIPC and EIPSW.

This is accomplished by the following procedure.

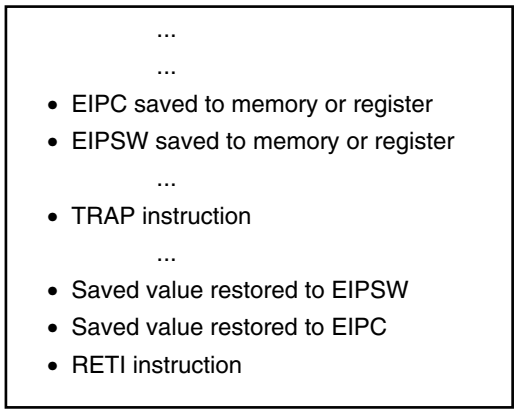
(1) Acknowledgment of maskable interrupts in service program

Service program of maskable interrupt or exception

```
        ...
        ...
• EIPC saved to memory or register
• EIPSW saved to memory or register
• EI instruction (interrupt acknowledgment enabled)
        ...
        ...                          ← Maskable interrupt acknowledgment
        ...
        ...
• DI instruction (interrupt acknowledgment disabled)
• Saved value restored to EIPSW
• Saved value restored to EIPC
• RETI instruction
```

(2) Generation of exception in service program

Service program of maskable interrupt or exception



The priority order for multiple interrupt servicing control has 8 levels, from 0 to 7 for each maskable interrupt request (0 is the highest priority), but it can be set as desired via software. Setting of the priority order level is done using the xxPRn0 to xxPRn2 bits of the interrupt control request register (xxICn), which is provided for each maskable interrupt request. After system reset, an interrupt request is masked by the xxMKn bit and the priority order is set to level 7 by the xxPRn0 to xxPRn2 bits.

The priority order of maskable interrupts is as follows.

(High) Level 0 > Level 1 > Level 2 > Level 3 > Level 4 > Level 5 > Level 6 > Level 7 (Low)

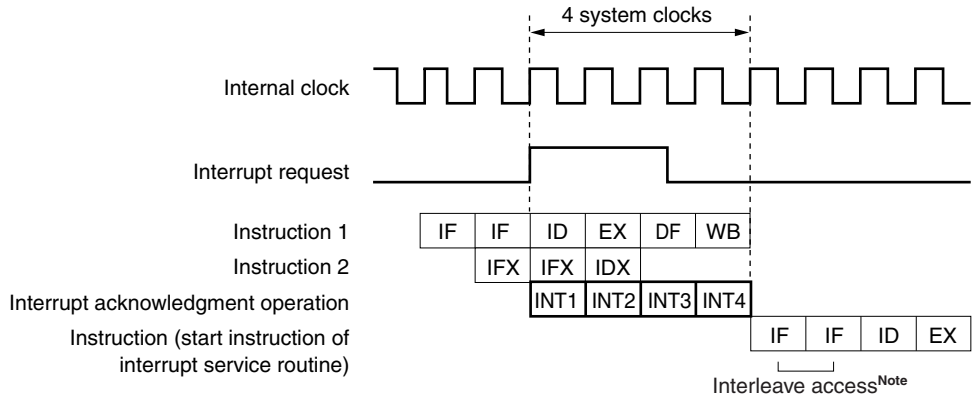
Interrupt servicing that has been suspended as a result of multiple servicing control is resumed after the servicing of the higher priority interrupt has been completed and the RETI instruction has been executed.
A pending interrupt request is acknowledged after the current interrupt servicing has been completed and the RETI instruction has been executed.

Caution In a non-maskable interrupt servicing routine (time until the RETI instruction is executed), maskable interrupts are suspended and not acknowledged.

7.7 Interrupt Response Time

The following table describes the V850E/IA2 interrupt response time (from interrupt generation to start of interrupt servicing).

Figure 7-14. Pipeline Operation at Interrupt Request Acknowledgment (Outline)



Note For details of interleave access, refer to **8.1.2 2-clock branch** in **V850E1 Architecture User's Manual (U14559E)**.

Remark INT1 to INT4: Interrupt acknowledgment processing
IFX: Invalid instruction fetch
IDX: Invalid instruction decode

Interrupt Response Time (Internal System Clock (f_{XX}))					Condition
	Internal Interrupt	External Interrupt			
		INTP0 to INTP4, INTP20 to INTP25	INTP20 to INTP25	INTP100, INTP30, INTP101, INTP31	
Minimum	4	4+ analog delay time	4+ digital noise filter	4 + **Note 1** + digital noise filter	The following cases are exceptions. • In IDLE/software STOP mode • External bus access • Two or more interrupt request non-sampling instructions are executed in succession • Access to on-chip peripheral I/O register
Maximum	7 Note 2	7+ analog delay time	7+ digital noise filter	7 + **Note 1** + digital noise filter	

Notes 1. The number of internal system clocks is as follows.

- For timer 10 (TM10) using INTP100 and INTP101 as external interrupt inputs (see **9.2.4 (1) Timer 1/timer 2 clock selection register (PRM02)**):
  $f_{CLK} = f_{XX}/2$ (PRM2 bit = 1): 2
  $f_{CLK} = f_{XX}/4$ (PRM2 bit = 0): 4
- For timer 3 (TM3) using INTP30 and INTP31 as external interrupt inputs (see **9.4.5 (1) Timer 3 clock selection register (PRM03)**):
  $f_{CLK} = f_{XX}$ (PRM3 bit = 1): 2
  $f_{CLK} = f_{XX}/2$ (PRM3 bit = 0): 4

2. When LD instruction is executed to internal ROM (during align access)

7.8 Periods in Which CPU Does Not Acknowledge Interrupts

The CPU acknowledges an interrupt while an instruction is being executed. However, no interrupt will be acknowledged between an interrupt non-sample instruction and the next instruction (interrupt is held pending).

The interrupt request non-sampling instructions are as follows.

- EI instruction
- DI instruction
- LDSR reg2, 0x5 instruction (for PSW)
- The store instruction for the command register (PRCMD)
- ★ • The store instructions or bit manipulation instructions of SET1, CLR1, and NOT1 instructions for the following registers:
  - Interrupt-related registers:
    Interrupt control register (xxICn), interrupt mask registers 0 to 3 (IMR0 to IMR3)
  - Power save control register (PSC)
  - CSI-related registers:
    Clocked serial interface mode registers 0, 1 (CSIM0, CSIM1)
    Clocked serial interface clock selection registers 0, 1 (CSIC0, CSIC1)
    Clocked serial interface receive buffer registers 0, 1 (SIRB0, SIRB1)
    Clocked serial interface receive buffer registers L0, L1 (SIRBL0, SIRBL1)
    Clocked serial interface transmit buffer registers 0, 1 (SOTB0, SOTB1)
    Clocked serial interface transmit buffer registers L0, L1 (SOTBL0, SOTBL1)
    Clocked serial interface read-only receive buffer registers 0, 1 (SIRBE0, SIRBE1)
    Clocked serial interface read-only receive buffer registers L0, L1 (SIRBEL0, SIRBEL1)
    Clocked serial interface initial transmit buffer registers 0, 1 (SOTBF0, SOTBF1)
    Clocked serial interface initial transmit buffer registers L0, L1 (SOTBFL0, SOTBFL1)
    Serial I/O shift registers 0, 1 (SIO0, SIO1)
    Serial I/O shift registers L0, L1 (SIOL0, SIOL1)
    Prescaler mode register (PRSM3)
    Prescaler compare register (PRSCM3)

Remark xx: Identification name of each peripheral unit (refer to **Table 7-2**)
n: Peripheral unit number (refer to **Table 7-2**)

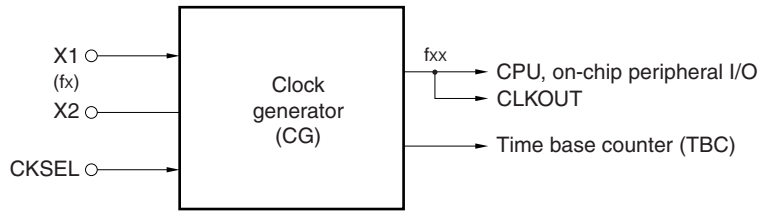
CHAPTER 8 CLOCK GENERATION FUNCTION

The clock generator (CG) generates and controls the internal system clock (fxx) that is supplied to each internal unit, such as the CPU.

8.1 Features

- Multiplier function using a phase locked loop (PLL) synthesizer
- Clock sources
  - Oscillation by connecting a resonator
  - External clock
- Power-saving modes
  - HALT mode
  - IDLE mode
  - Software STOP mode
- Internal system clock output function

8.2 Configuration



Remark fx: External resonator or external clock frequency
fxx: Internal system clock

8.3 Input Clock Selection

The clock generator consists of an oscillator and a PLL synthesizer. For example, connecting a 4.0 MHz crystal resonator or ceramic resonator to the X1 and X2 pins enables a 40 MHz internal system clock (f_{XX}) to be generated when the multiplier is 10. Also, an external clock can be input directly to the oscillator. In this case, the clock signal should be input only to the X1 pin (the X2 pin should be left open). Two basic operation modes are provided for the clock generator. These are the PLL mode and the direct mode. The operation mode is selected by the CKSEL pin. The input to this pin is latched on reset.

CKSEL	Operation Mode
0	PLL mode
1	Direct mode

Caution The input level for the CKSEL pin must be fixed. If it is switched during operation, a malfunction may occur.

8.3.1 Direct mode

In the direct mode, the external clock is divided by two and the divided clock is supplied as the internal system clock. The maximum frequency that can be input in the direct mode is 50 MHz. This mode is used in application system where the V850E/IA2 operates at relatively low frequencies.

Caution In direct mode, an external clock must be input (an external resonator should not be connected).

8.3.2 PLL mode

In PLL mode, an external resonator is connected or external clock is input and multiplied by the PLL synthesizer. The multiplied PLL output is divided by the division ratio specified by the clock control register (CKC) to generate a system clock that is 10, 5, 2.5, or 1 times the frequency (f_X) of the external resonator or external clock.

After reset, an internal system clock (f_{XX}) that is 1 time the frequency ($1 \times f_X$) of the internal clock frequency (f_X) is generated.

When a frequency that is 10 times the clock frequency (f_X) ($10 \times f_X$) is generated, a system with low noise and low power consumption can be realized because a frequency of up to 40 MHz is obtained based on a 4 MHz external resonator or external clock.

In PLL mode, if the clock supply from an external resonator or external clock source stops, operation of the internal system clock (f_{XX}) based on the self-propelled frequency of the clock generator's internal voltage controlled oscillator (VCO) continues. In this case, f_{XX} is undefined. However, do not devise an application method expecting to use this self-propelled frequency.

Example: Clocks when PLL mode ($f_{XX} = 10 \times f_X$) is used

Internal System Clock Frequency (f_{XX})	External Resonator or External Clock Frequency (f_X)
40.000 MHz	4.0000 MHz

Caution Only an fx value for which 10 × fx does not exceed the system clock maximum frequency (40 MHz) (i.e. 4 MHz) can be used for the oscillation frequency or external clock frequency. When 5 × fx, 2.5 × fx, or 1 × fx is used, a frequency of 4 to 6.4 MHz can be used.

Remark Note the following when PLL mode is selected ($f_{XX} = 5 \times f_X$, $f_{XX} = 2.5 \times f_X$, or $f_{XX} = 1 \times f_X$)
If the V850E/IA2 does not need to be operated at a high frequency, use $f_{XX} = 5 \times f_X$, $f_{XX} = 2.5 \times f_X$, or $f_{XX} = 1 \times f_X$ to reduce the power consumption by lowering the system clock frequency using software.

8.3.3 Peripheral command register (PHCMD)

This is an 8-bit register that is used to set protection for writing to registers that can significantly affect the system so that the application system is not halted unexpectedly due to erroneous program execution. This register is write-only in 8-bit units (when it is read, undefined data is read out).

★ Writing to the first specific register (CKC or FLPMC register) is only valid after first writing to the PHCMD register. Because of this, the register value can be overwritten only in the specified sequence, preventing an illegal write operation from being performed.

	7	6	5	4	3	2	1	0	Address	After reset
PHCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0	FFFFF800H	Undefined

Bit position	Bit name	Function
7 to 0	REG7 to REG0	Registration code (arbitrary 8-bit data) The specific registers targeted are as follows. • Clock control register (CKC) ★ • Flash programming mode control register (FLPMC)

The generation of an illegal store operation can be checked with the PRERR bit of the peripheral status register (PHS).

8.3.4 Clock control register (CKC)

The clock control register is an 8-bit register that controls the internal system clock (f_{XX}) in PLL mode. It can be written to only by a specific sequence combination so that it cannot easily be overwritten by mistake due to erroneous program execution.

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

Caution Do not change the CKDIV2 to CKDIV0 bits in direct mode.

	7	6	5	4	3	2	1	0	Address	After reset
CKC	0	0	TBCS	CESEL	0	CKDIV2	CKDIV1	CKDIV0	FFFFF822H	00H

Bit position	Bit name	Function
5	TBCS	Selects the time base counter clock. 0: $f_X/2^8$ 1: $f_X/2^9$ For details, see **8.6.2 Time base counter (TBC)**.
4	CESEL	Specifies the functions of the X1 and X2 pins. 0: A resonator is connected to the X1 and X2 pins 1: An external clock is connected to the X1 pin When CESEL = 1, the oscillator feedback loop is disconnected to prevent current leakage in software STOP mode.
2 to 0	CKDIV2 to CKDIV0	Sets the internal system clock frequency (f_{XX}) when PLL mode is used.

CKDIV2	CKDIV1	CKDIV0	Internal system clock (f_{XX})
0	0	0	f_X
0	0	1	$2.5 \times f_X$
0	1	1	$5 \times f_X$
1	1	1	$10 \times f_X$
Other than above			Setting prohibited

Caution When changing the internal system clock during operation, be sure to set the clock to be changed after setting the CKDIV2 to CKDIV0 bits to 000 (f_X).

Example Clock generator settings

Operation Mode	CKSEL Pin	CKC Register			Input Clock (f_X)	Internal System Clock (f_{XX})
		CKDIV2	CKDIV0	CKDIV0		
Direct mode	High-level input	0	0	0	16 MHz	8 MHz
PLL mode	Low-level input	0	0	0	4 MHz	4 MHz
		0	0	1	5 MHz	12.5 MHz
		0	1	1	6.4 MHz	32 MHz
		1	1	1	4 MHz	40 MHz
Other than above					Setting prohibited	Setting prohibited

Data is set in the clock control register (CKC) according to the following sequence.

<1> Disable interrupts (set the NP bit of PSW to 1)
<2> Prepare data in any one of the general-purpose registers to set in the specific register.
<3> Write arbitrary data to the peripheral command register (PHCMD)
<4> Set the clock control register (CKC) (with the following instructions).
- Store instruction (ST/SST instruction)

<5> Insert five or more NOP instructions (5 instructions (<5> to <9>))
<10> Release the interrupt disabled state (set the NP bit of PSW to 0).

[Sample coding]

```
<1>  LDSR  rX, 5
<2>  MOV   0X04, r10
<3>  ST.B  r10, PHCMD [r0]
<4>  ST.B  r10, CKC [r0]
<5>  NOP
<6>  NOP
<7>  NOP
<8>  NOP
<9>  NOP
<10> LDSR  rY, 5
```

Remark rX: Value written to PSW
rY: Value returned to PSW

No special sequence is required to read the specific register.

Cautions 1. If an interrupt is acknowledged between the issuing of data to PHCMD <3> and writing to the specific register immediately after <4>, the write operation to the specific register is not performed and a protection error (the PRERR bit of the PHS register = 1) may occur. Therefore, set the NP bit of the PSW to 1 <1> to disable interrupt acknowledgment. Also disable interrupt acknowledgment when selecting a bit manipulation instruction for the specific register setting.

2. Although the data written to the PHCMD register is dummy data, use the same register as the general-purpose register used in specific register setting <4> for writing to the PHCMD register (<3>). The same method should be applied when using a general-purpose register for addressing.

3. Before executing this processing, complete all DMA transfer operations.

8.3.5 Peripheral status register (PHS)

If a write operation is not performed in the correct sequence including access to the command register for the protection-targeted internal registers, writing is not performed and a protection error is generated, setting the status flag (PRERR) to 1. This flag is a cumulative flag. After checking the PRERR flag, it is cleared to 0 by an instruction.

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

	7	6	5	4	3	2	1	<0>	Address	After reset
PHS	0	0	0	0	0	0	0	PRERR	FFFFF802H	00H

Bit position	Bit name	Function
0	PRERR	0: Protection error does not occur 1: Protection error occurs

The operation conditions of the PRERR flag are as follows.

Set conditions: <1> If the operation of the relevant store instruction for the on-chip peripheral I/O is not a write operation for the PHCMD register, but the peripheral specific register is written to.

<2> If the first store instruction operation after the write operation to the PHCMD register is for memory other than the specific registers and on-chip peripheral I/O.

Reset conditions: <1> If the PRERR flag of the PHS register is set to 0.

<2> If the system is reset

8.4 PLL Lockup

The lockup time (frequency stabilization time) is the time from when the power is turned on or the software STOP mode is released until the phase locks at the prescribed frequency. The state until this stabilization occurs is called a lockup state, and the stabilized state is called a lock state.

The lock register (LOCKR) has a LOCK flag that reflects the stabilized state of the PLL frequency.

This register is read-only, in 8-bit or 1-bit units.

Caution When the PLL is locked, the LOCK flag is 0. If the system then enters an unlocked state due to a standby, the LOCK flag becomes 1. If anything other than a standby causes the system to enter an unlocked state, the LOCK flag is not affected (LOCK = 0).

	7	6	5	4	3	2	1	<0>	Address	After reset
LOCKR	0	0	0	0	0	0	0	LOCK	FFFFF824H	0000000xB

Bit position	Bit name	Function
0	LOCK	This is a read-only flag that indicates the PLL state. This flag holds the value 0 as long as a lockup state is maintained and is not initialized by a system reset. 0: Indicates that the PLL is locked. 1: Indicates that the PLL is not locked (UNLOCK state).

If the clock stops, the power fails, or some other factor operates to cause an unlock state to occur, for control processing that depends on software execution speed, such as real-time processing, be sure to judge the LOCK flag using software immediately after operation begins so that processing does not begin until after the clock stabilizes.

On the other hand, static processing such as the setting of internal hardware or the initialization of register data or memory data can be executed without waiting for the LOCK flag to be reset.

The relationship between the oscillation stabilization time (the time from when the resonator starts to oscillate until the input waveform stabilizes) when a resonator is used, and the PLL lockup time (the time until frequency stabilizes) is shown below.

Oscillation stabilization time < PLL lockup time

8.5 Power Save Control

8.5.1 Overview

The power save function has the following three modes.

(1) HALT mode

In this mode, the clock generator (oscillator and PLL synthesizer) continues to operate, but the CPU's operation clock stops. Since the supply of clocks to on-chip peripheral functions other than the CPU continues, operation continues. The power consumption of the overall system can be reduced by intermittent operation that is achieved due to a combination of HALT mode and normal operation mode.

The system is switched to HALT mode by a specific instruction (the HALT instruction).

(2) IDLE mode

In this mode, the clock generator (oscillator and PLL synthesizer) continues to operate, but the supply of internal system clocks is stopped, which causes the overall system to stop.

When the system is released from IDLE mode, it can be switched to normal operation mode quickly because the oscillator's oscillation stabilization time need not be secured.

The system is switched to IDLE mode according to a PSMR register setting.

IDLE mode is located midway between software STOP mode and HALT mode in relation to the clock stabilization time and current consumption. It is used for situations in which a low current consumption mode is to be used and the clock stabilization time is to be eliminated after the mode is released.

(3) Software STOP mode

In this mode, the overall system is stopped by stopping the clock generator (oscillator and PLL synthesizer). The system enters an ultra-low power consumption state in which only leak current is lost.

The system is switched to software STOP mode according to a PSMR register setting.

(a) PLL mode

The system is switched to software STOP mode by setting the register by software. The PLL synthesizer's clock output is stopped at the same time that the oscillator is stopped. After software STOP mode is released, the oscillator's oscillation stabilization time must be secured while the system clock stabilizes. Also, PLL lockup time may be required depending on the program. When a resonator or external clock is connected, following the release of the software STOP mode, execution of the program is started after the count time of the time base counter has elapsed.

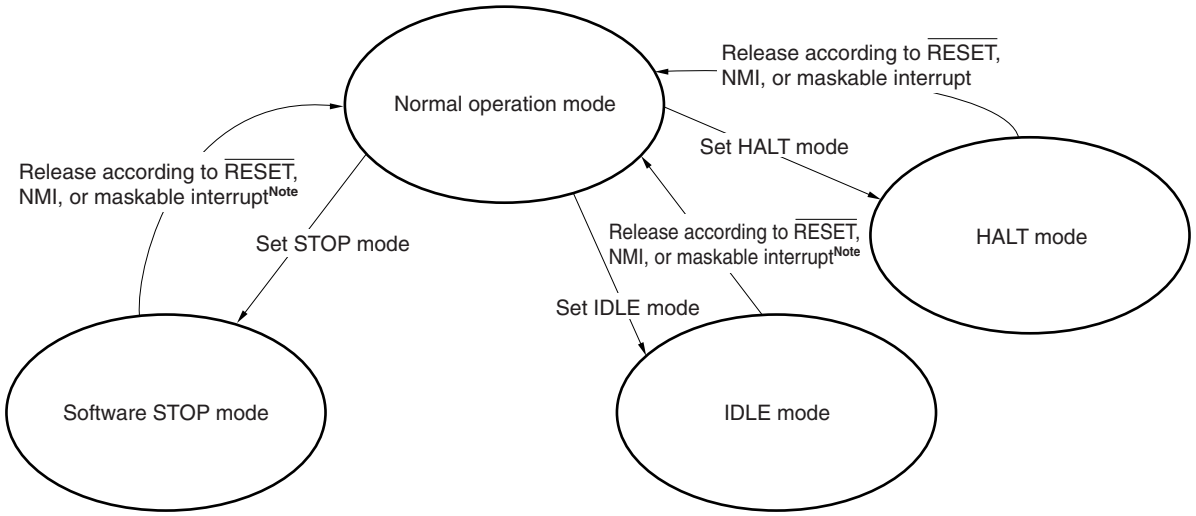
(b) Direct mode

To stop the clock, set the X1 pin to low level. After the release of software STOP mode, execution of the program is started after the count-time of the time base counter has elapsed.

Figure 8-1 shows the operation of the clock generator in normal operation mode, HALT mode, IDLE mode, and software STOP mode.

An effective low power consumption system can be realized by combining these modes and switching modes according to the required use.

Figure 8-1. Power Save Mode State Transition Diagram



Note INTPn (n = 0 to 4, 20 to 25)
However, in cases such as when a digital filter using clock sampling is selected as the noise eliminator for INTP20 to INTP25, the software STOP or IDLE mode cannot be released.

Table 8-1. Clock Generator Operation Using Power Save Control

Clock Source		Power Save Mode	Oscillator	PLL Synthesizer	Clock Supply to Peripheral I/O	Clock Supply to CPU
PLL mode	Oscillation with resonator	Normal operation	√	√	√	√
		HALT mode	√	√	√	–
		IDLE mode	√	√	–	–
		Software STOP mode	–	–	–	–
	External clock	Normal operation	–	√	√	√
		HALT mode	–	√	√	–
		IDLE mode	–	√	–	–
		Software STOP mode	–	–	–	–
Direct mode	External clock	Normal operation	–	–	√	√
		HALT mode	–	–	√	–
		IDLE mode	–	–	–	–
		Software STOP mode	–	–	–	–

Remark √: Operating
–: Stopped

8.5.2 Control registers

(1) Power save mode register (PSMR)

This is an 8-bit register that controls the power save mode. It is effective only when the STB bit of the PSC register is set to 1.

Writing to the PSMR is executed by store instructions (ST/SST instruction) and bit manipulation instructions (SET1/CLR1/NOT1 instruction).

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

	7	6	5	4	3	2	1	<0>	Address	After reset
PSMR	0	0	0	0	0	0	0	PSM	FFFFF820H	00H

Bit position	Bit name	Function
0	PSM	Specifies IDLE mode or software STOP mode. 0: Switches the system to IDLE mode 1: Switches the system to software STOP mode

(2) Command register (PRCMD)

This is an 8-bit register that is used to set protection for write operations to registers that can significantly affect the system so that the application system is not halted unexpectedly due to erroneous program execution.

Writing to the first specific register (power save control register (PSC)) is only valid after first writing to the PRCMD register. Because of this, the register value can be overwritten only by the specified sequence, preventing an illegal write operation from being performed.

This register is write-only in 8-bit units. Undefined data is read out if read.

	7	6	5	4	3	2	1	0	Address	After reset
PRCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0	FFFFF1FCH	Undefined

Bit position	Bit name	Function
7 to 0	REG7 to REG0	Registration code (arbitrary 8-bit data) The specific register targeted is the power save control register (PSC).

(3) Power save control register (PSC)

This is an 8-bit register that controls the power save function.

★ If releasing of interrupts are enabled by the setting of the NMIM and INTM bits, the software STOP mode can be released by an interrupt request (except when interrupt servicing is disabled by the interrupt mask registers (IMR0 to IMR3)).

★ The software STOP mode is specified by the setting of the STB bit.

This register, which is one of the specific registers, is effective only when accessed by a specific sequence during a write operation (see **3.4.9 Specific registers**).

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

★ Be sure to clear bits 7 and 6 to 0. If they are set to 1, the operation is not guaranteed.

Caution It is impossible to set the STB bit and NMIM or INTM bit at the same time. Be sure to set the STB bit after setting the NMIM or INTM bit.

	7	6	<5>	<4>	3	2	<1>	0	Address	After reset
PSC	0	0	NMIM	INTM	0	0	STB	0	FFFFF1FEH	00H

Bit position	Bit name	Function
5	NMIM	This is the enable/disable setting bit for standby mode release using valid edge input of NMI[Note]. 0: Enables NMI cancellation 1: Disables NMI cancellation
4	INTM	This is the enable/disable setting for standby mode release using an unmasked maskable interrupt (INTPn) (n = 0 to 4, 20 to 25, 30, 31, 100, 101)[Note]. 0: Enables maskable interrupt cancellation 1: Disables maskable interrupt cancellation
1	STB	Indicates the standby mode status. If 1 is written to this bit, the system enters standby mode (when it is in IDLE or software STOP mode). When standby mode is released, this bit is automatically reset to 0. 0: Standby mode is released 1: Standby mode is in effect

★ **Note** Setting these bits is valid only in the IDLE/software STOP mode.

Data is set in the power save control register (PSC) according to the following sequence.

<1> Set the power save mode register (PSMR) (with the following instructions).
- Store instruction (ST/SST instruction)
- Bit manipulation instruction (SET1/CLR1/NOT1 instruction)

<2> Prepare data in any one of the general-purpose registers to set to the specific register.

<3> Write arbitrary data to the command register (PRCMD).

<4> Set the power save control register (PSC) (with the following instructions).
- Store instruction (ST/SST instruction)
- Bit manipulation instruction (SET1/CLR1/NOT1 instruction)

<5> Assert the NOP instructions (5 instructions (<5> to <9>).

[Sample coding]

```
<1> ST.B   r11, PSMR [r0]       ; Set PSMR register
<2> MOV    0×04, r10            ; Prepare data for setting
                                  specific register in
                                  general-purpose register
<3> ST.B   r10, PRCMD [r0]      ; Write PRCMD register
<4> ST.B   r10, PSC [r0]        ; Set PSC register
<5> NOP                         ; Dummy instruction
<6> NOP                         ; Dummy instruction
<7> NOP                         ; Dummy instruction
<8> NOP                         ; Dummy instruction
<9> NOP                         ; Dummy instruction
(next instruction)              ; Execution routine after software
                                  STOP mode and IDLE mode release
```

No special sequence is required to read the specific register.

Cautions 1. Interrupts are not acknowledged in store instructions for the command register. This coding is made on assumption that <3> and <4> above are executed by the program with consecutive store instructions. If another instruction is set between <3> and <4>, the above sequence may become ineffective when the interrupt is acknowledged by that instruction, and a malfunction of the program may result.

2. Although the data written to the PRCMD register is dummy data, use the same register as the general-purpose register used in specific register setting <4> for writing to the PRCMD register (<3>). The same method should be applied when using a general-purpose register for addressing.

3. At least 5 NOP instructions must be inserted after executing a store instruction to the PSC register to set software STOP or IDLE mode.

4. Before executing this processing, complete all DMA transfer operations.

8.5.3 HALT mode

(1) Setting and operation status

In the HALT mode, the clock generator (oscillator and PLL synthesizer) continues to operate, but the operation clock of the CPU is stopped. Since the supply of clocks to on-chip peripheral I/O units other than the CPU continues, operation continues. The power consumption of the overall system can be reduced by setting the system to HALT mode while the CPU is idle.

The system is switched to HALT mode by the HALT instruction.

Although program execution stops in the HALT mode, the contents of all registers, internal RAM, and ports are maintained in the state they were in immediately before HALT mode began. Also, operation continues for all on-chip peripheral I/O units (other than ports) that do not depend on CPU instruction processing. Table 8-2 shows the status of each hardware unit in the HALT mode.

Table 8-2. Operation Status in HALT Mode

Function	Operation Status
Clock generator	Operating
Internal system clock	Operating
CPU	Stopped
Ports	Maintained
On-chip peripheral I/O (excluding ports)	Operating
Internal data	All internal data such as CPU registers, statuses, data, and the contents of internal RAM are maintained in the state they were in immediately before HALT mode began.
AD0 to AD15	Operating
A16 to A21	
$\overline{RD}$, ASTB	
$\overline{UWR}$, $\overline{LWR}$	
$\overline{WAIT}$	
CLKOUT	Clock output

(2) Release of HALT mode

HALT mode is released by a non-maskable interrupt request, an unmasked maskable interrupt request, or $\overline{\text{RESET}}$ pin input.

(a) Release by a non-maskable interrupt request or an unmasked maskable interrupt request

HALT mode is released by a non-maskable interrupt request or by an unmasked maskable interrupt request regardless of the priority. However, if the system is set to HALT mode during an interrupt servicing routine, operation will differ as follows.

(i) If an interrupt request is generated with a lower priority than that of the interrupt request that is currently being serviced, HALT mode is released, but the newly generated interrupt request is not acknowledged. The new interrupt request is held pending.

(ii) If an interrupt request (including non-maskable interrupt requests) is generated with a higher priority than that of the interrupt request that is currently being serviced, HALT mode is released and the newly generated interrupt request is acknowledged.

Table 8-3. Operation After HALT Mode Is Released by Interrupt Request

Release Source	Enable Interrupt (EI) Status	Disable Interrupt (DI) Status
Non-maskable interrupt request	Branch to handler address	
Maskable interrupt request	Branch to handler address or execute next instruction	Execute next instruction

(b) Release by $\overline{\text{RESET}}$ pin input

This is the same as a normal reset operation.

8.5.4 IDLE mode

(1) Setting and operation status

In the IDLE mode, the clock generator (oscillator and PLL synthesizer) continues to operate, but the supply of internal system clocks is stopped which causes the overall system to stop.

When IDLE mode is released, the system can be switched to normal operation mode quickly because the oscillator's oscillation stabilization time or the PLL lockup time do not need to be secured.

The system is switched to IDLE mode by setting the PSC or PSMR register using a store instruction (ST or SST instruction) or a bit manipulation instruction (SET1, CLR1, or NOT1 instruction) (see **8.5.2 Control registers**).

In the IDLE mode, program execution is stopped, and the contents of all registers, internal RAM, and ports are maintained in the state they were in immediately before execution stopped. The operation of on-chip peripheral I/O units (excluding ports) also is stopped.

Table 8-4 shows the status of each hardware unit in the IDLE mode.

Table 8-4. Operation Status in IDLE Mode

Function	Operation Status
Clock generator	Operating
Internal system clock	Stopped
CPU	Stopped
Ports	Maintained
★ On-chip peripheral I/O (excluding ports)	Stopped (CSI0 and CSI1 are operable in slave mode)
Internal data	All internal data such as CPU registers, statuses, data, and the contents of internal RAM are maintained in the state they were in immediately before IDLE mode began.
AD0 to AD15	High impedance
A16 to A21	
$\overline{RD}$	High level output
$\overline{UWR}$, $\overline{LWR}$	
$\overline{WAIT}$	Input (no sampling)
ASTB	High-level output
CLKOUT	Low-level output

(2) Release of IDLE mode

IDLE mode is released by a non-maskable interrupt request, an unmasked maskable interrupt request (INTPn)[Note], or $\overline{\text{RESET}}$ pin input (n = 0 to 4, 20 to 25).

Note When a digital filter using clock sampling is selected as the noise eliminator for INTP20 to INTP25, IDLE mode cannot be released.

(a) Release by a non-maskable interrupt request or an unmasked maskable interrupt request

IDLE mode is released by an interrupt request only when transition to IDLE mode is performed with the INTM and NMIM bits of the PSC register set to 0.

IDLE mode is released by a non-maskable interrupt request or by an unmasked maskable interrupt request (INTPn) regardless of the priority (n = 0 to 4, 20 to 25). The operation after release is as follows.

★ **Caution When the NMIM and INTM bits of the PSC register = 1, the IDLE mode cannot be released by the non-maskable interrupt request signal and unmasked maskable interrupt request signal.**

Table 8-5. Operation After IDLE Mode Is Released by Interrupt Request

Release Source	Enable Interrupt (EI) Status	Disable Interrupt (DI) Status
Non-maskable interrupt request	Branch to handler address	
Maskable interrupt request	Branch to handler address or execute next instruction	Execute next instruction

If the system is set to IDLE mode during a maskable interrupt servicing routine, operation will differ as follows.

(i) If an interrupt request is generated with a lower priority than that of the interrupt request that is currently being serviced, IDLE mode is released, but the newly generated interrupt request is not acknowledged. The new interrupt request is held pending.

(ii) If an interrupt request (including non-maskable interrupt requests) is generated with a higher priority than that of the interrupt request that is currently being serviced, IDLE mode is released and the newly generated interrupt request is acknowledged.

If the system is set to IDLE mode during an NMI servicing routine, IDLE mode is released, but the interrupt is not acknowledged (interrupt is held pending).

Interrupt servicing that is started when IDLE mode is released by NMI pin input is handled in the same way as normal NMI interrupt servicing that occurs during an emergency (because the NMI interrupt handler address is unique). Therefore, when a program must be able to distinguish between these two situations, a software status must be prepared in advance and that status must be set before setting the PSMR register using a store instruction or a bit manipulation instruction. By checking for this status during NMI interrupt servicing, an ordinary NMI can be distinguished from the processing that is started when IDLE mode is released by NMI pin input.

(b) Release by $\overline{\text{RESET}}$ pin input

This is the same as a normal reset operation.

8.5.5 Software STOP mode

(1) Setting and operation status

In the software STOP mode, the clock generator (oscillator and PLL synthesizer) is stopped. The overall system is stopped, and ultra-low power consumption is achieved in which only leak current is lost.

The system is switched to software STOP mode by using a store instruction (ST or SST instruction) or bit manipulation instruction (SET1, CLR1, or NOT1 instruction) to set the PSC and PSMR registers (see **8.5.2 Control registers**).

When PLL mode and resonator connection mode (CESEL bit of CKC register = 1) are used, the oscillator's oscillation stabilization time must be secured after software STOP mode is released.

In both PLL and direct mode, following the release of software STOP mode, execution of the program is started after the count time of the time base counter has elapsed.

Although program execution stops in software STOP mode, the contents of all registers, internal RAM, and ports are maintained in the state they were in immediately before software STOP mode began. The operation of all on-chip peripheral I/O units (excluding ports) is also stopped.

Table 8-6 shows the status of each hardware unit in the software STOP mode.

Table 8-6. Operation Status in Software STOP Mode

Function	Operation Status
Clock generator	Stopped
Internal system clock	Stopped
CPU	Stopped
Ports	Maintained[Note]
★ On-chip peripheral I/O (excluding ports)	Stopped (CSI0 and CSI1 are operable in slave mode)
Internal data	All internal data such as CPU registers, statuses, data, and the contents of internal RAM are retained in the state before software STOP mode has been set[Note].
AD0 to AD15	High impedance
A16 to A21	
$\overline{RD}$	High-level output
$\overline{UWR}$, $\overline{LWR}$	
$\overline{WAIT}$	Input (no sampling)
ASTB	High-level output
CLKOUT	Low-level output

Note When the V_{DD} value is within the operable range. However, even if it drops below the minimum operable voltage, as long as the data retention voltage V_{DDDR} is maintained, the contents of only the internal RAM will be maintained.

(2) Release of software STOP mode

Software STOP mode is released by a non-maskable interrupt request, an unmasked maskable interrupt request (INTPn)[Note], or $\overline{\text{RESET}}$ pin input. Also, to release software STOP mode when PLL mode (CKSEL pin = low level) and resonator connection mode (CESEL bit of CKC register = 0) are used, the oscillator's oscillation stabilization time must be secured (n = 0 to 4, 20 to 25)

Moreover, the oscillation stabilization time must be secured even when an external clock is connected (CESEL bit = 1). See **8.4 PLL Lockup** for details.

Note When a digital filter using clock sampling is selected as the noise eliminator for INTP20 to INTP25, software STOP mode cannot be released.

(a) Release by a non-maskable interrupt request or an unmasked maskable interrupt request

Software STOP mode is released by an interrupt request only when transition to software STOP mode is performed with the INTM and NMIM bits of the PSC register set to 0.

Software STOP mode is released by a non-maskable interrupt request or by an unmasked maskable interrupt request (INTPn) regardless of the priority (n = 0 to 4, 20 to 25). The operation after release is as follows.

★ **Caution When the NMIM and INTM bits of the PSC register = 1, the software STOP mode cannot be released by the non-maskable interrupt request signal and unmasked maskable interrupt request signal.**

Table 8-7. Operation After Software STOP Mode Is Released by Interrupt Request

Cancellation Source	Enable Interrupt (EI) Status	Disable Interrupt (DI) Status
Non-maskable interrupt request	Branch to handler address	
Maskable interrupt request	Branch to handler address or execute next instruction	Execute next instruction

If the system is set to software STOP mode during an interrupt servicing routine, operation will differ as follows.

(i) If an interrupt request is generated with a lower priority than that of the interrupt request that is currently being servicing, software STOP mode is released, but the newly generated interrupt request is not acknowledged. The new interrupt request is held pending.

(ii) If an interrupt request (including non-maskable interrupt requests) is generated with a higher priority than that of the interrupt request that is currently being serviced, software STOP mode is released and the newly generated interrupt request is acknowledged.

If the system is set to software STOP mode during an NMI servicing routine, software STOP mode is released, but the interrupt is not acknowledged (interrupt is held pending).

Interrupt servicing that is started when software STOP mode is released by NMI pin input is handled in the same way as normal NMI interrupt servicing that occurs during an emergency (because the NMI interrupt handler address is unique). Therefore, when a program must be able to distinguish between these two situations, a software status must be prepared in advance and that status must be set before setting the PSMR register using a store instruction or a bit manipulation instruction.

By checking for this status during NMI interrupt servicing, an ordinary NMI can be distinguished from the servicing that is started when software STOP mode is released by NMI pin input.

(b) Release by $\overline{\text{RESET}}$ pin input

This is the same as a normal reset operation.

8.6 Securing Oscillation Stabilization Time

8.6.1 Oscillation stabilization time security specification

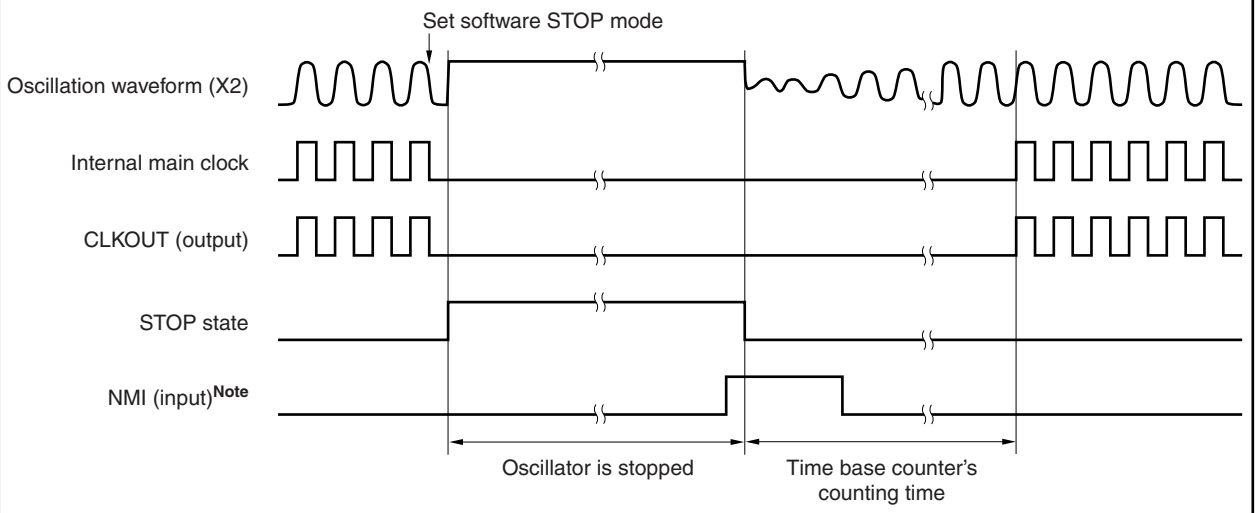
Two specification methods can be used to secure the time from when software STOP mode is released until the stopped oscillator stabilizes.

(1) Securing the time using an on-chip time base counter

Software STOP mode is released when a valid edge is input to the NMI pin or a maskable interrupt request is input (INTPn). When a valid edge is input to the pin causing the start of oscillation, the time base counter (TBC) starts counting, and the time until the clock output from the oscillator stabilizes is secured during that counting time (n = 0 to 4, 20 to 25).

Oscillation stabilization time = TBC counting time

After a fixed time, internal system clock output begins, and processing branches to the NMI interrupt or maskable interrupt (INTPn) handler address.



Note Valid edge: When specified as the rising edge.

The NMI pin should usually be set to an inactive level (for example, high level when the valid edge is specified as the falling edge) in advance.

Software STOP mode is immediately released if an operation that sets software STOP mode before the CPU can acknowledge interrupts is performed due to NMI valid edge input or maskable interrupt request input (INTPn).

If the direct mode or external clock connection mode (CESEL bit of CKC register = 1) is used, program execution begins after the count time of the time base counter has elapsed.

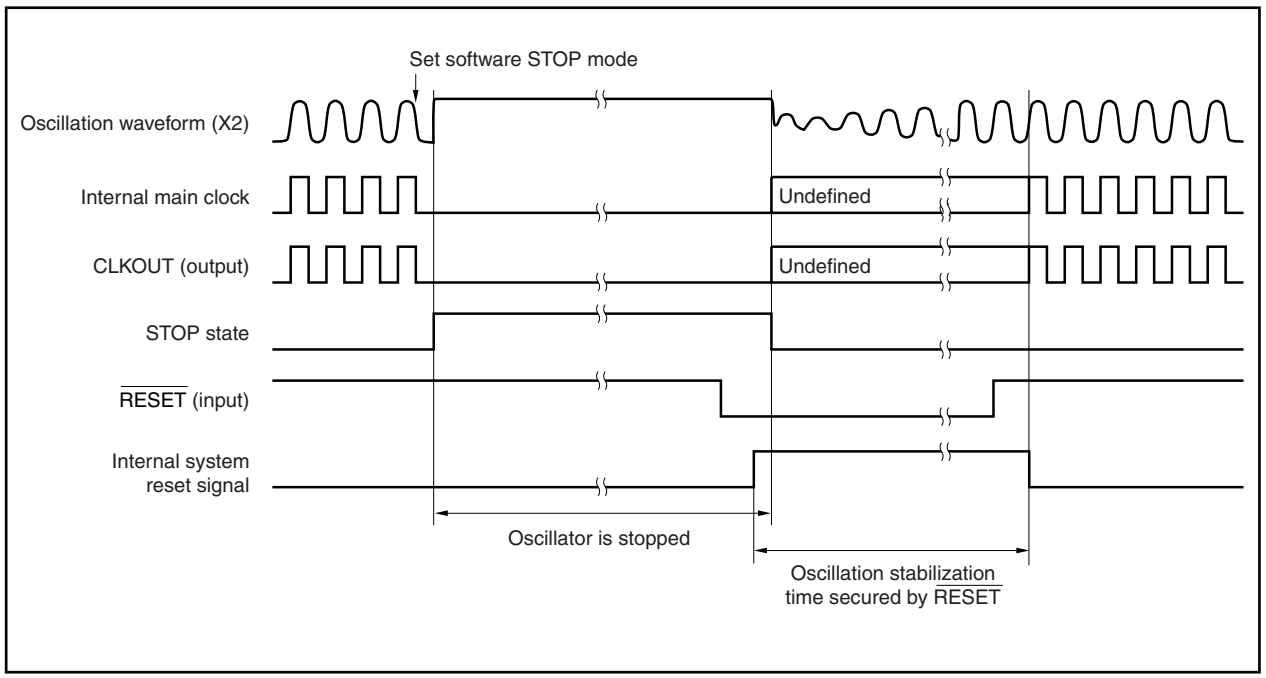
Also, even if the PLL mode and resonator connection mode (CESEL bit of CKC register = 0) are used, program execution begins after the oscillation stabilization time is secured by the time base counter.

(2) Securing the time according to the signal level width ($\overline{\text{RESET}}$ pin input)

Software STOP mode is released by falling edge input to the $\overline{\text{RESET}}$ pin.

The time until the clock output from the oscillator stabilizes is secured based on the low-level width of the signal that is input to the pin.

The supply of internal system clocks begins after a rising edge is input to the $\overline{\text{RESET}}$ pin, and processing branches to the handler address used for a system reset.



8.6.2 Time base counter (TBC)

The time base counter (TBC) is used to secure the oscillator's oscillation stabilization time when software STOP mode is released.

When an external clock is connected (CESEL bit of CKC register = 1) or a resonator is connected (PLL mode and CESEL bit of CKC register = 0), the TBC counts the oscillation stabilization time after software STOP mode is released, and program execution begins after the count is completed.

The TBC count clock is selected by the TBCS bit of the CKC register, and the next counting time can be set (reference).

Table 8-8. Counting Time Examples ($f_{XX} = 10 \times f_X$)

TBCS Bit	Count Clock	Counting Time
		f_X = 4.0000 MHz
0	$f_X/2^8$	16.4 ms
1	$f_X/2^9$	32.8 ms

f_{XX}: Internal system clock

f_X: External oscillation frequency

CHAPTER 9 TIMER/COUNTER FUNCTION

9.1 Timer 0

9.1.1 Features (timer 0)

Timers 00 and 01 (TM00, TM01) are 16-bit timer/counters ideal for controlling high-speed inverters such as motors.

- 3-phase PWM output function
  PWM mode 0 (symmetric triangular wave)
  PWM mode 1 (asymmetric triangular wave)
  PWM mode 2 (sawtooth wave)
- Interrupt culling function
  Culling ratios: 1/1, 1/2, 1/4, 1/8, 1/16
- Forcible 3-phase PWM output stop function
  3-phase PWM output can be forcibly stopped by inputting a signal to the external signal input pin ESOn when an anomaly occurs.
  This function can also be used when the clock is stopped.
- Real-time output function
  3-phase PWM output or rectangular wave output can be selected at the desired timing.
- Output of positive phase and negative phase or positive phase and in-phase of 3-phase PWM output

9.1.2 Function overview (timer 0)

- 16-bit timer (TM0n) for 3-phase PWM inverter control: 2 channels
- Compare registers: 6 registers × 2 channels
- 12-bit dead-time timers (DTMn0 to DTMn2): 3 timers × 2 channels
- Count clock division selectable by prescaler (set the frequency of the count clock to 40 MHz or less)
- Base clock (f_{CLK}): 2 types (set f_{CLK} to 40 MHz or less)
  f_{XX} and $f_{XX}/2$ can be selected
- Prescaler division ratio
  The following division ratios can be selected according to the base clock (f_{CLK}).

Division Ratio	Base Clock (f_{CLK})	
	f_{XX} Selected	$f_{XX}/2$ Selected
1/1	f_{XX}	$f_{XX}/2$
1/2	$f_{XX}/2$	$f_{XX}/4$
1/4	$f_{XX}/4$	$f_{XX}/8$
1/8	$f_{XX}/8$	$f_{XX}/16$
1/16	$f_{XX}/16$	$f_{XX}/32$
1/32	$f_{XX}/32$	$f_{XX}/64$

- Interrupt request sources
  (a) Compare-match interrupt request: 9 types
    - Interrupt request signal INTCM0n3 generated by match of TM0n register count value and compare register CM0n3
    - Interrupt request signals INTCM010 to INTCM012, INTCM0n4, and INTCM0n5 generated by match of TM0n register count value and compare registers CM010 to CM012, CM0n4, and CM0n5

Setting Condition	INTCM010 to INTCM012, INTCM0n4, INTCM0n5 Signal Occurrence Status
CM010 to CM012, CM0n4, CM0n5 ≤ CM0n3	Occurs
CM010 to CM012, CM0n4, CM0n5 = 0000H	Occurs
CM010 to CM012, CM0n4, CM0n5 > CM0n3	Does not occur

  (b) Underflow interrupt request: 2 types
    - Interrupt request signal INTTM0n generated by underflow of the TM0n register
- External pulse output (TO0n0 to TO0n5): 6 × 2 channels

Remark f_{XX}: Internal system clock
n = 0, 1

9.1.3 Functions added to V850E/IA2

(1) Addition of BFCMn4 and CM0n4 registers, and BFCMn5 and CM0n5 registers

When the TM0CEn bit of the TMC0n register is 1 (counting enabled), transferring data from the BFCMn4 or BFCMn5 register to the CM0n4 or CM0n5 register is enabled or disabled by the BFTEN bit of the TMC0n register (n = 0, 1).

(2) Compare-match interrupt output function of CM010 to CM012, CM0n4, and CM0n5 registers (INTCM010 to INTCM012, INTCM0n4, INTCM0n5)

The features of the compare-match interrupt output function (INTCM010 to INTCM012, INTCM0n4, INTCM0n5) of the CM010 to CM012, CM0n4, and CM0n5 registers are as follows (n = 0, 1):

(a) This interrupt signal is not affected by the STINTn bit of the TMC0n register that specifies occurrence of an interrupt when timer TM0n is started.

(b) The compare-match interrupt output function of the CM010 to CM012, CM0n4, and CM0n5 registers does not have an interrupt culling function. Therefore, it is not affected by the CUL02 to CUL00 bits of the TMC0n register.

The sources of this interrupt signal are shown below.

Table 9-1. Sources of INTCM010 to INTCM012, INTCM0n4, and INTCM0n5

Unit	Interrupt Name	A/D Trigger Function	Interrupt Function	DMA Trigger Source
TM00	INTCM000 to INTCM002[Note]	×	×	×
	INTCM004, INTCM005	○	○	×
TM01	INTCM010 to INTCM012	×	○	○
	INTCM014, INTCM015	○	○	○

Note The V850E/IA2 does not include INTCM000 to INTCM002.

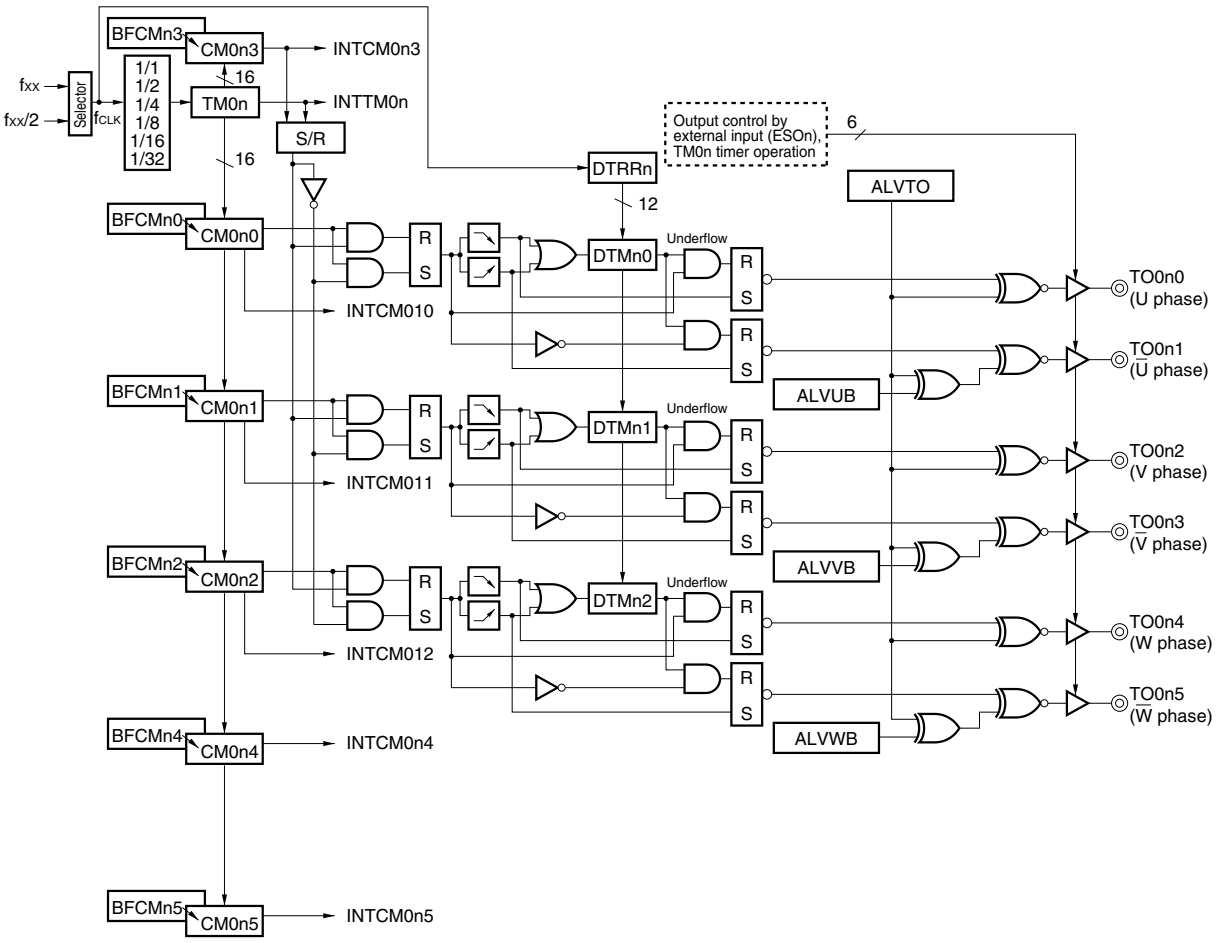
Remarks 1. ○: Function provided
×: Function not provided

2. n = 0, 1

9.1.4 Basic configuration

The basic configuration is shown below.

Figure 9-1. Block Diagram of Timer 0 (Mode 0: Symmetric Triangular Wave, Mode 1: Asymmetric Triangular Wave)



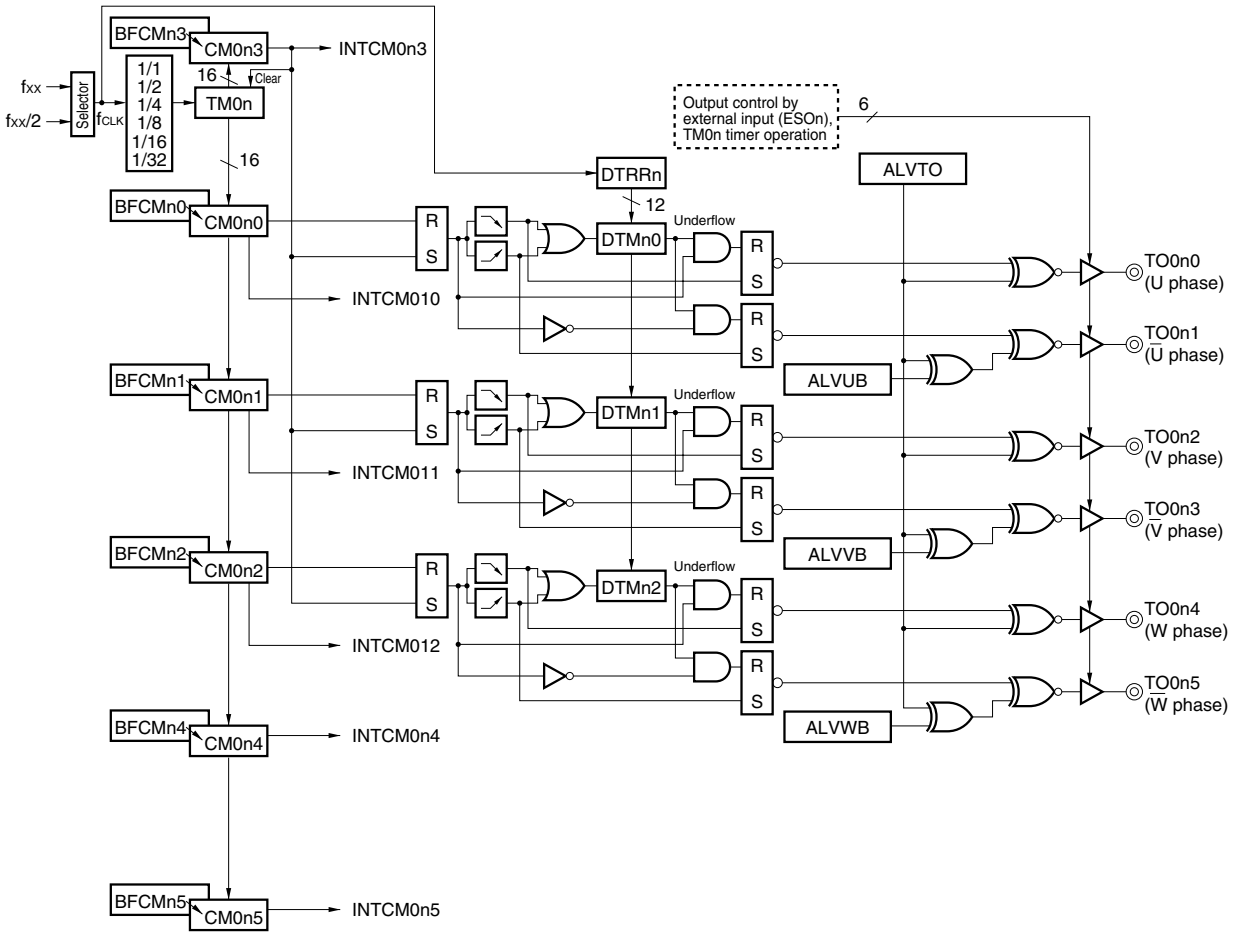
Remarks 1. TM0n: Timer register
CM0n0 to CM0n5: Compare registers
BFCMn0 to BFCMn5: Buffer registers
DTRRn: Dead-time timer reload register
DTMn0 to DTMn2: Dead-time timers
ALVTO: Bit 7 of TOMRn register
ALVUB: Bit 6 of TOMRn register
ALVVB: Bit 5 of TOMRn register
ALVWB: Bit 4 of TOMRn register
S/R: Set/Reset

2. n = 0, 1

3. f_{XX}: Internal system clock

4. f_{CLK}: Base clock (40 MHz (MAX.))

Figure 9-2. Block Diagram of Timer 0 (Mode 2: Sawtooth Wave)



Remarks 1. TM0n: Timer register
CM0n0 to CM0n5: Compare registers
BFCMn0 to BFCMn5: Buffer registers
DTRRn: Dead-time timer reload register
DTMn0 to DTMn2: Dead-time timers
ALVTO: Bit 7 of TOMRn register
ALVUB: Bit 6 of TOMRn register
ALVVB: Bit 5 of TOMRn register
ALVWB: Bit 4 of TOMRn register

2. n = 0, 1

3. f_{XX}: Internal system clock

4. f_{CLK}: Base clock (40 MHz (MAX.))

(1) Timers 00, 01 (TM00, TM01)

TM0n operates as a 16-bit up/down timer or up timer. The cycle is controlled by compare register 0n3 (CM0n3) (n = 0, 1).

TM0n start/stop is controlled by the TM0CEn bit of timer control register 0n (TMC0n).

Division by the prescaler can be selected for the count clock from among f_{CLK}, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/16$, $f_{CLK}/32$ using the PRM02 to PRM00 bits of the TMC0n registers (f_{CLK}: base clock, see **9.1.5 (1) Timer 0 clock selection register (PRM01)**).

The conditions when TM0n becomes 0000H are as follows.

- Reset input
- TM0CEn bit = 0
- TM0n register and compare register 0n3 (CM0n3) match (PWM mode 2 (sawtooth wave) only)
- Immediately after overflow or underflow

The TM0n timer has 3 operation modes, shown in Table 9-2. The operation mode is selected using timer control register 0n (TMC0n).

Table 9-2. Operation Modes of Timer 0

Operation Mode	Count Operation	Timer Clear Source	Interrupt Source	BFCMn3 → CM0n3 Transfer Timing	BFCMn0 to BFCMn2, BFCMn4, BFCMn5 → CM0n0 to CM0n2, CM0n4, CM0n5 Transfer Timing
PWM mode 0 (symmetric triangular wave)	Up/down	–	INTTM0n, INTCM010 to INTCM012, INTCM0n3 to INTCM0n5	INTTM0n	INTTM0n
PWM mode 1 (asymmetric triangular wave)	Up/down	–	INTTM0n, INTCM010 to INTCM012, INTCM0n3 to INTCM0n5	INTTM0n	INTTM0n, INTCM0n3
PWM mode 2 (sawtooth wave)	Up	INTCM0n3	INTCM010 to INTCM012, INTCM0n3 to INTCM0n5	INTCM0n3	INTCM0n3

Caution Even if TM0ICn, CM03ICn, or an interrupt mask flag of the IMR0 register (TM0MKn or CM03MKn) is set (interrupt disabled) as the interrupt sources INTTM0n and INTCM0n3, it simply results in no interrupt occurrence and does not affect the operation of timer 0.
The interrupt sources INTCM010 to INTCM012, INTCM0n4, and INTCM0n5 do not affect the operation of timer 0 regardless of whether the interrupt is masked or not.

Remark n = 0, 1

(2) Dead-time timers 00 to 02, 10 to 12 (DTM00 to DTM02, DTM10 to DTM12)

DTMn0 to DTMn2 are dedicated 12-bit down timers that generate dead time, which is effective for inverter control applications. DTMn0 to DTMn2 operate as one-shot timers.

Counting by a dead-time timer is enabled or disabled by the TM0CEDn bit of timer control register 0n (TMC0n) and cannot be controlled by software. Dead-time timer count start and stop is controlled by hardware.

A dead-time timer starts counting down when the value of dead-time timer reload register n (DTRRn) is transferred in synchronization with the compare match timing of CM0n0 to CM0n2.

When the value of a dead-time timer changes from 000H to FFFH, the dead-time timer generates an underflow signal, and the timer stops at the value FFFH.

If the value of a dead-time timer matches the value of the corresponding compare register before underflow of the dead-time timer takes place, the value of DTRRn is transferred to the dead-time timer again, and the timer starts counting down.

The count clock of the dead-time timer is fixed to the base clock (f_{CLK}), and the dead-time width is (set value of DTRRn + 1)/base clock (f_{CLK}).

If TM0n operates in PWM mode 0 or PWM mode 1 with the dead-time timer count operation disabled, an inverted signal without dead time is output to TO0n0 and TO0n1, TO0n2 and TO0n3, and TO0n4 and TO0n5.

(3) Dead-time timer reload registers 0, 1 (DTRR0, DTRR1)

The DTRRn register is a 12-bit register used to set the values of the three dead-time timers (DTMn0 to DTMn2 registers) (n = 0, 1). However, a value is transferred from the DTRRn register to each dead-time register independently.

DTRRn can be read/written in 16-bit units. All 0s are read for the higher 4 bits when the DTRRn register is read accessed in 16 bits.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DTRR0	0	0	0	0													FFFFF570H	0FFFH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DTRR1	0	0	0	0													FFFFF5B0H	0FFFH

Cautions 1. Changing the value of the DTRRn register during TM0n operation (TM0CEn bit of TMC0n register = 1) is prohibited.

2. Be sure to write 0 to the higher 4 bits.

(4) Compare registers 000 to 002, 010 to 012 (CM000 to CM002, CM010 to CM012)

CM0n0 to CM0n2 are 16-bit registers that always compare their own values with the value of TM0n. If the value of a compare register matches the value of TM0n, the compare register outputs a trigger signal, and changes the contents of the flip-flop (F/F) connected to the compare register. Each of CM0n0 to CM0n2 is provided with a buffer register (BFCMn0 to BFCMn2), so that the contents of the buffer are transferred to CM0n0 to CM0n2 at the next transfer timing. Transfer is enabled or disabled by the BFTEN bit of the TMC0n register.

If CM010 to CM012 of timer 01 match TM01, the INTCM010 to INTCM012 interrupts occur.

(5) Compare registers 004, 005, 014, 015 (CM004, CM005, CM014, CM015)

CM0n4 and CM0n5 are 16-bit registers that always compare their value with TM0n. If the value of these registers matches the value of TM0n, the registers generate an interrupt signal (INTCM0n4 or INTCM0n5). CM0n4 and CM0n5 are also provided with a buffer register (BFCMn4 or BFCMn5), the contents of which are transferred to CM0n4 or CM0n5 at the next transfer timing. Transfer is enabled or disabled by the BFTEN bit of the TMC0n register.

(6) Compare registers 003, 013 (CM003, CM013)

CM0n3 is a 16-bit register that always compare its value with the value of TM0n. If the values match, CM0n3 outputs an interrupt signal (INTCM0n3). CM0n3 controls the maximum count value of TM0n, and if the values match, it performs the following operations at the next timer count clock.

- In triangular wave setting mode (PWM modes 0, 1): Switches TM0n operation from count up to count down
- Sawtooth wave setting mode (PWM mode 2): Clears the count value of TM0n

CM0n3 also has a buffer register (BFCMn3) and transfers the buffer contents to CM0n3 at the next transfer timing. Transfer enable or disable is controlled by the BFTE3 bit of the TMC0n register.

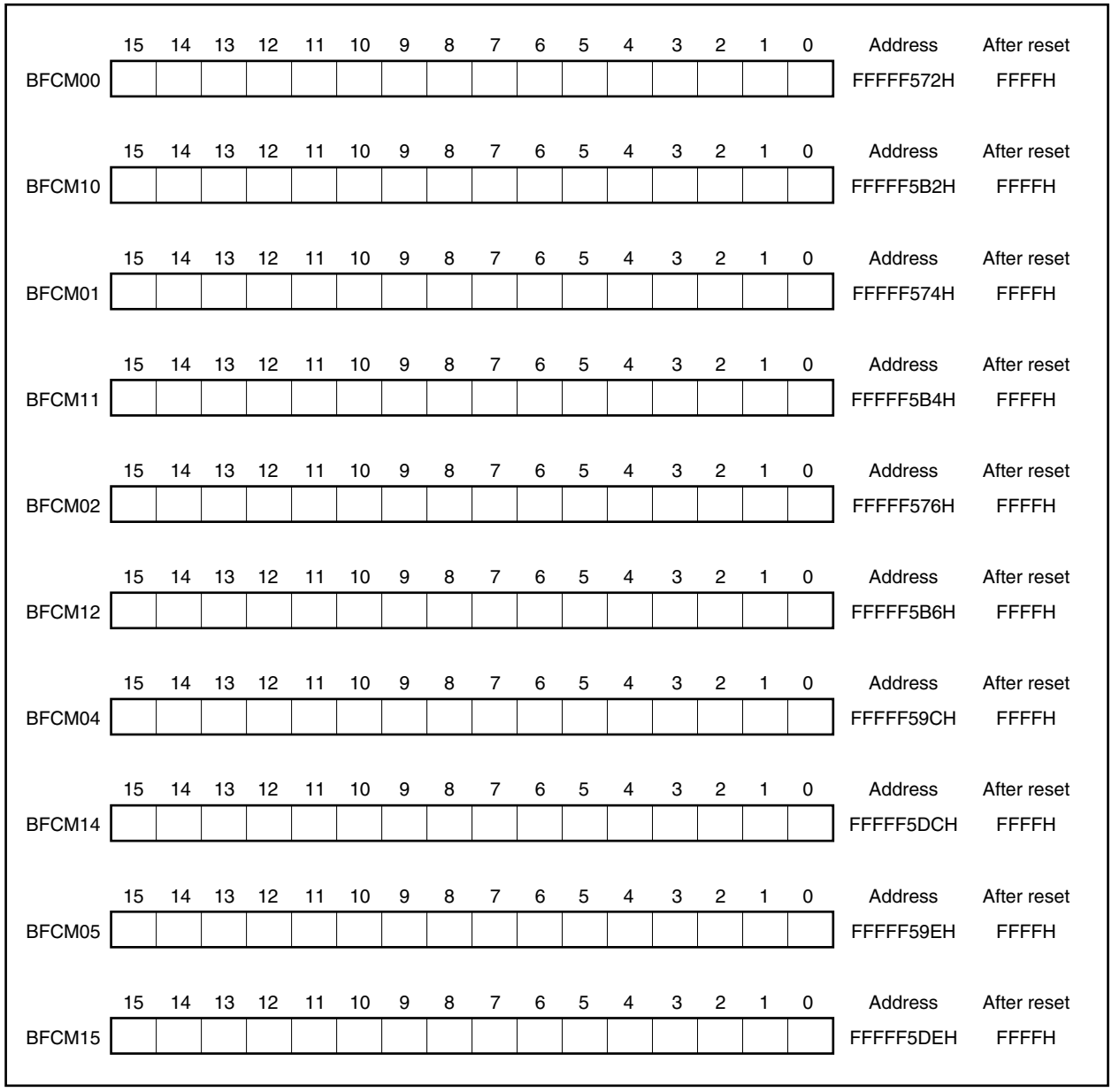
(7) Buffer registers CM00 to CM02, CM04, CM05, CM10 to CM12, CM14, CM15 (BFCM00 to BFCM02, BFCM04, BFCM05, BFCM10 to BFCM12, BFCM14, BFCM15)

BFCMn0 to BFCMn2, BFCMn4, and BFCMn5 are 16-bit registers that transfer data to the compare register (CM0n0 to CM0n2, CM0n4, CM0n5) corresponding to each buffer register when an interrupt signal (INTCM0n3/INTTM0n) is generated.

These registers can be read/written in 16-bit units.

Caution The set values of the BFCMn0 to BFCMn2, BFCMn4, and BFCMn5 registers are transferred to the CM0n0 to CM0n2, CM0n4, and CM0n5 registers at the following timing (n = 0, 1).

- **When TM0CEn bit of TMC0n register = 0: Transfer at the next operation timing after writing to the BFCMn0 to BFCMn2, BFCMn4, and BFCMn5 registers**
- **When TM0CEn bit of TMC0n register = 1: The value of the BFCMn0 to BFCMn2, BFCMn4, and BFCMn5 registers is transferred to the CM0n0 to CM0n2, CM0n4, and CM0n5 registers upon occurrence of INTTM0n or INTCM0n3. At this time, transfer enable or disable is controlled by the BFTEN bit of the timer control register (TMC0n).**


15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Address After reset
BFCM00 FFFFF572H FFFFH
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Address After reset
BFCM10 FFFFF5B2H FFFFH
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Address After reset
BFCM01 FFFFF574H FFFFH
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Address After reset
BFCM11 FFFFF5B4H FFFFH
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Address After reset
BFCM02 FFFFF576H FFFFH
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Address After reset
BFCM12 FFFFF5B6H FFFFH
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Address After reset
BFCM04 FFFFF59CH FFFFH
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Address After reset
BFCM14 FFFFF5DCH FFFFH
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Address After reset
BFCM05 FFFFF59EH FFFFH
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Address After reset
BFCM15 FFFFF5DEH FFFFH

(8) Buffer registers CM03, CM13 (BFCM03, BFCM13)

BFCMn3 is a 16-bit register that transfers data to the compare register at any timing. Transfer enable or disable is controlled by the BFTE3 bit of the TMC0n register.

BFCMn3 can be read/written in 16-bit units.

Cautions 1. The set value of the BFCMn3 register is transferred to the CM0n3 register at the following timing (n = 0, 1).

- **When TM0CEn bit of TMC0n register = 0: Transfer at the next operation timing after writing to the BFCMn3 register**
- **When TM0CEn bit of TMC0n register = 1: The value of the BFCMn3 register is transferred to the CM0n3 register upon occurrence of INTTM0n. At this time, transfer enable or disable is controlled by the BFTE3 bit of the timer control register (TMC0n).**

2. Setting the BFCMn3 register to 0000H is prohibited.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
BFCM03																	FFFFF578H	FFFFH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
BFCM13																	FFFFF5B8H	FFFFH

9.1.5 Control registers

(1) Timer 0 clock selection register (PRM01)

The PRM01 register is used to select the base clock (f_{CLK}) of timer 0 (TM0n).

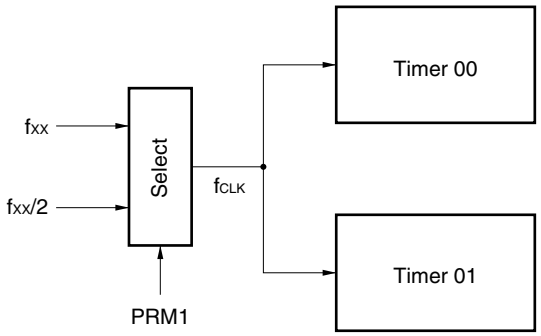
It can be read/written in 8-bit or 1-bit units.

Caution Always set this register before using the timer.

	7	6	5	4	3	2	1	0	Address	After reset
PRM01	0	0	0	0	0	0	0	PRM1	FFFFF5D0H	00H

Bit position	Bit name	Function
0	PRM1	Specifies the base clock (f_{CLK}) of timer 0 (TM0n) (See **Figure 9-3**). 0: $f_{XX}/2$ 1: f_{XX} **Caution Set f_{CLK} to 40 MHz or less.** **Remark** f_{XX}: Internal system clock

Figure 9-3. Timer 00 and Timer 01 Clock



Remarks 1. f_{XX}: Internal system clock

2. f_{CLK}: Base clock

(2) Timer control registers 00, 01 (TMC00, TMC01)

TMC0n is a 16-bit register that sets the operation of timer 0 (TM0n).

The TMC0n register can be read/written in 16-bit units.

If the higher 8 bits of the TMC0n register are used as the TMC0nH register and the lower 8 bits as the TMC0nL register, the register can be read/written in 8-bit or 1-bit units.

Caution To operate timer 0, first set TM0CEn = 0 and then set TM0CEn = 1.

(1/4)

	<15>	<14>	13	12	11	10	9	8	7	6	<5>	4	3	2	1	0	Address	After reset
TMC00	TM0CE0	STINT0	CUL02	CUL01	CUL00	PRM02	PRM01	PRM00	0	0	TM0CED0	BFTE3	BFTEN	MBFTE	MOD01	MOD00	FFFFF57AH	0508H

	<15>	<14>	13	12	11	10	9	8	7	6	<5>	4	3	2	1	0	Address	After reset
TMC01	TM0CE1	STINT1	CUL02	CUL01	CUL00	PRM02	PRM01	PRM00	0	0	TM0CED1	BFTE3	BFTEN	MBFTE	MOD01	MOD00	FFFFF5BAH	0508H

Bit position	Bit name	Function
15	TM0CEn	Specifies the operation of TM0n. 0: Count disabled (stops after all count values are cleared) 1: Count enabled **Caution When TM0CEn = 0, TO0n0 to TO0n5 output becomes high impedance.**
14	STINTn	Specifies interrupt during TM0n timer start. 0: Interrupt not generated at operation start 1: Interrupt generated at operation start When STINTn = 1, an interrupt is generated immediately after the rising edge of the TM0CEn signal. When MOD01 = 0 (triangular wave mode), the INTTM0n interrupt (see **Figure 9-4**) is generated, and when MOD01 = 1 (sawtooth wave mode), the INTCM0n3 interrupt is generated. **Cautions 1. Changing the STINTn bit during TM0n operation (TM0CEn bit = 1) is prohibited.** **2. The INTCM010 to INTCM012, INTCM0n4, and INTCM0n5 interrupts are not affected by the STINTn bit (an interrupt does not occur when the timer is started if STINTn = 1).**
13 to 11	CUL02 to CUL00	Specifies the interrupt culling ratio.

CUL02	CUL01	CUL00	Interrupt culling ratio
0	0	0	1/1
0	0	1	1/2
0	1	0	1/4
0	1	1	1/8
1	0	0	1/16
Other than above			Culling not performed

Remark n = 0, 1

(2/4)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>13 to 11</td><td>CUL02 to CUL00</td><td>Cautions 1. The INTTM0n and INTCM0n3 interrupts can be culled at the same culling ratio (1/1, 1/2, 1/4, 1/8, 1/16).
2. Even when BFTE3 = 1, BFTEN = 1 (settings to transfer data from the BFCMn0 to BFCMn3 registers to the CM0n0 to CM0n3 registers), transfer is not performed at the generation timing of the culled INTTM0n and INTCM0n3 interrupts if MBFTE = 0.
3. If the culling ratio is changed during a count operation, the new culling ratio is applied after an interrupt has occurred at the culling ratio prior to the change (see Figure 9-5).
4. The INTCM010 to INTCM012, INTCM0n4, and INTCM0n5 interrupts are not affected by the CUL02 to CUL00 bits (the interrupts occur each time at the same culling ratio as when CUL02 to CUL00 = 000 (1/1)).</td></tr>
<tr><td>10 to 8</td><td>PRM02 to PRM00</td><td>Specifies the count clock for TM0n.
<table>
<tr><th>PRM02</th><th>PRM01</th><th>PRM00</th><th>Count clock</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>f_{CLK}</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>$f_{CLK}/2$</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>$f_{CLK}/4$</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>$f_{CLK}/8$</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>$f_{CLK}/16$</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>$f_{CLK}/32$</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table>
Caution The division ratio switch timing is from when the TM0n value has become 0000H and the INTTM0n interrupt has occurred. Therefore, the division ratio is not switched at the timing that corresponds to interrupt culling.

Remark For the base clock (f_{CLK}), see 9.1.5 (1) Timer 0 clock selection register (PRM01).</td></tr>
<tr><td>5</td><td>TM0CEDn</td><td>Specifies the operation of the DTMn0 to DTMn2 timers..
0: DTMn0 to DTMn2 perform count operation
1: DTMn0 to DTMn2 stopped

Cautions1. Changing the TM0CEDn bit during TM0n operation (TM0CEn = 1) is prohibited.
2. If TM0n is operated when TM0CEDn = 1, a signal without dead time is output to the TO0n0 to TO0n5 pins.</td></tr>
</table>

Remark n = 0, 1

(3/4)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>4</td><td>BFTE3</td><td>Specifies transfer of data from the BFCMn3 register to the CM0n3 register.
0: Transfer disabled
1: Transfer enabled
The transfer timing from the BFCMn3 register to the CM0n3 register is as follows.
<table>
<tr><th>BFTE3</th><th>TM0n operation mode</th><th>BFCMn3 → CM0n3 transfer timing</th></tr>
<tr><td>0</td><td>All modes</td><td>No transfer</td></tr>
<tr><td>1</td><td>PWM mode 0 (symmetric triangular wave)</td><td>INTTM0n</td></tr>
<tr><td>1</td><td>PWM mode 1 (asymmetric triangular wave)</td><td>INTTM0n</td></tr>
<tr><td>1</td><td>PWM mode 2 (sawtooth wave)</td><td>INTCM0n3</td></tr>
</table>
When BFTE3 = 1, the value of the BFCMn3 register is transferred to the CM0n3 register upon occurrence of the INTTM0n or INTCM0n3 interrupt.</td></tr>
<tr><td>3</td><td>BFTEN</td><td>Specifies transfer of data from the BFCMn0 to BFCMn2, BFCMn4, and BFCMn5 registers to the CM0n0 to CM0n2, CM0n4, CM0n5 registers.
0: Transfer disabled
1: Transfer enabled
<table>
<tr><th>BFTEN</th><th>TM0n operation mode</th><th>BFCMn0 to BFCMn2, BFCMn4, BFCMn5 → CM0n0 to CM0n2, CM0n4, CM0n5 transfer timing</th></tr>
<tr><td>0</td><td>All modes</td><td>Don’t transfer</td></tr>
<tr><td>1</td><td>PWM mode 0 (symmetric triangular wave)</td><td>INTTM0n</td></tr>
<tr><td>1</td><td>PWM mode 1 (asymmetric triangular wave)</td><td>INTTM0n, INTCM0n3</td></tr>
<tr><td>1</td><td>PWM mode 2 (sawtooth wave)</td><td>INTCM0n3</td></tr>
</table>
When BFTEN = 1, the values of the BFCMn0 to BFCMn2, BFCMn4, and BFCMn5 registers are transferred to the CM0n0 to CM0n2, CM0n4, and CM0n5 registers upon occurrence of the INTTM0n or INTCM0n3 interrupt.</td></tr>
<tr><td>2</td><td>MBFTE</td><td>When culling of the INTTM0n and INTCM0n3 interrupts is set by the CUL02 to CUL00 bits, this bit specifies whether to enable or disable the BFTE3 and BFTEN bit settings upon occurrence of an interrupt for culling.
0: Disable the set values of the BFTE3 and BFTEN bits upon occurrence of a culling interrupt
1: Enable the set values of the BFTE3aand BFTEN bits upon occurrence of a culling interrupt
The various combinations are as follows.
<table>
<tr><th colspan="2" rowspan="2">MBFTE</th><th colspan="2">Operation upon occurrence of interrupt for culling</th></tr>
<tr><th>0</th><th>1</th></tr>
<tr><td rowspan="2">BFTEN</td><td>0</td><td>BFCMn0 to BFCMn2 → CM0n0 to CM0n2 transfer disabled</td><td>BFCMn0 to BFCMn2 → CM0n0 to CM0n2 transfer disabled</td></tr>
<tr><td>1</td><td>BFCMn0 to BFCMn2 → CM0n0 to CM0n2 transfer disabled</td><td>BFCMn0 to BFCMn2 → CM0n0 to CM0n2 transfer enabled</td></tr>
<tr><td rowspan="2">BFTE3</td><td>0</td><td>BFCMn3 → CM0n3 transfer disabled</td><td>BFCMn3 → CM0n3 transfer disabled</td></tr>
<tr><td>1</td><td>BFCMn3 → CM0n3 transfer disabled</td><td>BFCMn3 → CM0n3 transfer enabled</td></tr>
</table>
.</td></tr>
</table>

Remark n = 0, 1

(4/4)

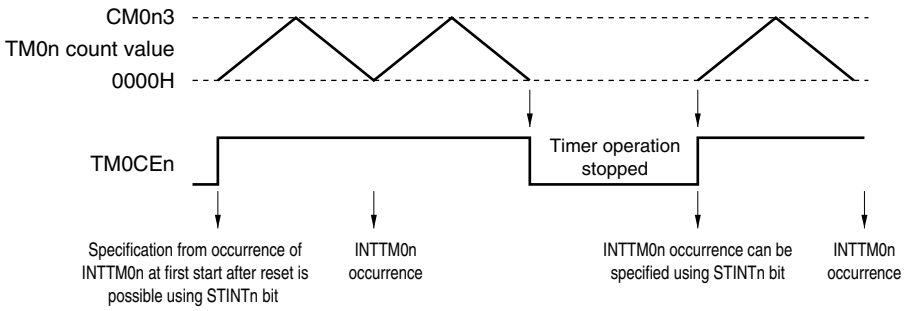
Bit position	Bit name	Function
1, 0	MOD01, MOD00	Specifies the operation mode of TM0n.

MOD 01	MOD 00	Operation mode	TM0n operation	Timer clear source	BFCMn3 → CM0n3 timing	BFCMn0 to BFCMn2, BFCMn4, BFCMn5 → CM0n0 to CM0n2, CM0n4, CM0n5 timing
0	0	PWM mode 0 (symmetric triangular wave)	Up/down	–	INTTM0n	INTTM0n
0	1	PWM mode 1 (asymmetric triangular wave)	Up/down	–	INTTM0n	INTTM0n, INTCM0n3
1	0	PWM mode 2 (sawtooth wave)	Up	INTCM0n3	INTCM0n3	INTCM0n3
1	1	Setting prohibited				

Caution Changing the value of the MOD01 and MOD00 bits during TM0n operation (TM0CEn bit = 1) is prohibited.

Remark n = 0, 1

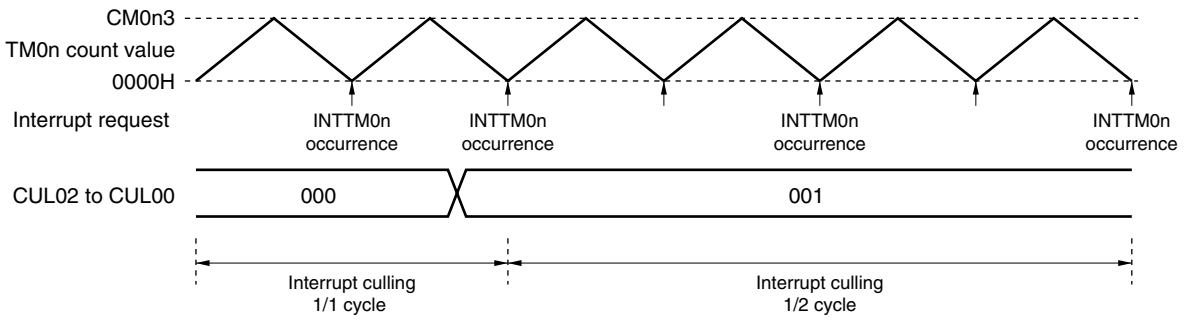
Figure 9-4. Specification of INTTM0n Interrupt in PWM Mode 0 (Symmetric Triangular Wave), PWM Mode 1 (Asymmetric Triangular Wave) (MOD01, MOD00 Bits of TMC0n Register = 0n)



Remark n = 0, 1

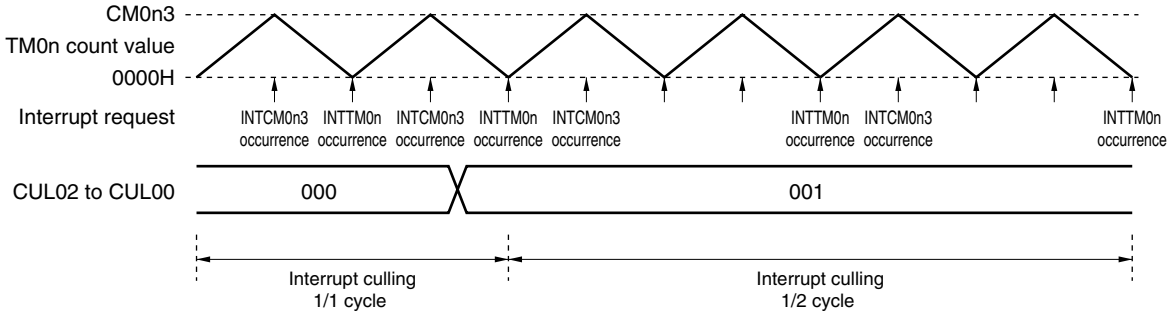
Figure 9-5. Interrupt Culling Processing

(a) PWM mode 0 (symmetric triangular wave)



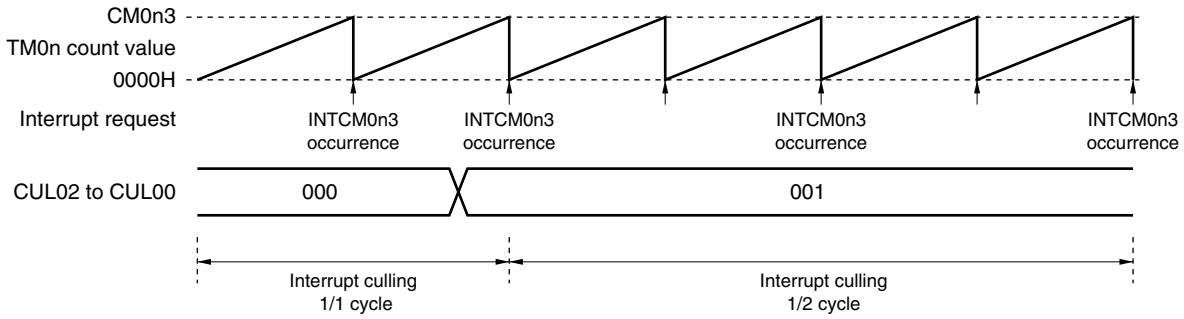
Remark n = 0, 1

(b) PWM mode 1 (asymmetric triangular wave)



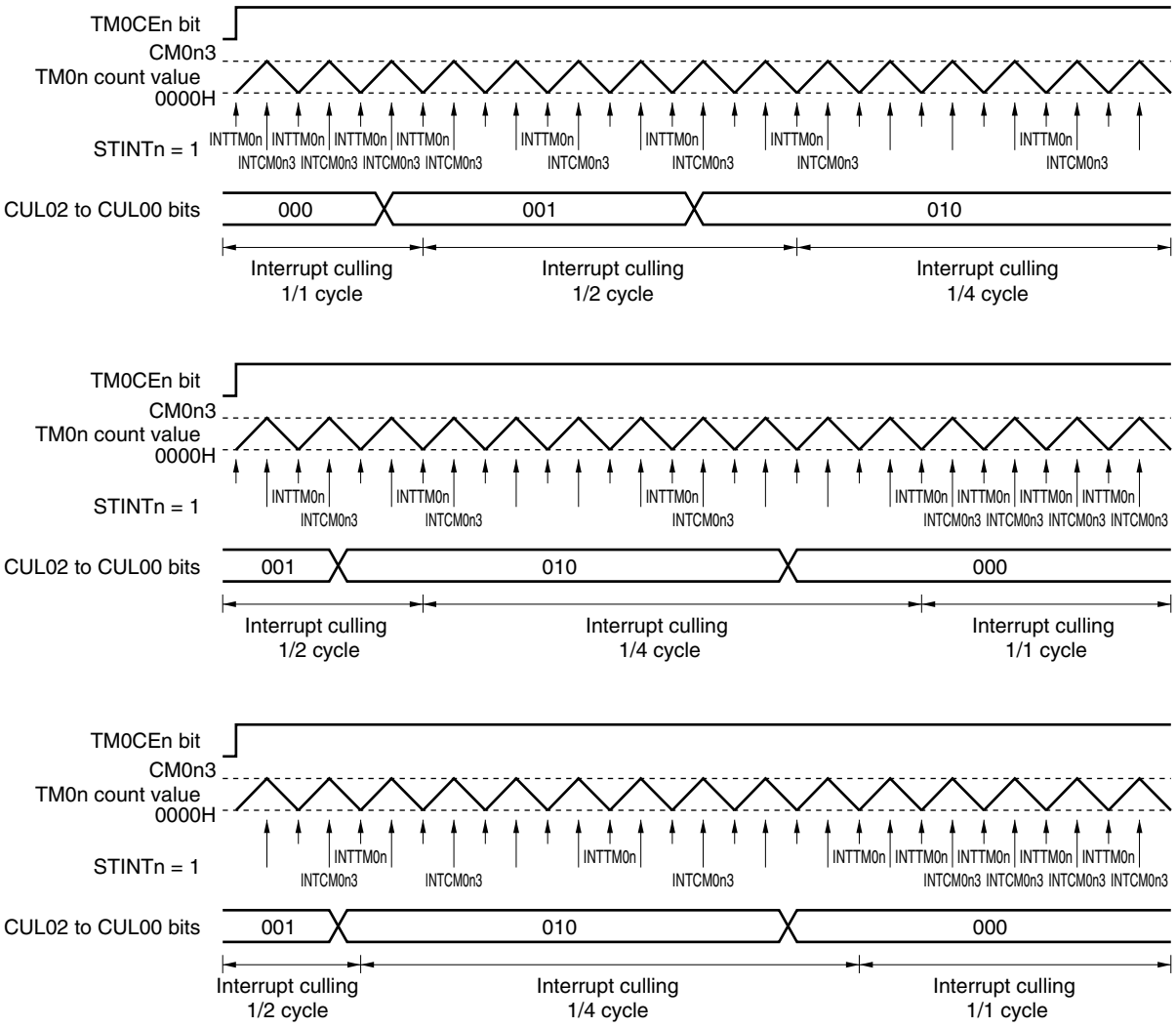
Remark n = 0, 1

(c) PWM mode 2 (sawtooth wave)



Remark n = 0, 1

Figure 9-6. Interrupt Culling Ratio Change Timing
(Relationship Between STINTn Bit Setting and CUL Bit Change): PWM Mode 1 (Asymmetric Triangular Wave)



Caution If, in TM0n, to realize the INTTM0n and INTCM0n3 culling function, the culling ratio is set to a value other than 1/1 by bits CUL02 to CUL00 and counting is started, the subsequent interrupt output sequence will differ due to the set value of the STINTn bit at count start.

Remark n = 0, 1

(3) Timer unit control registers 00, 01 (TUC00, TUC01)

TUC0n is an 8-bit register that controls the TO0n0 to TO0n5 outputs.

TUC0n can be read/written in 8-bit or 1-bit units. However, bit 0 is read-only.

	7	6	5	4	3	2	<1>	<0>	Address	After reset
TUC00	0	0	0	0	0	0	TORS0	TOSTA0	FFFFF57CH	01H

	7	6	5	4	3	2	<1>	<0>	Address	After reset
TUC01	0	0	0	0	0	0	TORS1	TOSTA1	FFFFF5BCH	01H

Bit position	Bit name	Function
1	TORSn	Flag that restarts TO0n0 to TO0n5 pin outputs that were forcibly stopped by ESOn pin input. Output is resumed by writing “1” to the TORSn bit. **Cautions1. If the level is set to the ESOn pin input level (TOMR register TOEDG1 bit = 1, TOEDG0 bit = 0 or 1), the output disabled state is not released (TOSTAn bit = 1) even if “1” is written to the TORSn bit while output is disabled (TOSTAn bit = 1).** **If the input level is the inactive level, the output disabled state is released (TOSTAn bit = 0).** **2. If the edge is set to the ESOn pin input (TOEDG1 bit = 0, TOEDG0 bit = 0 or 1), the output disabled state is released (TOSTAn bit = 0) when “1” is written to the TORSn bit while output is disabled (TOSTAn bit = 1).** **3. After reset, be sure to write “1” to the TORSn bit prior to starting TO0n0 to TO0n5 output. “0” is read when the TORSn bit is read.**
0	TOSTAn	Flag indicating TO0n0 to TO0n5 pin output status according to ES0n pin input 0: Output enabled status 1: Output disabled status

Remark n = 0, 1

(4) Timer output mode registers 0, 1 (TOMR0, TOMR1)

The TOMRn register controls timer output from the TO0n0 to TO0n5 pins.

To prevent abnormal output from the TO0n0 to TO0n5 pins due to illegal access, data is written to the TOMRn register in the following two sequences.

(a) Write access to the TOMR write enable register (SPECn), followed by
(b) Write access to the TOMRn register

Write is not enabled via hardware unless these two sequences are implemented.

TOMRn can be read/written in 8-bit units.

Caution When interrupt requests are generated during write access to the TOMRn register (after write access to the SPECn register and prior to writing to the TOMRn register), write processing to the TOMRn register may not be performed normally if access to other addresses is performed using the internal bus during servicing of these interrupts. Add one of the following processing items during the TOMRn register write routine.

- **Prior to write access to the TOMRn register, disable acknowledgment of all interrupts of the CPU.**
- **Following write access to the TOMRn register, check that write was performed normally.**

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
TOMR0	ALVTO	ALVUB	ALVVB	ALVWB	TOSP	0	TOEDG1	TOEDG0	FFFFF57DH	00H

	7	6	5	4	3	2	1	0	Address	After reset
TOMR1	ALVTO	ALVUB	ALVVB	ALVWB	TOSP	0	TOEDG1	TOEDG0	FFFFF5BDH	00H

Bit position	Bit name	Function
7	ALVTO	Specifies the active level of the TO0n0, TO0n2, and TO0n4 pins. 0: Active level is low level 1: Active level is high level **Caution Changing the ALVTO bit during TM0n operation (TM0CEn = 1) is prohibited.**
6	ALVUB	Specifies the output level of the TO0n1 pin. 0: Inverted level of active level set by ALVTO bit 1: Active level set by ALVTO bit When ALVUB = 1, the output level of TO0n1 output is the same as TO0n0. **Caution Changing the ALVUB bit during TM0n operation (TM0CEn = 1) is prohibited.**

Remark n = 0, 1

(2/2)

Bit position	Bit name	Function
5	ALVVB	Specifies the output level of the TO0n3 pin. 0: Inverted level of active level set by ALVTO bit 1: Active level set by ALVTO bit When ALVVB = 1, the output level of TO0n3 output is the same as TO0n2. **Caution Changing the ALVVB bit during TM0n operation (TM0CEn = 1) is prohibited**
4	ALVWB	Specifies the output level of the TO0n5 pin. 0: Inverted level of active level set by ALVTO bit 1: Active level set by ALVTO bit When ALVWB = 1, the output level of TO0n5 output is the same as TO0n4. **Caution Changing the ALVWB bit during TM0n operation (TM0CEn = 1) is prohibited.**
3	TOSP	Controls TO0n0 to TO0n5 pin output stop via ESOn pin input. 0: Enables ESOn pin input 1: Disables ESOn pin input **Cautions 1. The output stop status can be released by writing "1" to the TORSn bit of the TUC0n register. The operation continues even if output is prohibited for all timers and counters.** **2. Before changing the ESOn pin input status from disabled to enabled (changing the TOSP bit from 1 to 0), write "1" to the TORSn bit of the TUCn register to reset the ESOn pin input status.**
1, 0	TOEDG1, TOEDG0	These bits select the valid edge or level when setting forcible stop of TO0n0 to TO0n5 output via ESOn pin input using the TOSP bit.

TOEDG1	TOEDG0	Operation
0	0	Rising edge
0	1	Falling edge
1	0	Low level
1	1	High level

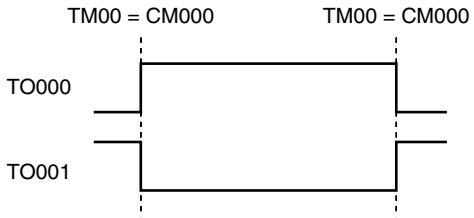
Cautions 1. Changing the TOEDG1 and TOEDG0 bits during TM0n operation (TM0CEn = 1) is prohibited.
2. Before changing the settings of bits TOEDG1 and TOEDG0, write "1" to the TORSn bit of the TUC0n register to reset the ESOn pin input status.

Remark n = 0, 1

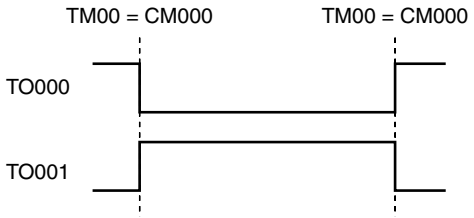
Examples of the output waveforms of TO000 and TO001 when the higher 4 bits (ALVTO, ALVUB, ALVVB, and ALVWB) of the TOMRn register are set in PWM mode 0 (asymmetric triangular waves) are shown below.

Figure 9-7. Output Waveforms of TO000 and TO001 in PWM Mode 0 (Symmetric Triangular Waves) (Without Dead Time (TM0CED0 Bit = 1))

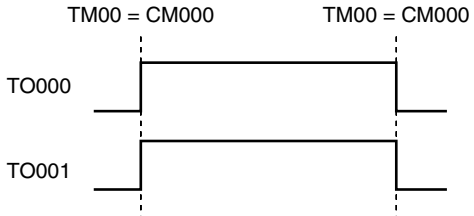
(a) TOMR0 register value = 80H



(b) TOMR0 register value = 00H



(c) TOMR0 register value = C0H



(d) TOMR0 register value = 40H

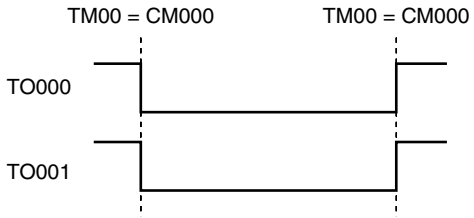
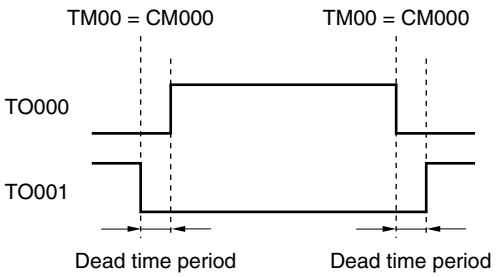
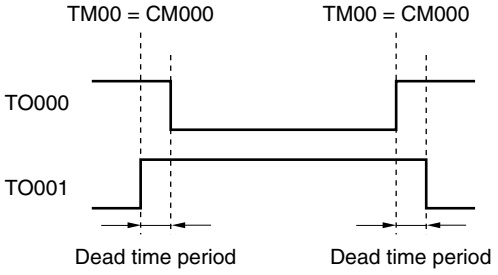


Figure 9-8. Output Waveforms of TO000 and TO001 in PWM Mode 0 (Symmetric Triangular Waves) (With Dead Time (TM0CED0 Bit = 0))

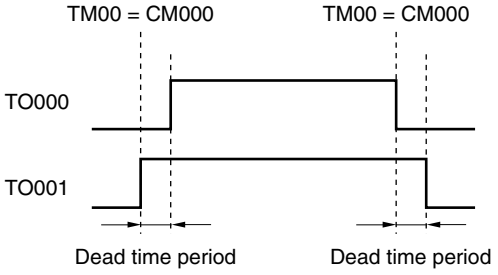
(a) TOMR0 register value = 80H



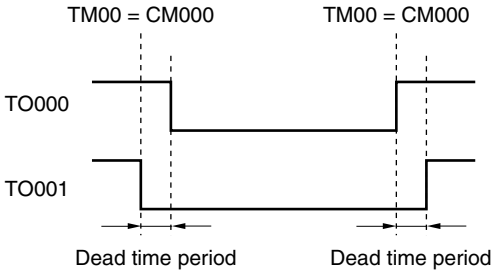
(b) TOMR0 register value = 00H



(c) TOMR0 register value = C0H



(d) TOMR0 register value = 40H



Data is set to timer output mode registers 0 and 1 (TOMR0, TOMR1) in the following sequence.

<1> Prepare the data to be set to timer output mode registers 0 and 1 (TOMR0, TOMR1) in a general-purpose register.
<2> Write data to TOMR write enable registers 0 and 1 (SEPC0, SPEC1).
<3> Set timer output mode registers 0 and 1 (TOMR0, TOMR1) (using the following instructions).
- Store instruction (ST/SST instructions)
- Bit manipulation instruction (SET1/CLR1/NOT1 instructions)

[Description Example]

```
<1> MOV     0x04, r10
<2> ST.B    r10, SPECn [r0]
<3> ST.B    r10, TOMRn [r0]
```

Remark n = 0, 1

To read the TOMRn register, no special sequence is required.

Cautions 1. Prohibit interrupts between SPECn issuance (<2>) and the TOMRn register write that immediately follows (<3>).
2. The data written to the SPECn register is dummy data; use the same register as the general-purpose register used to set the TOMRn register (<3> in the above example) for SPECn register write (<2> in the above example). The same applies when using a general-purpose register for addressing.
3. Do not write to the SPECn register or TOMRn register using DMA transfer.

(5) PWM output enable registers 0, 1 (POER0, POER1)

The POERn register is used to make the external pulse output (TO0n0 to TO0n5) status inactive by software. POERn can be read/written in 8-bit or 1-bit units.

	7	6	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset
POER0	0	0	OE210	OE200	OE110	OE100	OE010	OE000	FFFFF57FH	00H

	7	6	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset
POER1	0	0	OE211	OE201	OE111	OE101	OE011	OE001	FFFFF5BFH	00H

Bit position	Bit name	Function
5	OE21n	Specifies the output status of the TO0n5 pin. 0: TO0n5 output status is high impedance. 1: TO0n5 output status is controlled by TM0CEn bit of TMC0n register and TORTOn bit of PSTOn register and ESOn pin.
4	OE20n	Specifies the output status of the TO0n4 pin. 0: TO0n4 output status is high impedance. 1: TO0n4 output status is controlled by TM0CEn bit of TMC0n register and TORTOn bit of PSTOn register and ESOn pin.
3	OE11n	Specifies the output status of the TO0n3 pin. 0: TO0n3 output status is high impedance. 1: TO0n3 output status is controlled by TM0CEn bit of TMC0n register and TORTOn bit of PSTOn register and ESOn pin.
2	OE10n	Specifies the output status of the TO0n2 pin. 0: TO0n2 output status is high impedance. 1: TO0n2 output status is controlled by TM0CEn bit of TMC0n register and TORTOn bit of PSTOn register and ESOn pin.
1	OE01n	Specifies the output status of the TO0n1 pin. 0: TO0n1 output status is high impedance. 1: TO0n1 output status is controlled by TM0CEn bit of TMC0n register and TORTOn bit of PSTOn register and ESOn pin.
0	OE00n	Specifies the output status of the TO0n0 pin. 0: TO0n0 output status is high impedance. 1: TO0n0 output status is controlled by TM0CEn bit of TMC0n register and TORTOn bit of PSTOn register and ESOn pin.

Remark n = 0, 1

(6) PWM software timing output registers 0, 1 (PSTO0, PSTO1)

The PSTOn register is used to perform settings to output the desired waveforms to the external pulse output pins (TO0n0 to TO0n5) by software.

PSTOn can be read/written in 8-bit or 1-bit units.

Cautions 1. When the value of the TORTOn bit has been changed from 0 to 1 during timer output (setting changed to software output), the timing is delayed by the dead-time portion when the output level differs from the timer output signal during output due to the settings of the UPORTn, VPORTn, and WPORTn bits.

When the output level is the same as the timer output signal during output due to the settings of the UPORTn, VPORTn, and WPORTn bits, output is performed maintaining the same output level.

2. If software output is enabled (TORTOn bit = 1), the INTTM0n and INTCM0n3 interrupts and TO0n0 to TO0n5 output statuses are as follows during TM0n operation (TM0CEn bit = 1).

INTTM0n and INTCM0n3 interrupts: Continue occurring at each timing in accordance with timer and compare operations.

TO0n0 to TO0n5 outputs: Software output has priority.

3. If the TORTOn bit is changed from 1 to 0 during TM0n operation (TM0CEn bit = 1), the software output state is retained for the TO0n0 to TO0n5 outputs until one of the set/reset condition of the flip-flop for the TO0n0 to TO0n5 outputs shown in (a) below is generated.

(a) Set/reset conditions of flip-flop for TO0n0 to TO0n5 outputs

	Output Status	Operation Mode	Conditions
Set	Timer output	Triangular wave mode (PWM mode 0, 1)	Compare match while TM0n is counting up
		Sawtooth wave mode (PWM mode 2)	Match between TM0n and CM0n3 registers
	Software output	–	Set (to 1) UPORTn, VPORTn, and WPORTn bits
Reset	Timer output	Triangular wave mode (PWM mode 0, 1)	Compare match while TM0n is counting down
		Sawtooth wave mode (PWM mode 2)	Compare match with TM0n
	Software output	–	Clear (to 0) UPORTn, VPORTn, and WPORTn bits

Remark n = 0, 1

4. If the same value is written to the UPORTn (VPORTn, WPORTn) bit when TORTOn =1, the TO0n0 and TO0n1 outputs (TO0n2 and TO0n3, TO0n4 and TO0n5) are not changed.

(1/2)

	<7>	6	5	4	3	<2>	<1>	<0>	Address	After reset
PSTO0	TORTO0	0	0	0	0	UPORT0	VPORT0	WPORT0	FFFFF57EH	00H

	<7>	6	5	4	3	<2>	<1>	<0>	Address	After reset
PSTO1	TORTO1	0	0	0	0	UPORT1	VPORT1	WPORT1	FFFFF5BEH	00H

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7</td><td>TORTOn</td><td>Specifies TO0n0 to TO0n5 output control.
0: Timer output
1: Software output
The change of the TO0n0 to TO0n5 signals during software output occurs when the TORTOn bit is set (to 1) and a value is written to the UPORTn, VPORTn, and WPORTn bits. A dead-time timer can also be used.</td></tr>
<tr><td>2</td><td>UPORTn</td><td>Specifies the TO0n0 (U phase)/TO0n1 ($\overline{U}$ phase) pin output value.
<table>
<tr><th>UPORTn</th><th colspan="3">Operation</th></tr>
<tr><td rowspan="3">0</td><td>TO0n0</td><td colspan="2">Inverted level of ALVTO bit setting</td></tr>
<tr><td rowspan="2">TO0n1</td><td>When ALVUB = 0</td><td>Level of ALVTO bit setting</td></tr>
<tr><td>When ALVUB = 1</td><td>Inverted level of ALVTO bit setting</td></tr>
<tr><td rowspan="3">1</td><td>TO0n0</td><td colspan="2">Level of ALVTO bit setting</td></tr>
<tr><td rowspan="2">TO0n1</td><td>When ALVUB = 0</td><td>Inverted level of ALVTO bit setting</td></tr>
<tr><td>When ALVUB = 1</td><td>Level of ALVTO bit setting</td></tr>
</table>
Caution If the UPORTn bit setting value is changed when TORTOn = 1, the dead-time setting becomes valid for the TO0n0/TO0n1 output signal in the same way as during normal timer operation.</td></tr>
<tr><td>1</td><td>VPORTn</td><td>Specifies the TO0n2 (V phase)/TO0n3 ($\overline{V}$ phase) pin output value.
<table>
<tr><th>VPORTn</th><th colspan="3">Operation</th></tr>
<tr><td rowspan="3">0</td><td>TO0n2</td><td colspan="2">Inverted level of ALVTO bit setting</td></tr>
<tr><td rowspan="2">TO0n3</td><td>When ALVVB = 0</td><td>Level of ALVTO bit setting</td></tr>
<tr><td>When ALVVB = 1</td><td>Inverted level of ALVTO bit setting</td></tr>
<tr><td rowspan="3">1</td><td>TO0n2</td><td colspan="2">Level of ALVTO bit setting</td></tr>
<tr><td rowspan="2">TO0n3</td><td>When ALVVB = 0</td><td>Inverted level of ALVTO bit setting</td></tr>
<tr><td>When ALVVB = 1</td><td>Level of ALVTO bit setting</td></tr>
</table>
Caution If the VPORTn bit setting value is changed when TORTOn = 1, the dead-time setting becomes valid for the TO0n2/TO0n3 output signal in the same way as during normal timer operation.</td></tr>
</table>

Remark n = 0, 1
ALVTO bit: Bit 7 of the TOMRn register
ALVUB bit: Bit 6 of the TOMRn register
ALVVB bit: Bit 5 of the TOMRn register

(2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>0</td><td>WPORTn</td><td>Specifies the TO0n4 (W phase)/TO0n5 ($\overline{W}$ phase) pin output value.
<table>
<tr><th>WPORTn</th><th colspan="3">Operation</th></tr>
<tr><td rowspan="3">0</td><td rowspan="3">TO0n4</td><td colspan="2">Inverted level of ALVTO bit setting</td></tr>
<tr><td rowspan="2">TO0n5</td><td>When ALVWB = 0</td><td>Level of ALVTO bit setting</td></tr>
<tr><td>When ALVWB = 1</td><td>Inverted level of ALVTO bit setting</td></tr>
<tr><td rowspan="3">1</td><td rowspan="3">TO0n4</td><td colspan="2">Inverted level of ALVTO bit setting</td></tr>
<tr><td rowspan="2">TO0n5</td><td>When ALVWB = 0</td><td>Inverted level of ALVTO bit setting</td></tr>
<tr><td>When ALVWB = 1</td><td>Level of ALVTO bit setting</td></tr>
</table>
Caution If the WPORTn bit setting value is changed when TORTOn = 1, the dead-time setting becomes valid for the TO0n4/TO0n5 output signal in the same way as during normal timer operation.</td></tr>
</table>

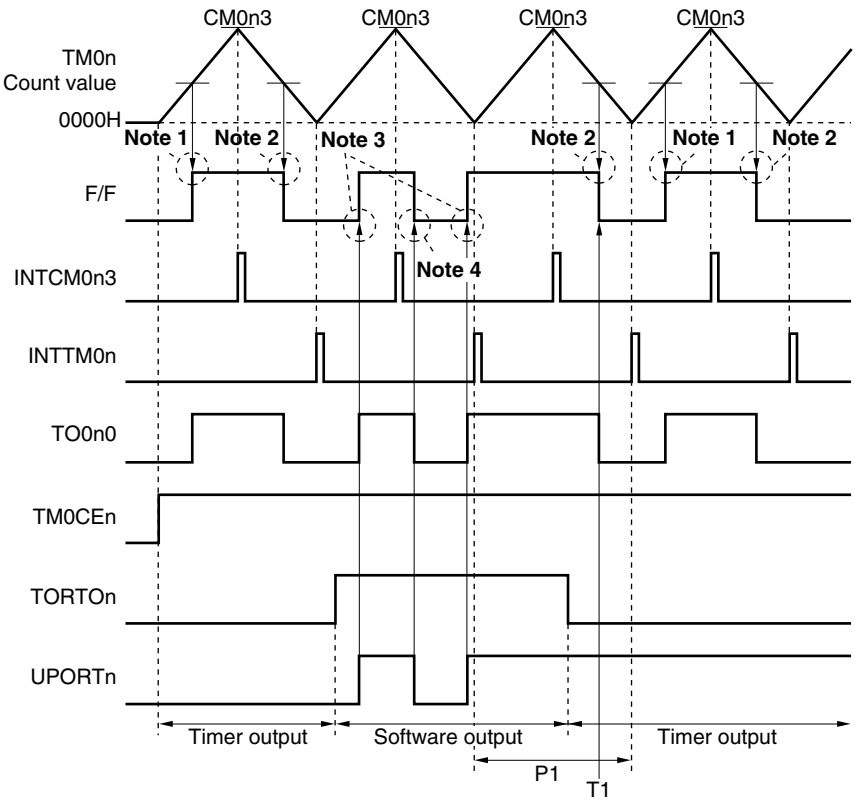
Remark n = 0, 1
ALVTO bit: Bit 7 of the TOMRn register
ALVWB bit: Bit 4 of the TOMRn register

The TO0n0 to TO0n5 pins can be set to timer output by a match between TM0n and the compare register or to software output using the PSTOn register (TORTOn bit = 1). Software output has the priority over timer output.

Consequently, when the setting changes from TM0CEn = 1 (timer operation enabled), TORTOn = 1 (software output enabled) to TM0CEn = 1 (timer operation enabled), TORTOn = 0 (software output disabled), the TO0n0 to TO0n5 pins continue to perform software output until the occurrence of the first F/F set/reset due to a match between TM0n and the compare register after the TORTOn bit setting changes.

The relationship between the settings of the TORTOn and TM0CEn bits when ALVTO = 1 and the output of TO0n0 (negative phase side) is shown on the following pages (the positive phase side (TO0n1, TO0n3, and TO0n5) is dependent on the ALVUB, ALVVB, and ALVWB bits, so refer to the explanations of each of these bits).

Figure 9-9. When UPORTn = 1 Is Set Immediately Before TORTOn = 0 (Switched by Active Value)



Notes 1. F/F set by compare match during up count

2. F/F reset by compare match during down count

3. F/F set by writing UPORTn bit

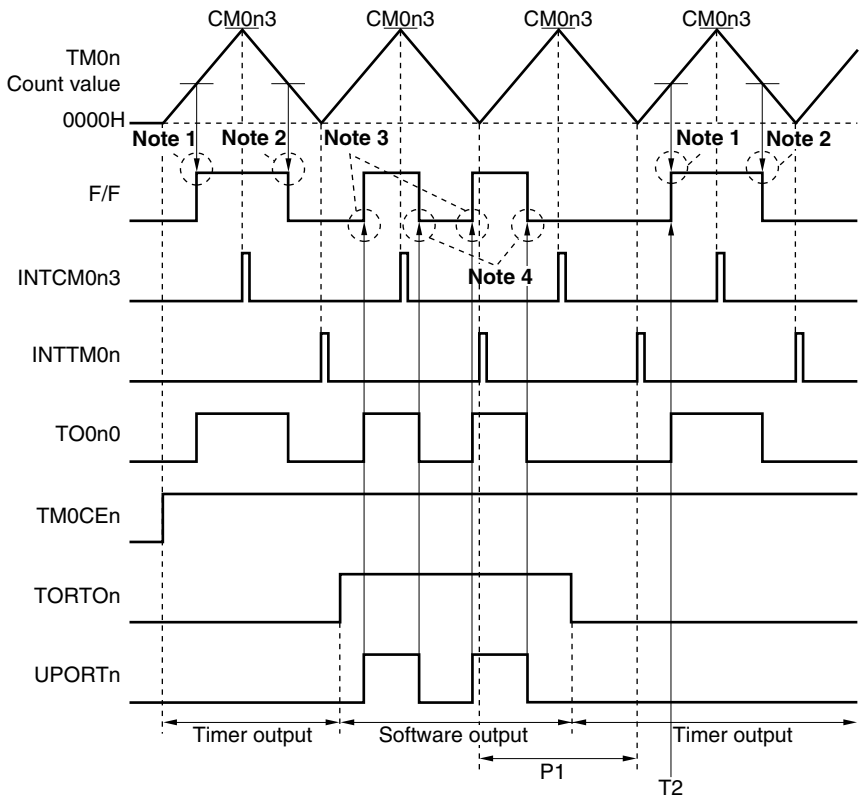
4. F/F reset by writing UPORTn bit

Remark n = 0, 1

If the setting of the TORTOn bit changes from 1 to 0 while the UPORTn bit is set to 1 in the P1 period in Figure 9-9 above, the F/F continues to hold the TORTOn bit setting of “1” until the T1 timing.

However, because the F/F is reset at the T1 timing (by a compare match of TM0n during down counting), the TO0n0 output changes from 1 to 0.

Figure 9-10. When UPORTn = 0 Is Set Immediately Before TORTOn = 0 (Switched by Inactive Value)



Notes 1. F/F set by compare match during up count

2. F/F reset by compare match during down count

3. F/F set by writing UPORTn bit

4. F/F reset by writing UPORTn bit

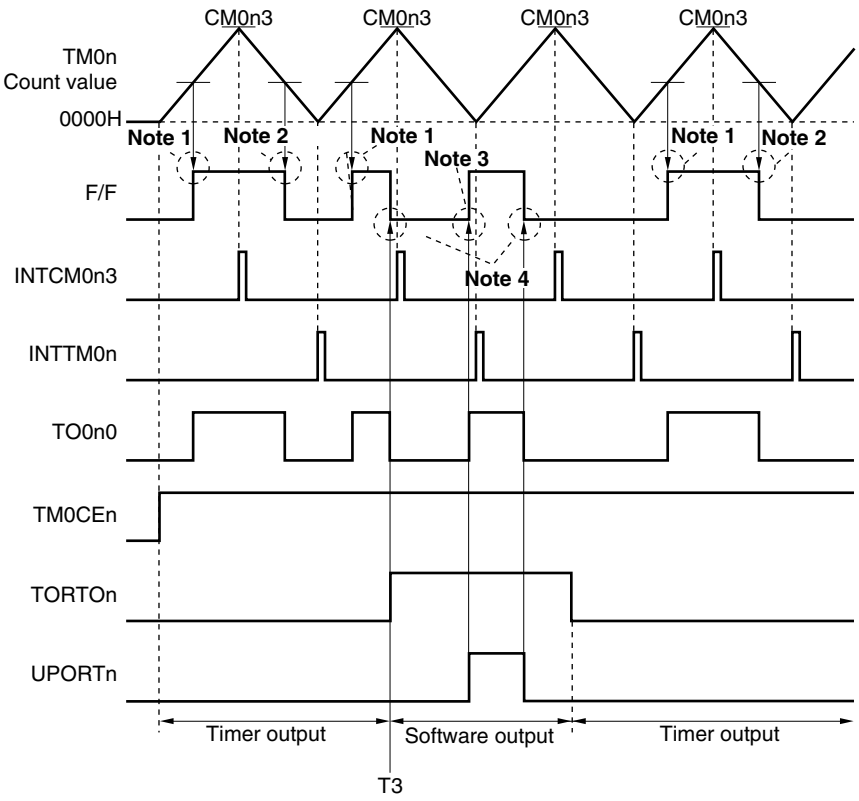
Remark n = 0, 1

If the setting of the TORTOn bit changes from 1 to 0 while the UPORTn bit is set to 0 in the P1 period in Figure 9-10 above, the F/F continues to hold the TORTOn bit setting of “0” until the T2 timing.

However, because the F/F is set at the T2 timing (by a compare match of TM0n during up counting), the TO0n0 output changes from 1 to 0.

Note that TO0n0 to TO0n5 output will stop if the TORTOn bit setting is changed from 1 to 0 while the TM0CEn bit is 0.

Figure 9-11. When UPORTn = 0 Is Set Immediately Before TORTOn = 1



Notes 1. F/F set by compare match during up count

2. F/F reset by compare match during down count

3. F/F set by writing UPORTn bit

4. F/F reset by writing UPORTn bit

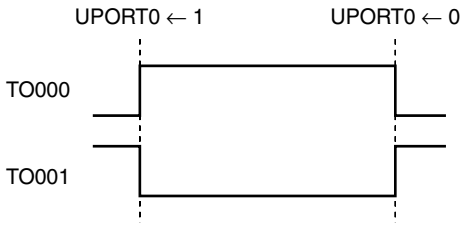
Remark n = 0, 1

If the setting of the TORTOn bit changes from 0 to 1 while the UPORTn bit is set to 0 during TM0n operation (TM0CEn = 1), the TO0n0 output changes from 1 to 0 because the F/F is reset at the T3 timing.

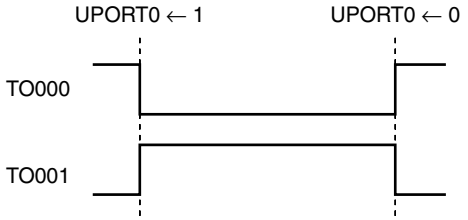
Examples of the software output waveforms of TO000 and TO001 based on the settings of the TORTOn, UPORTn, VPORTn, and WPORTn bits are shown on the following pages.

Figure 9-12. Software Output Waveforms of TO000 and TO001 (Without Dead Time (TM0CED0 = 1))

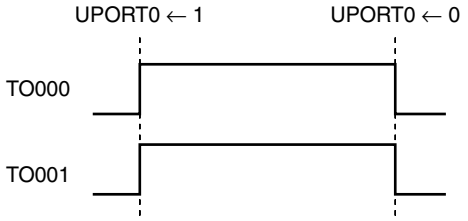
(a) TOMR0 register value = 80H



(b) TOMR0 register value = 00H



(c) TOMR0 register value = C0H



(d) TOMR0 register value = 40H

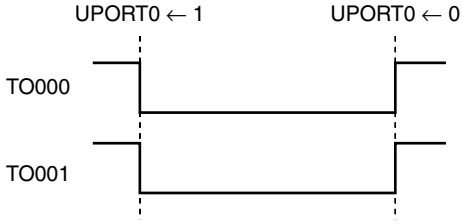
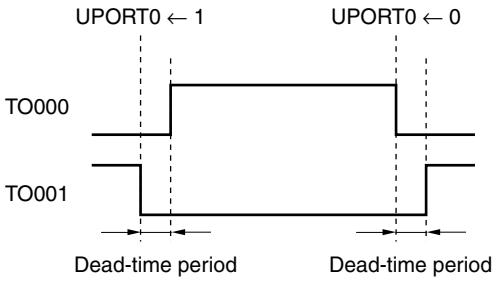
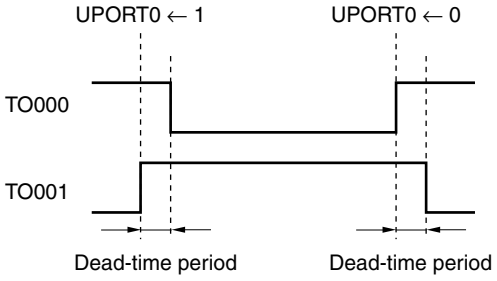


Figure 9-13. Software Output Waveforms of TO000 and TO001 (With Dead Time (TM0CED0 = 0))

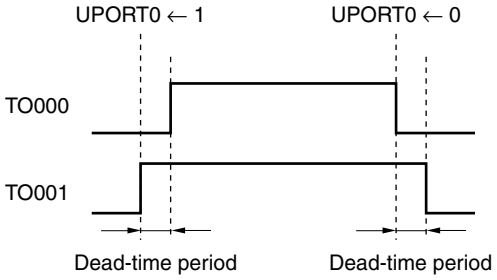
(a) TOMR0 register value = 80H



(b) TOMR0 register value = 00H



(c) TOMR0 register value = C0H



(d) TOMR0 register value = 40H

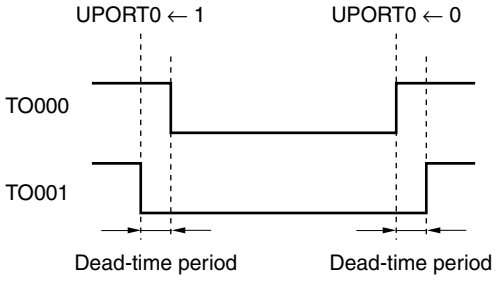
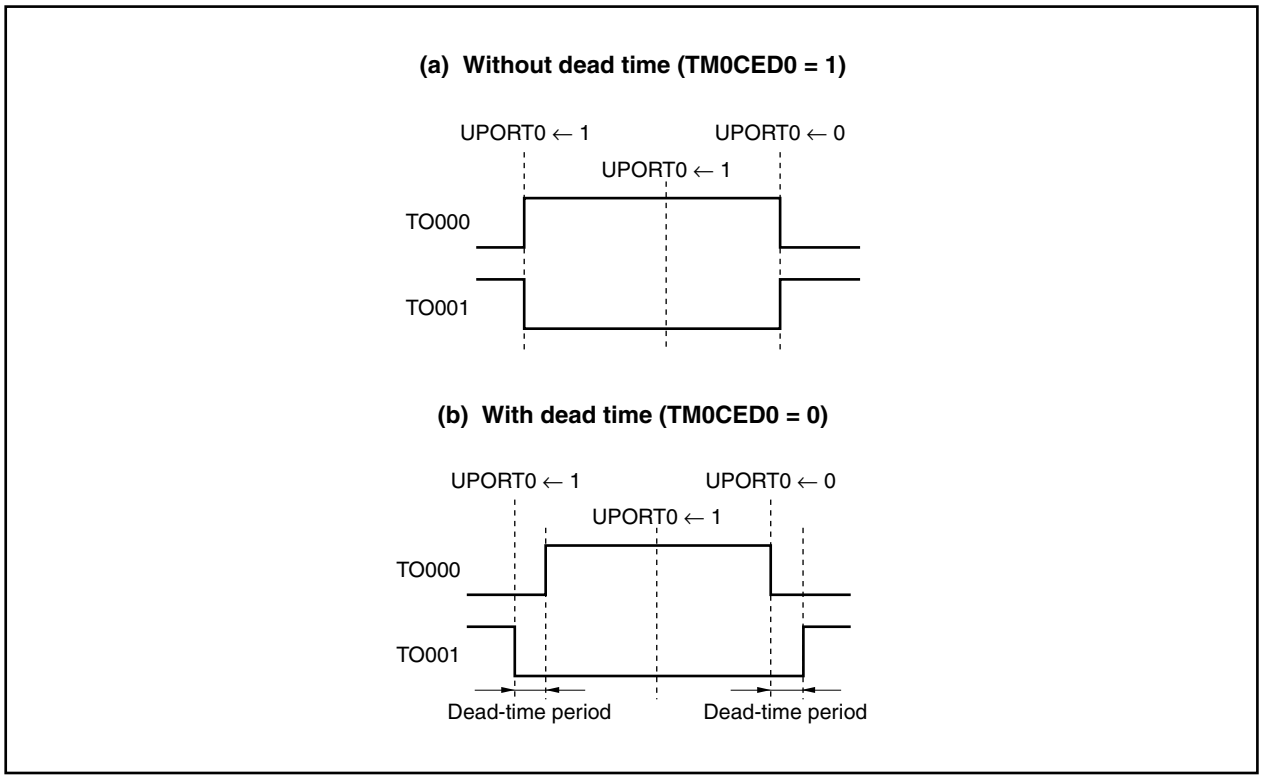


Figure 9-14. Software Output Waveforms of TO000 and TO001 When "1" Is Written to UPORT0 Bit While TORTO0 = 1 (When TOMR0 Register Value = 80H)



The following table shows the output status of external pulse output (in the case of TO0n0).

Table 9-3. Output Status of External Pulse Output (In Case of TO0n0)

OE00n Bit	TORTOn, UPORTn Bits	TM0CEn Bit	TO0n0
0	0/1	0/1	High impedance
1	0	0	High impedance
		1	Timer output
	1	0/1	Output by UPORTn bit

Remarks 1. OE00n bit: Bit 0 of POERn register
TORTOn bit: Bit 7 of PSTOn register
UPORTn bit: Bit 2 of PSTOn register
TM0CEn bit: Bit 15 of TMC0n register

2. n = 0, 1

(7) TOMR write enable registers 0, 1 (SPEC0, SPEC1)

The SPECn register enables writing to the TOMRn register. Unless writing to the TOMRn register is performed immediately after writing to the SPECn register (any data can be written), write processing to the TOMRn register is not performed normally. Normally, 0000H is read.

The SPECn register can be read/written in 16-bit units.

Remark n = 0, 1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
SPEC0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FFFFF580H	0000H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
SPEC1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FFFFF5C0H	0000H

9.1.6 Operation

Remarks 1. In the explanation of operations in this section, the bits that affect the TO0n0 to TO0n5 outputs are assumed to be set as follows.

ALVTO = 1, ALVUB = 0, ALVVB = 0, ALVWB = 0, TORTOn =0

2. The F/F in this section indicates the flip-flop for controlling the output of the TO0n0 to TO0n5 pins.

(1) Basic operation

Timer 0 (TM0n) is a 16-bit interval timer that operates as an up/down timer or as an up timer. The cycle is controlled by compare register 0n3 (CM0n3) (n = 0, 1).

All TM0n bits are cleared (0) by $\overline{\text{RESET}}$ input and the count operation is stopped.

Count operation enable/disable is controlled by the TM0CEn bit of timer control register 0n (TMC0n). The count operation is started by setting the TM0CEn bit to 1 by software. Resetting the TM0CEn bit to 0 clears TM0n and stops the count operation.

When the value of compare register 0n3 (CM0n3) set beforehand and the value of the TM0n counter match, a match interrupt (INTCM0n3) is generated.

The count clock to TM0n can be selected from among 6 internal clocks using the TMC0n register. If TM0n has been set as an up/down timer, an underflow interrupt (INTTM0n) is generated when TM0n becomes 0000H during down counting.

TM0n has the following three operation modes, which are selected using timer control register 0n (TMC0n).

- PWM mode 0: Triangular wave modulation (right-left symmetric waveform control)
- PWM mode 1: Triangular wave modulation (right-left asymmetric waveform control)
- PWM mode 2: Sawtooth wave modulation control

Table 9-4. Operation Modes of Timer 0 (TM0n)

TMC0n Register		Operation Mode	TM0n Operation	Timer Clear Source	Interrupt Source	BFCMn3 → CM0n3 Timing	BFCMn0 to BFCMn2, BFCMn4, BFCMn5 → CM0n0 to CM0n2, CM0n4, CM0n5 Timing
MOD01	MOD00						
0	0	PWM mode 0 (Symmetric triangular wave)	Up/down	–	INTTM0n, INTCM010 to INTCM012, INTCM0n3 to INTCM0n5	INTTM0n	INTTM0n
0	1	PWM mode 1 (Asymmetric triangular wave)	Up/down	–	INTTM0n, INTCM010 to INTCM012, INTCM0n3 to INTCM0n5	INTTM0n	INTTM0n, INTCM0n3
1	0	PWM mode 2 (Sawtooth wave)	Up	INTCM0n3	INTCM010 to INTCM012, INTCM0n3 to INTCM0n5	INTCM0n3	INTCM0n3
1	1	Setting prohibited					

Caution Changing the MOD01 and MOD00 bits during TM0n operation (TM0CEn = 1) is prohibited.

Remark n = 0, 1

The various operation modes are described below.

(2) PWM mode 0: Triangular wave modulation (right-left symmetric waveform control)

[Setting procedure]

(a) Set PWM mode 0 (symmetric triangular wave) using the MOD01 and MOD00 bits of the TMC0n register. Also set the active level of the TO0n0 to TO0n5 pins using the ALVTO bit of the TOMRn register (n = 0, 1).

(b) Set the count clock of TM0n using the PRM02 to PRM00 bits of the TMC0n register. The transfer operation from BFCMn3 to CM0n3 is set using the BFTE3 bit, and the transfer operation from BFCMn0 to BFCMn2, BFCMn4, and BFCMn5 to CM0n0 to CM0n2, CM0n4, and CM0n5 is set using the BFTEN bit.

(c) Set the initial values.

(i) Specify the interrupt culling ratio using the CUL02 to CUL00 bits of the TMC0n register.

(ii) Set the half-cycle width of the PWM cycle in BFCMn3.
- PWM cycle = BFCMn3 value $\times$ 2 $\times$ TM0n count clock
(The TM0n count clock is set by the TMC0n register.)

(iii) Set the dead-time width in DTRRn.
- Dead-time width = (DTRRn + 1)/f_{CLK}
f_{CLK}: Base clock

(iv) Set the set/reset timing of the F/F used in the PWM cycle in BFCMn0 to BFCMn2.

(d) Clear (0) the TM0CEDn bit of the TMC0n register to enable dead-time timer operation. Set TM0CEDn = 1 when not using dead time.

(e) Setting (1) the TM0CEn bit of the TMC0n register starts TM0n counting, and a 6-channel PWM signal is output from the TO0n0 to TO0n5 pins.

Cautions 1. Setting CM0n3 to 0000H is prohibited.

2. Setting BFCMnx > BFCMn3 is prohibited when the TM0CEn bit of the TMC0n register is 0 because the outputs of the TO0n0 to TO0n5 pins are the inverted levels of the settings (x = 0 to 2). Also, setting BFCMnx > BFCMn3 is prohibited if the CM0nx register is 0 when the TM0CEn bit of the TMC0n register.

Remark The TM0CEn bit of the TMC0n register indicates a transfer operation under the following conditions.
- When TM0CEn bit of TMC0n register is 0
Transfer to the CM0n0 to CM0n2, CM0n4, and CM0n5 registers is performed at the next base clock (f_{CLK}) after writing to the BFCMn0 to BFCMn2, BFCMn4, and BFCMn5 registers.
- When TM0CEn bit of TMC0n register is 1
The value of the BFCMn0 to BFCMn2, BFCMn4, and BFCMn5 registers is transferred to the CM0n0 to CM0n2, CM0n4, and CM0n5 registers upon occurrence of the INTTM0n interrupt. Transfer enable/disable at this time is controlled by the BFTEN bit of the TMC0n register.

[Operation]

In PWM mode 0, TM0n performs up/down count operation. When TM0n = 0000H during down counting, an underflow interrupt (INTTM0n) is generated, and when TM0n = CM0n3 during up counting, a match interrupt (INTCM0n3) is generated (n = 0, 1).

Switching from up counting to down counting is performed when TM0n and CM0n3 match (INTCM0n3), and switching from down counting to up counting is performed when a TM0n underflow occurs after TM0n becomes 0000H.

The PWM cycle in this mode is (BFCMn3 value $\times$ 2 $\times$ TM0n count clock). Note that the next PWM cycle width is set to BFCMn3.

The data of BFCMn3 is automatically transferred by hardware to CM0n3 upon generation of the INTTM0n interrupt. Furthermore, calculation is performed by software processing started by INTTM0n, and the data for the next cycle is set to BFCMn3.

Data setting to CM0n0 to CM0n2, which control the PWM duty, is explained next.

Setting of data to CM0n0 to CM0n2 consists of setting the duty output from BFCMn0 to BFCMn2.

The values of BFCMn0 to BFCMn2 are automatically transferred by hardware to CM0n0 to CM0n2 upon generation of the INTTM0n interrupt. Furthermore, software processing is started up and calculation performed, and the set/reset timing of the F/F for the next cycle is set to BFCMn0 to BFCMn2.

The PWM cycle and the PWM duty are set in the above procedure.

The F/F set/reset conditions upon match of CM0n0 to CM0n2 are as follows.

- Set: CM0n0 to CM0n2 match detection during TM0n up count operation
- Reset: CM0n0 to CM0n2 match detection during TM0n down count operation

In this mode, the F/F set/reset timing is performed at the same timing (right-left symmetric control). The values of DTRRn are transferred to the corresponding dead-time timers (DTMn0 to DTMn2) in synchronization with the set/reset timing of the F/F, and down counting is started. DTMn0 to DTMn2 count down to 000H, and stop when they count down further to FFFH.

DTMn0 to DTMn2 can automatically generate a width at which the active levels of the positive phase (TO0n0, TO0n2, TO0n4) and negative phase (TO0n1, TO0n3, TO0n5) do not overlap (dead time).

In this way, software processing is started by an interrupt (INTTM0n) that occurs once during every PWM cycle after initial setting has been performed, and by setting the PWM cycle and PWM duty to be used in the next cycle, it is possible to automatically output a PWM waveform to pins TO0n0 to TO0n5 taking into consideration the dead-time width (in the case of an interrupt culling ratio of 1/1).

[Output waveform width with respect to set value]

- PWM cycle = BFCMn3 × 2 × T_{TM0n}
- Dead-time width T_{Dnm} = (DTRRn + 1)/f_{CLK}
- Active width of positive phase (TO0n0, TO0n2, TO0n4 pins)
  = { (CM0n3 − CM0nX_{up}) + (CM0n3 − CM0nX_{down}) } × T_{TM0n} − T_{Dnm}
- Active width of negative phase (TO0n1, TO0n3, TO0n5 pins)
  = (CM0nX_{down} + CM0nX_{up}) × T_{TM0n} − T_{Dnm}
- In this mode, CM0nX_{up} = CM0nX_{down} (however, within the same PWM cycle).
  Since CM0nX_{up} and CM0nX_{down} in the negative phase formula are prepared in a separate PWM cycle, CM0nX_{up} ≠ CM0nX_{down}.

f_{CLK}: Base clock
T_{TM0n}: TM0n count clock
CM0nX_{up}: Set value of CM0n0 to CM0n2 while TM0n is counting up
CM0nX_{down}: Set value of CM0n0 to CM0n2 while TM0n is counting down

The pin level when the TO0n0 to TO0n5 pins are reset is the high impedance state. When the control mode is selected thereafter, the following levels are output until TM0n is started.

- TO0n0, TO0n2, TO0n4… When active low → High level
  When active high → Low level
- TO0n1, TO0n3, TO0n5… When active low → Low level
  When active high → High level

The active level is set with the ALVTO bit of the TOMRn register. The default is active low.

Caution If a value such that the positive phase or negative phase active width is "0" or a negative value is set in the above formula, the TO0n0 to TO0n5 pins output a waveform fixed to the inactive level waveform with active width "0".

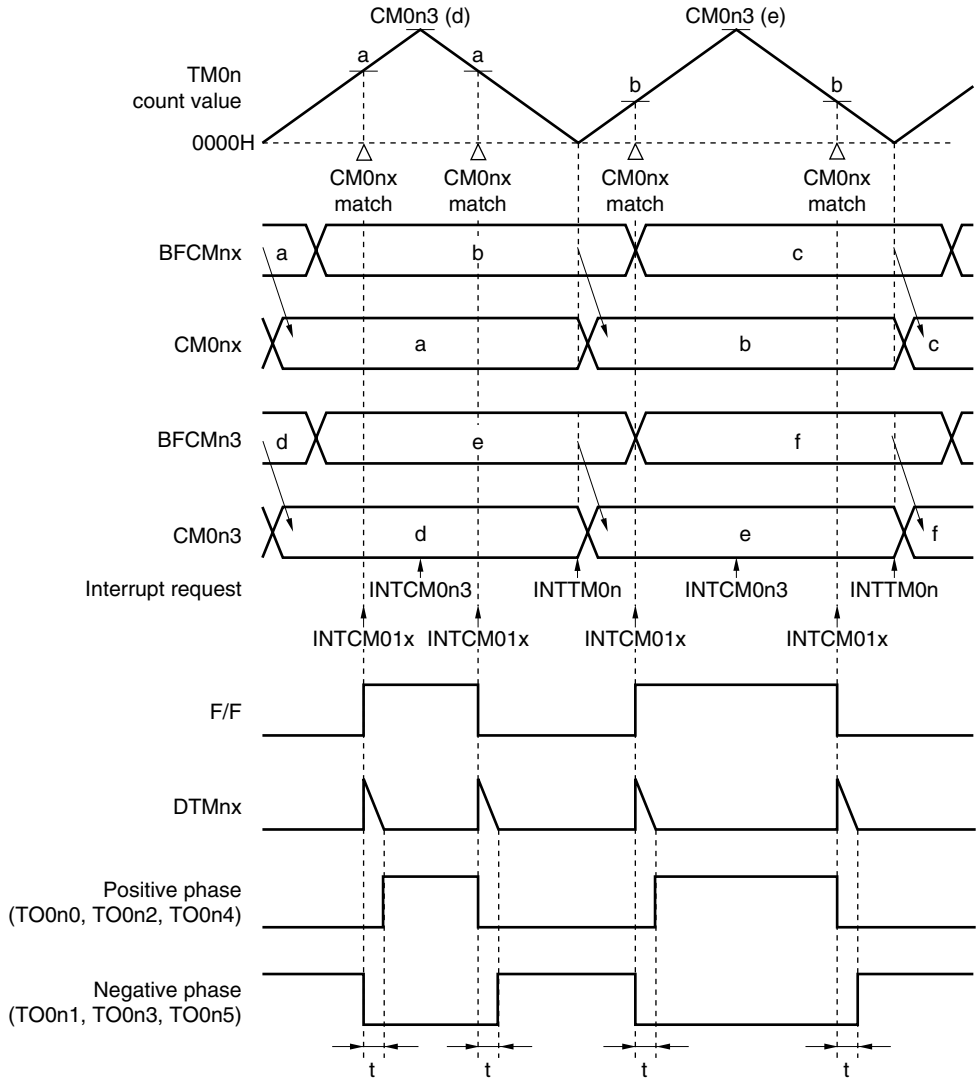
Remarks. 1 m = 0 to 2
n = 0, 1

2. The interrupt request signal occurrence conditions of INTCM010 to INTCM012, INTCM0n4, and INTCM0n5 are shown below.

Setting Condition	INTCM010 to INTCM012, INTCM0n4, INTCM0n5 Signal Occurrence Status
CM010 to CM012, CM0n4, CM0n5 ≤ CM0n3	Occurs
CM010 to CM012, CM0n4, CM0n5 = 0000H	Occurs
CM010 to CM012, CM0n4, CM0n5 > CM0n3	Does not occur

Figure 9-15. Operation Timing in PWM Mode 0 (Symmetric Triangular Wave) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Remarks 1. The above figure shows the timing chart when both BFTE3 and BFTEN of the TMC0n register are 1, and transfer from BFCMn3 to CM0n3, or from BFCMnx to CM0nx is enabled. Transfer is not performed when BFTE3 = 0 or BFTEN = 0.

2. n = 0, 1

3. x = 0 to 2

4. t: Dead time = (DTRRn + 1)/f_{CLK} (f_{CLK}: Base clock)

5. To not use dead time, set the TM0CEDn bit of the TMC0n register to 1.

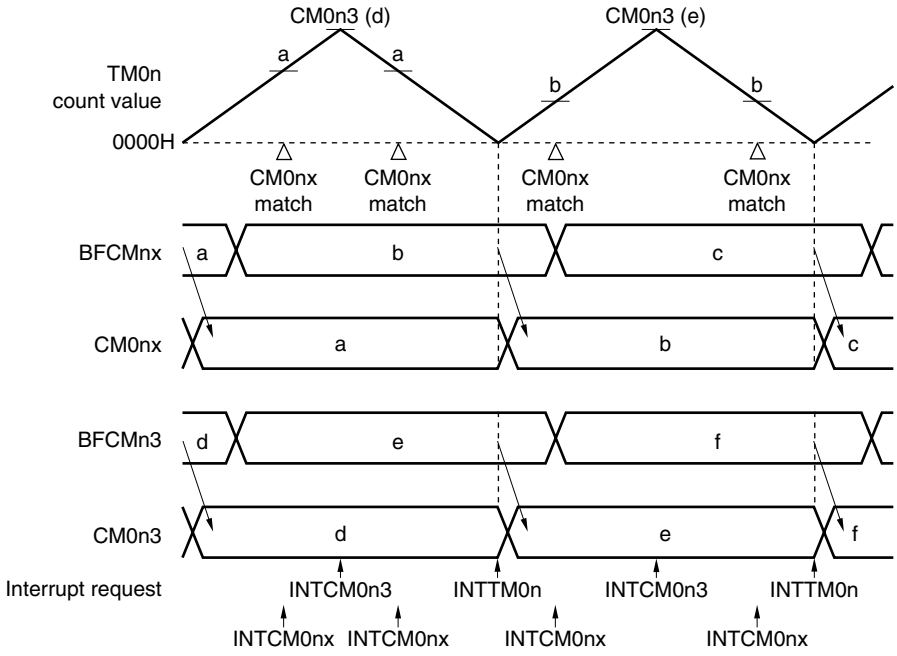
6. The above figure shows an active-high case.

7. INTCM01x is generated on a match between TM01 and CM01x (a and b in the above figure). INTCM00x is not generated.

Figure 9-16 shows the overall operation image.

Figure 9-15. Operation Timing in PWM Mode 0 (Symmetric Triangular Wave) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. The above figure shows the timing chart when both BFTE3 and BFTEN of the TMC0n register are 1, and transfer from BFCMn3 to CM0n3, or from BFCMnx to CM0nx is enabled. Transfer is not performed when BFTE3 = 0 or BFTEN = 0.

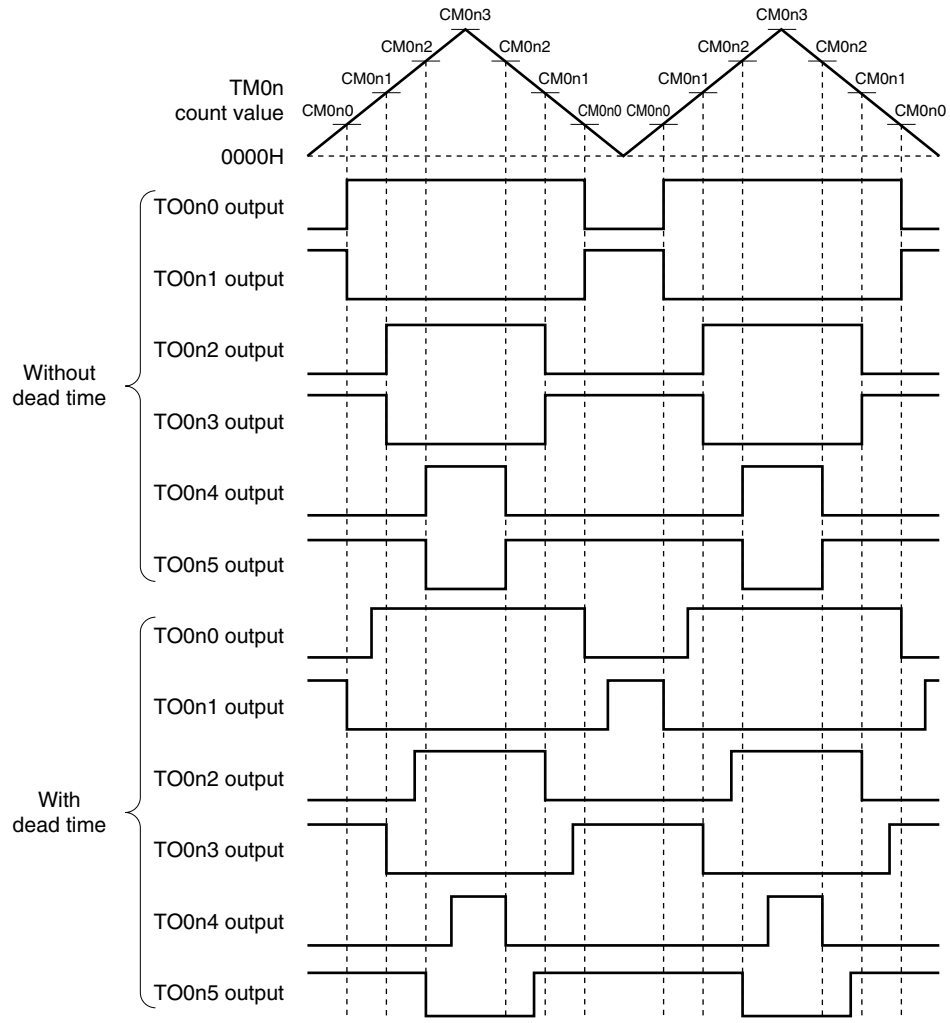
2. n = 0, 1

3. x = 4, 5

4. INTCM0nx is generated on a match between TM0n and CM0nx (a and b in the above figure).

Figure 9-16 shows the overall operation image.

Figure 9-16. Overall Operation Image of PWM Mode 0 (Symmetric Triangular Wave)



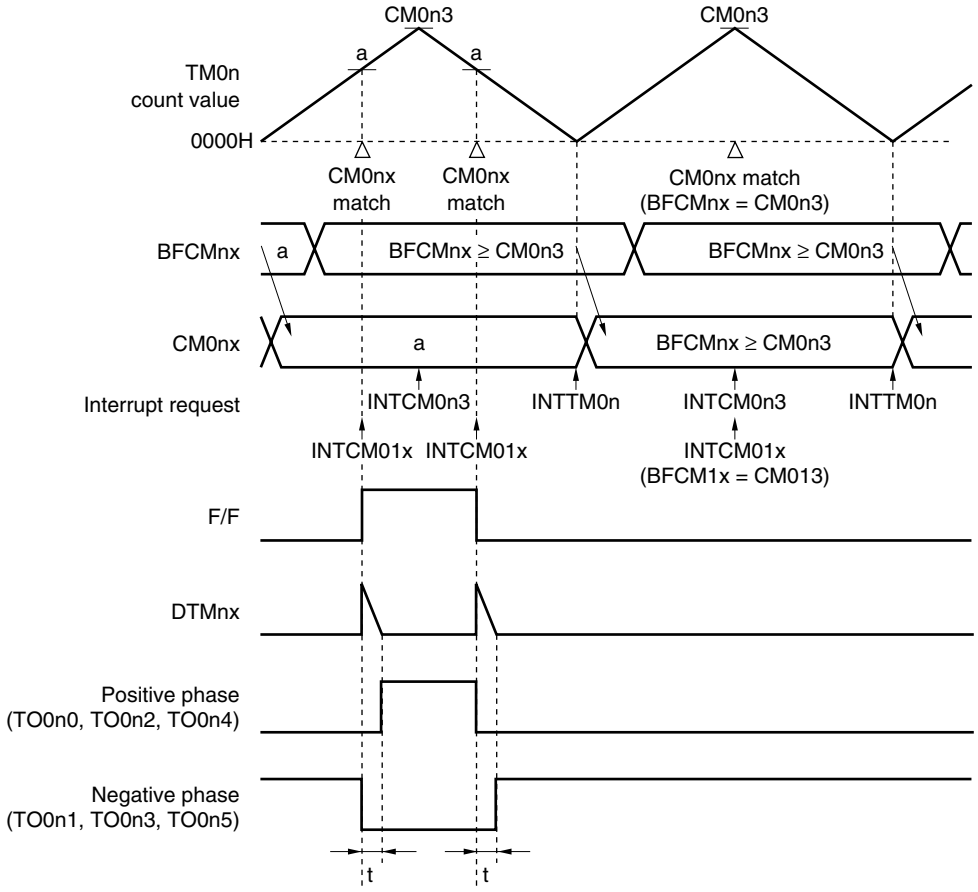
Remark n = 0, 1

Next, an example of the operation timing, which depends on the values set to CM0n0 to CM0n2, CM0n4, and CM0n5 (BFCMn0 to BFCMn2, BFCMn4, BFCMn5) is shown.

(a) When CM0nx (BFCMnx) ≥ CM0n3 is set

Figure 9-17. Operation Timing in PWM Mode 0 (Symmetric Triangular Wave, BFCMnx ≥ CM0n3) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Remarks 1. n = 0, 1

2. x = 0 to 2

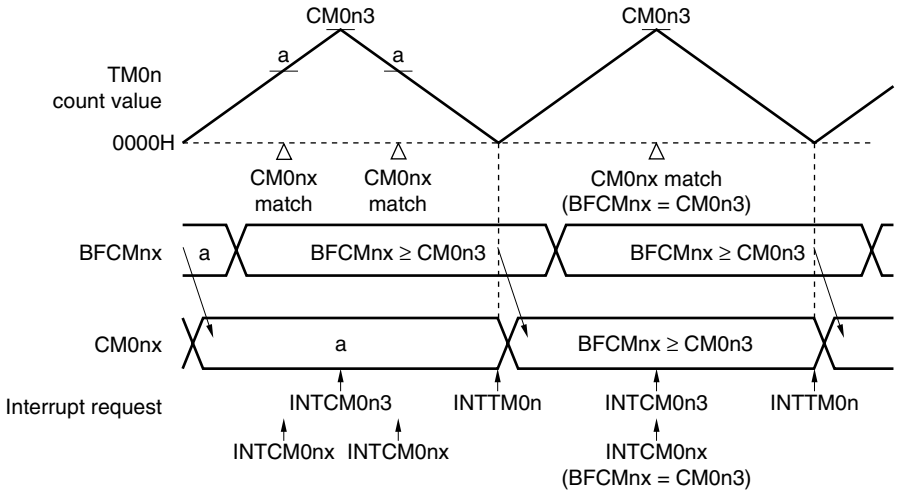
3. t: Dead time = (DTRRn + 1)/f_{CLK} (f_{CLK}: Base clock)

4. The above figure shows an active-high case.

5. INTCM01x is generated on a match between TM01 and CM01x (a in the above figure). INTCM00x is not generated.

Figure 9-17. Operation Timing in PWM Mode 0 (Symmetric Triangular Wave, BFCMnx ≥ CM0n3) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. n = 0, 1

2. x = 4, 5

3. INTCM0nx is generated on a match between TM0n and CM0nx (a in the above figure).

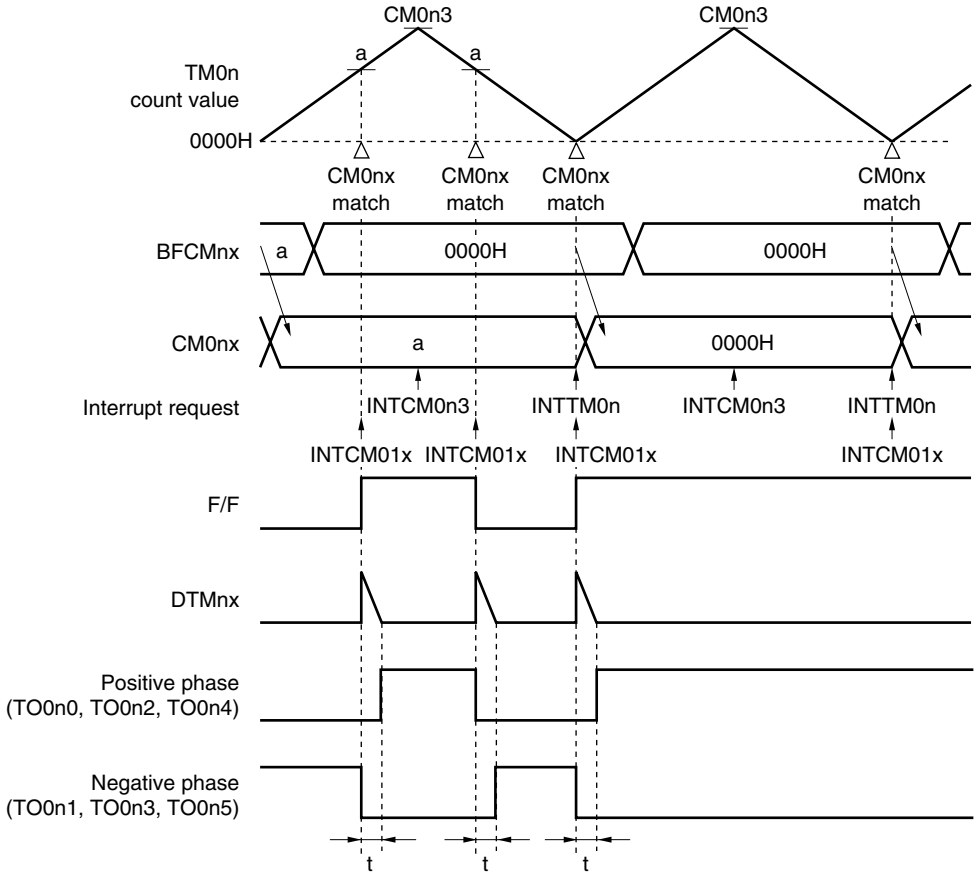
When a value greater than CM0n3 is set to BFCMn0 to BFCMn2, the positive phase side (TO0n0, TO0n2, TO0n4 pins) outputs a low level, and the negative phase side (TO0n1, TO0n3, TO0n5 pins) continues to output a high level. This feature is effective for outputting a low-level or high-level width exceeding the PWM cycle in an application such as inverter control. Furthermore, if CM0n0 to CM0n2 = CM0n3 is set, matching of TM0n and CM0n0 to CM0n2 is detected during down counting by TM0n, so that the F/F remains reset as is, and is not set.

The above explanation applies to an active high case. In an active low case, the levels of positive and negative phases are merely inverted and other operations remain the same.

(b) When CM0nx (BFCMnx) = 0000H is set

Figure 9-18. Operation Timing in PWM Mode 0 (Symmetric Triangular Wave, BFCMnx = 0000H) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Remarks 1. n = 0, 1

2. x = 0 to 2

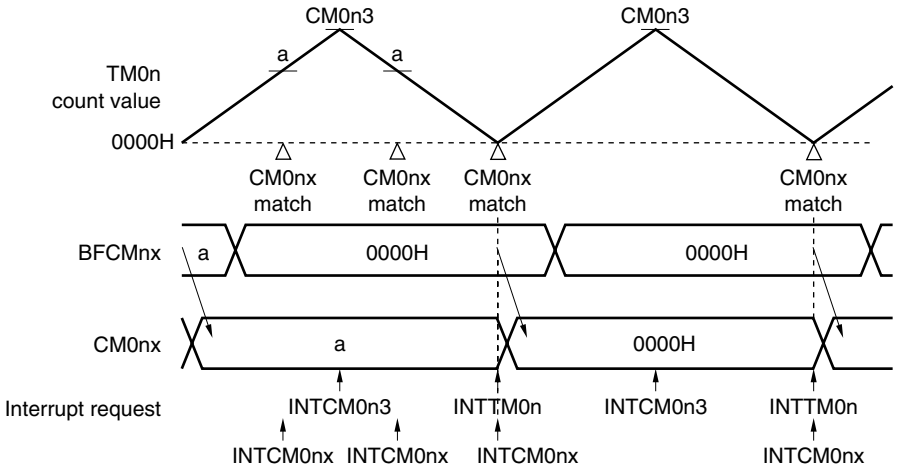
3. t: Dead time = (DTRRn + 1)/f_{CLK} (f_{CLK}: Base clock)

4. The above figure shows an active-high case.

5. INTCM01x is generated on a match between TM01 and CM01x (a in the above figure). INTCM00x is not generated.

Figure 9-18. Operation Timing in PWM Mode 0 (Symmetric Triangular Wave, BFCMnx = 0000H) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. n = 0, 1

2. x = 4, 5

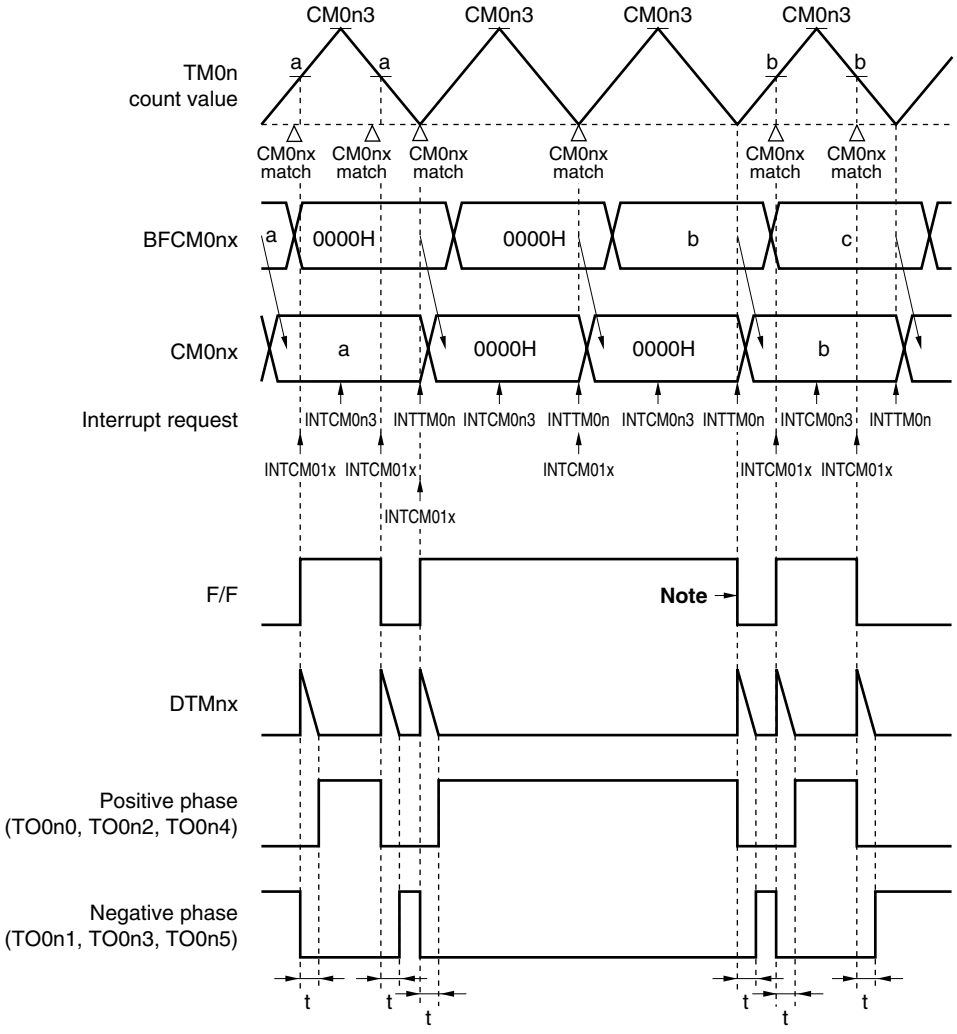
3. INTCM0nx is generated on a match between TM0n and CM0nx (a in the above figure).

Since TM0n = CM0n0 to CM0n2 = 0000H match is detected during up counting by TM0n, the F/F is just set and does not get reset. Even when the setting value is 0000H, F/F is changed in the cycle during which transfer is performed from BFCMn0 to BFCMn2 to CM0n0 to CM0n2 similarly to when the setting value is other than 0000H.

Figure 9-19 shows the change timing from the 100% duty state.

Figure 9-19. Change Timing from 100% Duty State (PWM Mode 0) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Note F/F is reset upon INTTM0n occurrence.

Remarks 1. n = 0, 1

2. x = 0 to 2

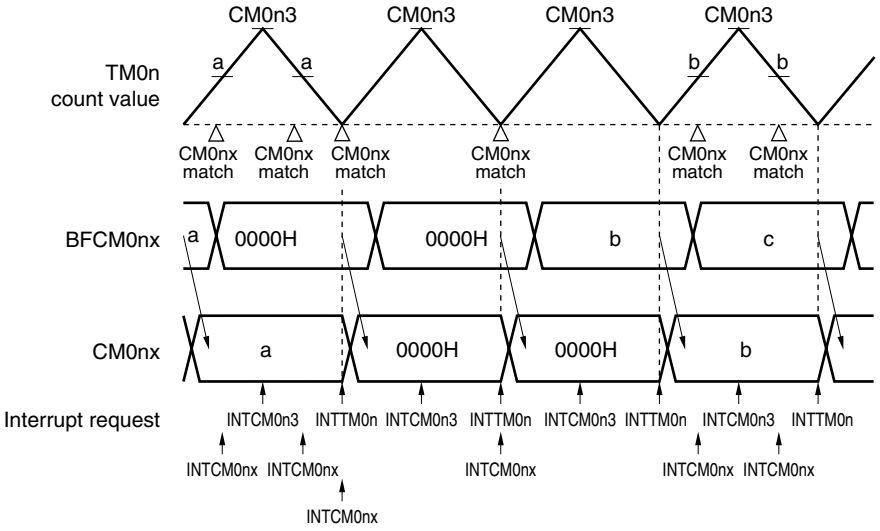
3. t: Dead time = (DTRRn + 1)/f_{CLK} (f_{CLK}: Base clock)

4. The above figure shows an active-high case.

5. INTCM01x is generated on a match between TM01 and CM01x (a and b in the above figure). INTCM00x is not generated.

Figure 9-19. Change Timing from 100% Duty State (PWM Mode 0) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. n = 0, 1

2. x = 4, 5

3. INTCM0nx is generated on a match between TM0n and CM0nx (a and b in the above figure).

(3) PWM mode 1: Triangular wave modulation (right-left asymmetric waveform control)

[Setting procedure]

(a) Set PWM mode 1 (asymmetric triangular wave) using the MOD01 and MOD00 bits of the TMC0n register. Also set the active level of the TO0n0 to TO0n5 pins using the ALVTO bit of the TOMRn register (n = 0, 1).

(b) Set the count clock of TM0n using the PRM02 to PRM00 bits of the TMC0n register. The transfer operation from BFCMn3 to CM0n3 is set using the BFTE3 bit, and the transfer operation from BFCMn0 to BFCMn2, BFCMn4, BFCMn5 to CM0n0 to CM0n2, CM0n4, and CM0n5 is set using the BFTEN bit.

(c) Set the initial values.

(i) Specify the interrupt culling ratio using the CUL02 to CUL00 bits of the TMC0n register.

(ii) Set the half-cycle width of the PWM cycle in BFCMn3.
- PWM cycle = BFCMn3 value $\times$ 2 $\times$ TM0n count clock
  (The TM0n count clock is set by the TMC0n register.)

(iii) Set the dead-time width in DTRRn.
- Dead-time width = (DTRRn + 1)/f_{CLK}
  f_{CLK}: Base clock

(iv) Set the set timing of the F/F used in the PWM cycle in BFCMn0 to BFCMn2, BFCMn4, and BFCMn5.

(d) Clear (0) the TM0CEDn bit of the TMC0n register to enable dead-time timer operation. Set TM0CEDn = 1 when not using dead time.

(e) Setting (1) the TM0CEn bit of the TMC0n register starts TM0n counting, and a 6-channel PWM signal is output from the TO0n0 to TO0n5 pins.

Caution Setting CM0n3 to 0000H is prohibited.

Remark The TM0CEn bit of the TMC0n register indicates transfer operation under the following conditions.
- When TM0CEn bit of TMC0n register is 0
  Transfer to the CM0n0 to CM0n2, CM0n4, and CM0n5 registers is performed at the next base clock (f_{CLK}) after writing to the BFCMn0 to BFCMn2, BFCMn4, and BFCMn5 registers.
- When TM0CEn bit of TMC0n register is 1
  The value of the BFCMn0 to BFCMn2, BFCMn4, and BFCMn5 registers is transferred to the CM0n0 to CM0n2, CM0n4, and CM0n5 registers upon occurrence of the INTTM0n or INTCM0n3 interrupt. Transfer enable/disable at this time is controlled by the BFTEN bit of the TMC0n register.

[Operation]

In PWM mode 1, TM0n performs up/down count operation. When TM0n = 0000H during down counting, an underflow interrupt (INTTM0n) is generated, and when TM0n = CM0n3 during up counting, a match interrupt (INTCM0n3) is generated (n = 0, 1).

Switching from up counting to down counting is performed when TM0n and CM0n3 match (INTCM0n3), and switching from down counting to up counting is performed by INTTM0n.

The PWM cycle in this mode is (BFCMn3 value $\times$ 2 $\times$ TM0n count clock). Note that the next PWM cycle width is set to BFCMn3.

The data of BFCMn3 is automatically transferred by hardware to CM0n3 upon generation of the INTTM0n interrupt. Furthermore, calculation is performed by software processing started by INTTM0n, and the data for the next cycle is set to BFCMn3.

Data setting to CM0n0 to CM0n2, which control the PWM duty, is explained next.

Setting of data to CM0n0 to CM0n2 consists of setting the duty output from BFCMn0 to BFCMn2.

The values of BFCMn0 to BFCMn2 are automatically transferred by hardware to CM0n0 to CM0n2 upon generation of INTTM0n and INTCM0n3 (TM0n and CM0n3 match interrupts). Furthermore, software processing is started up and calculation performed, and the set/reset timing of the F/F after a half cycle is set in BFCMn0 to BFCMn2.

The PWM cycle and the PWM duty are set in the above procedure.

The F/F set/reset conditions upon match of CM0n0 to CM0n2 are as follows.

- Set: CM0n0 to CM0n2 match detection during TM0n up count operation
- Reset: CM0n0 to CM0n2 match detection during TM0n down count operation

The values of DTRRn are transferred to the corresponding dead-time timers (DTMn0 to DTMn2) in synchronization with the set/reset timing of the F/F, and down counting is started. DTMn0 to DTMn2 count down to 000H, and stop when they count down further to FFFH.

DTMn0 to DTMn2 can automatically generate a width at which the active levels of the positive phase (TO0n0, TO0n2, TO0n4) and negative phase (TO0n1, TO0n3, TO0n5) do not overlap (dead time).

In this way, software processing is started by two interrupts (INTTM0n and INTCM0n3) that occur during every PWM cycle after initial setting has been performed, and by setting the PWM cycle and PWM duty to be used after a half cycle, it is possible to automatically output a PWM waveform to pins TO0n0 to TO0n5 taking into consideration the dead-time width (in the case of an interrupt culling ratio of 1/1).

The difference between right-left symmetric waveform control and control in this mode (right-left asymmetric waveform control) is that BFCMn0 to BFCMn2 are transferred to CM0n0 to CM0n2, and that the interrupt signals that start software processing consist just of INTTM0n (generated once per PWM cycle) in the case of right-left symmetric waveform control, and INTTM0n and INTCM0n3 (generated twice per PWM cycle, or once per half cycle) in the case of right-left asymmetric waveform control.

[Output waveform width with respect to set value]

- PWM cycle = BFCMn3 × 2 × T_{TM0n}
- Dead time width T_{Dnm} = (DTRRn + 1)/f_{CLK}
- Active width of positive phase (TO0n0, TO0n2, TO0n4 pins)
  = { (CM0n3 − $CM0nX_{up}$) + (CM0n3 − $CM0nX_{down}$) } × T_{TM0n} − T_{Dnm}
- Active width of negative phase (TO0n1, TO0n3, TO0n5 pins)
  = ($CM0nX_{down}$ + $CM0nX_{up}$) × T_{TM0n} − T_{Dnm}

f_{CLK}: Base clock
T_{TM0n}: TM0n count clock
$CM0nX_{up}$: Set value of CM0n0 to CM0n2 while TM0n is counting up
$CM0nX_{down}$: Set value of CM0n0 to CM0n2 while TM0n is counting down

The pin level when the TO0n0 to TO0n5 pins are reset is high impedance state. When the control mode is selected thereafter, the following levels are output until TM0n is started.

- TO0n0, TO0n2, TO0n4… When active low → High level
  When active high → Low level
- TO0n1, TO0n3, TO0n5… When active low → Low level
  When active high → High level

The active level is set with the ALVTO bit of the TOMRn register. The default is active low.

Caution If a value such that the positive phase or negative phase active width is "0" or a negative value is set in the above formula, the TO0n0 to TO0n5 pins output a waveform fixed to the inactive level waveform with active width "0".

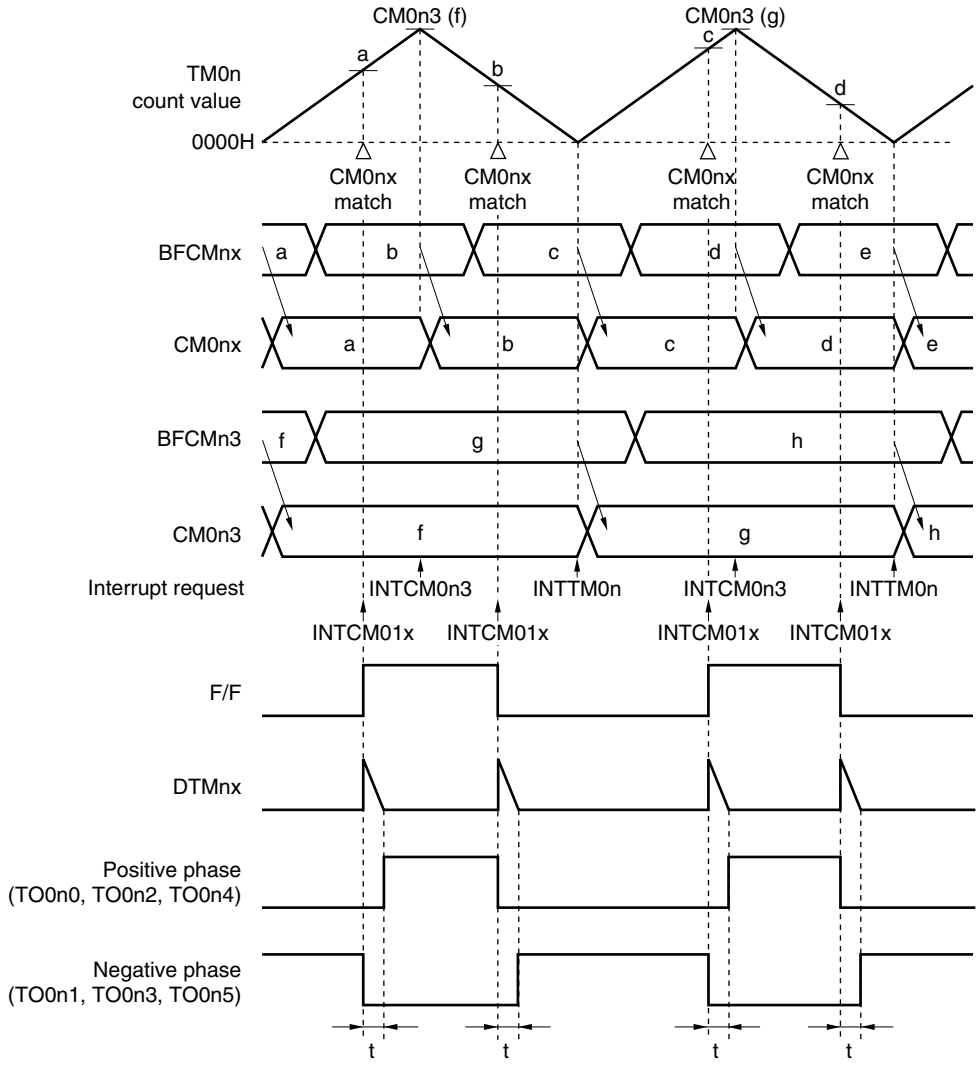
Remarks. 1 m = 0 to 2
n = 0, 1

2. The interrupt request signal occurrence conditions of INTCM010 to INTCM012, INTCM0n4, and INTCM0n5 are shown below.

Setting Condition	INTCM010 to INTCM012, INTCM0n4, INTCM0n5 Signal Occurrence Status
CM010 to CM012, CM0n4, CM0n5 ≤ CM0n3	Occurs
CM010 to CM012, CM0n4, CM0n5 = 0000H	Occurs
CM010 to CM012, CM0n4, CM0n5 > CM0n3	Does not occur

Figure 9-20. Operation Timing in PWM Mode 1 (Asymmetric Triangular Wave) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Remarks 1. The above figure shows the timing chart when both BFTE3 and BFTEN of the TMC0n register are 1, and transfer from BFCMn3 to CM0n3, or from BFCMnx to CM0nx is enabled. Transfer is not performed when BFTE3 = 0 or BFTEN = 0.

2. n = 0, 1

3. x = 0 to 2

4. t: Dead time = (DTRRn + 1)/f_{CLK} (f_{CLK}: Base clock)

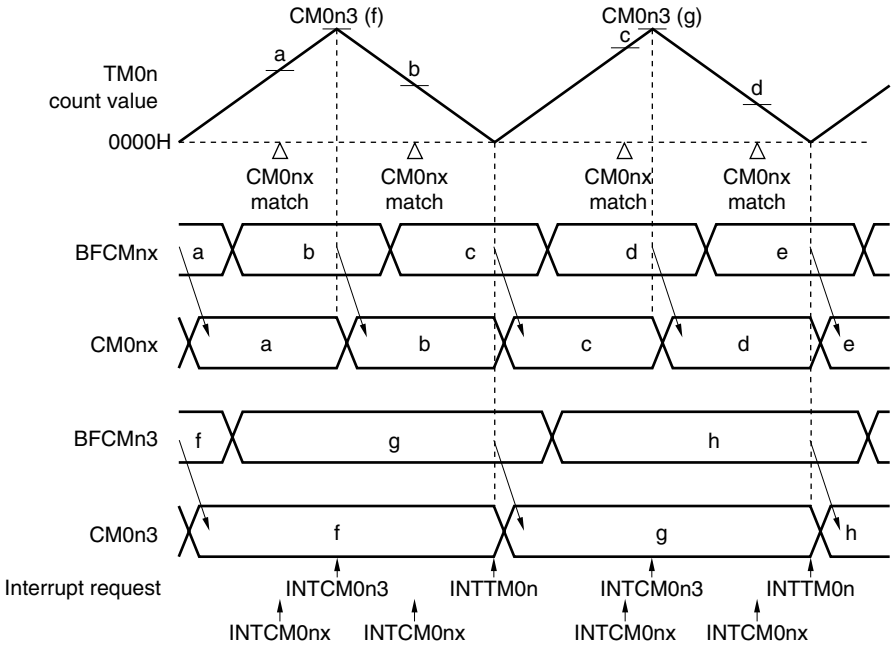
5. To not use dead time, set the TM0CEDn bit of the TMC0n register to 1.

6. The above figure shows an active-high case.

7. INTCM01x is generated on a match between TM01 and CM01x (a to d in the above figure). INTCM00x is not generated.

Figure 9-20. Operation Timing in PWM Mode 1 (Asymmetric Triangular Wave) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5))



Remarks 1. The above figure shows the timing chart when both BFTE3 and BFTEN of the TMC0n register are 1, and transfer from BFCMn3 to CM0n3, or from BFCMnx to CM0nx is enabled. Transfer is not performed when BFTE3 = 0 or BFTEN = 0.

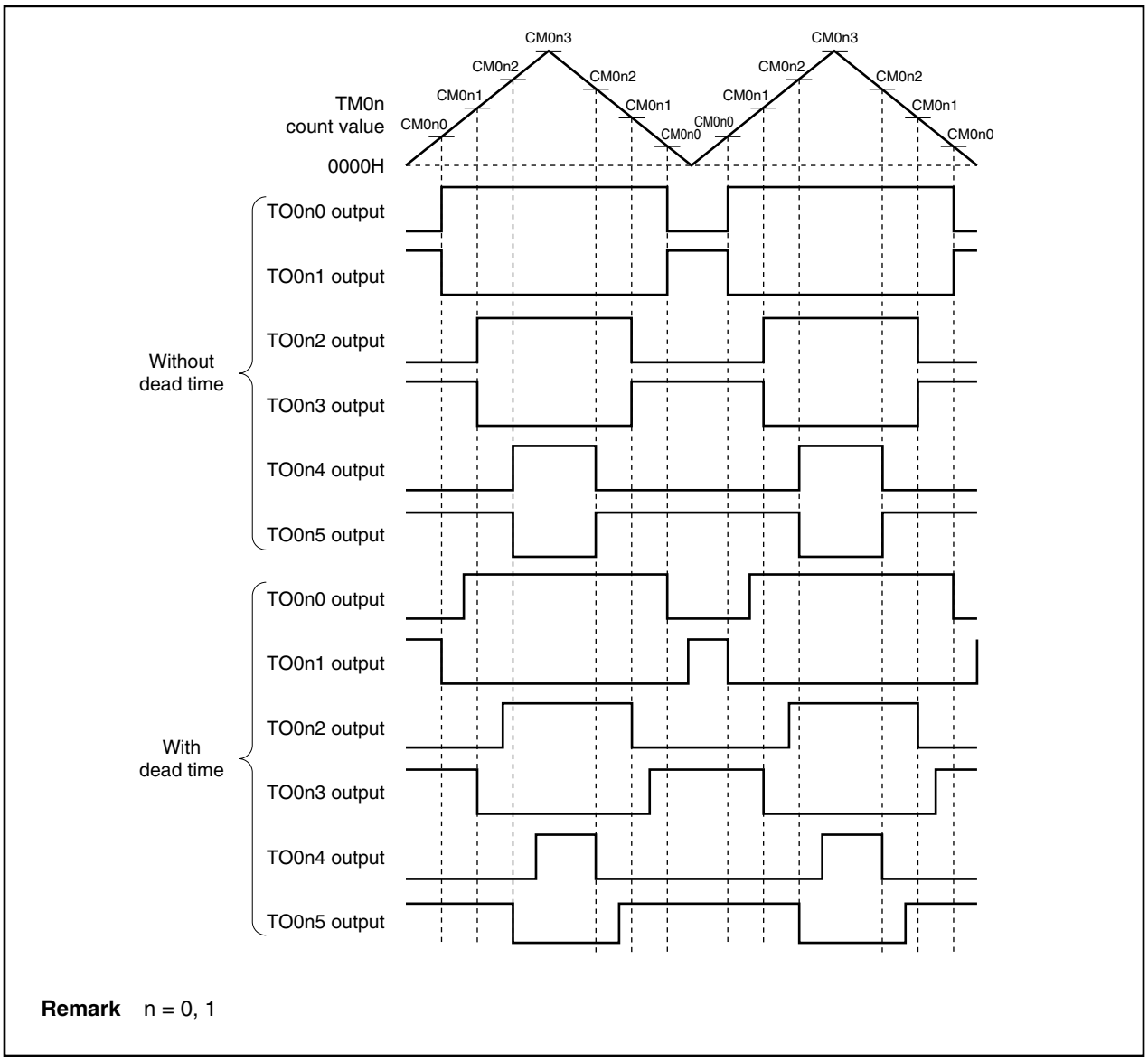
2. n = 0, 1

3. x = 4, 5

4. INTCM0nx is generated on a match between TM0n and CM0nx (a to d in the above figure).

Figure 9-21 shows the overall operation image.

Figure 9-21. Overall Operation Image of PWM Mode 1 (Asymmetric Triangular Wave)

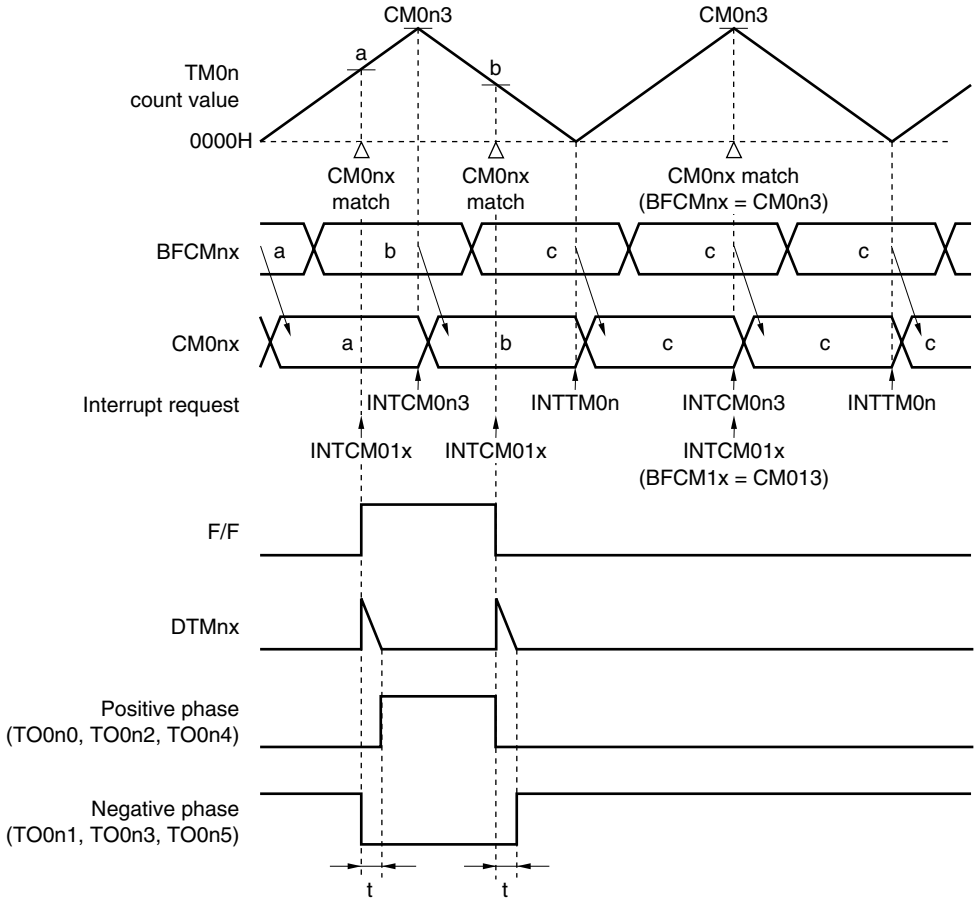


Remark n = 0, 1

(a) When BFCMnx ≥ CM0n3 is set in software processing started by INTCM0n3

Figure 9-22. Operation Timing in PWM Mode 1 (Asymmetric Triangular Wave, BFCMnx ≥ CM0n3) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Remarks 1. n = 0, 1

2. x = 0 to 2

3. c ≥ CM0n3

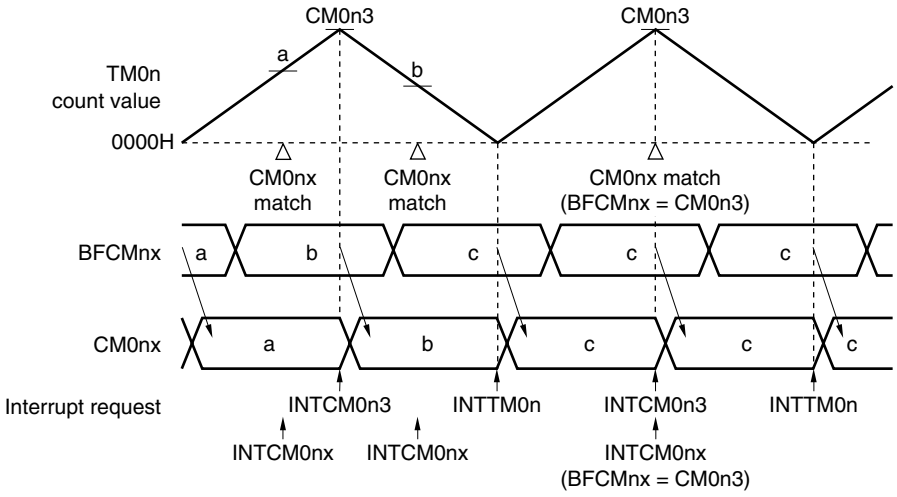
4. t: Dead time = (DTRRn + 1)/f_{CLK} (f_{CLK}: Base clock)

5. The above figure shows an active-high case.

6. INTCM01x is generated on a match between TM01 and CM01x (a and b in the above figure). INTCM00x is not generated.

Figure 9-22. Operation Timing in PWM Mode 1 (Asymmetric Triangular Wave, BFCMnx ≥ CM0n3) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. n = 0, 1

2. x = 4, 5

3. c ≥ CM0n3

4. INTCM0nx is generated on a match between TM0n and CM0nx (a and b in the above figure).

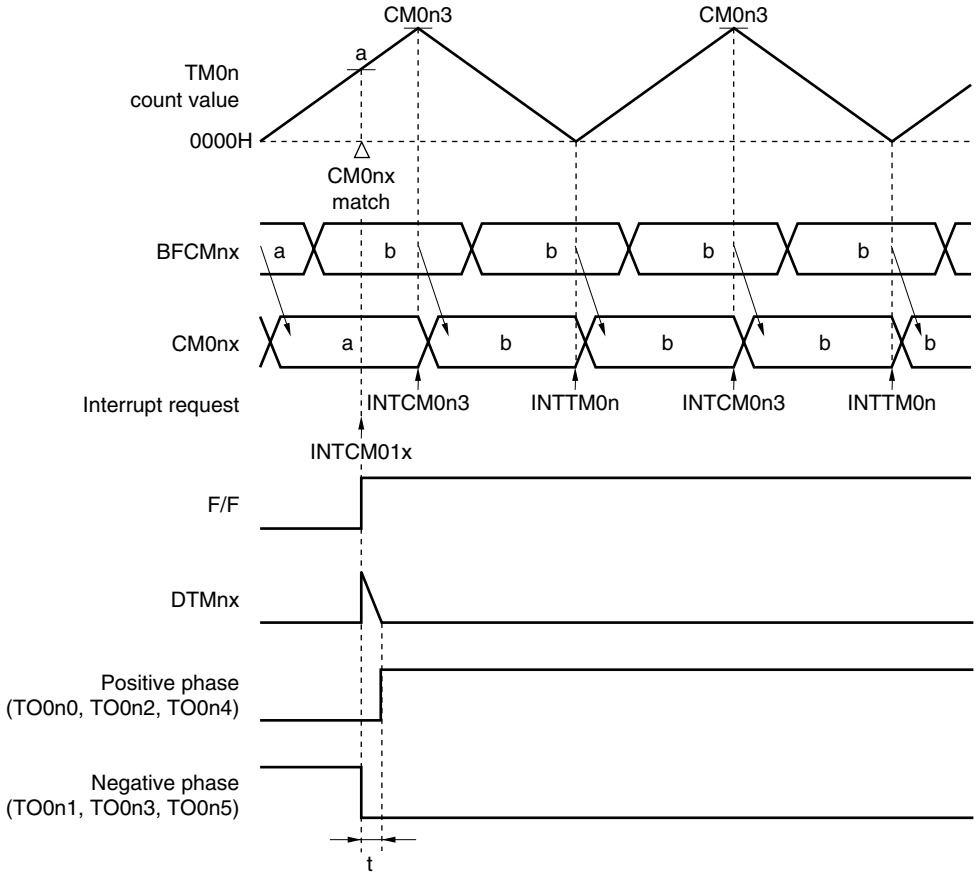
When a value greater than CM0n3 is set to BFCMn0 to BFCMn2, the positive phase side (TO0n0, TO0n2, TO0n4 pins) outputs a low level, and the negative phase side (TO0n1, TO0n3, TO0n5 pins) continues to output a high level. This feature is effective for outputting a low-level or high-level width exceeding the PWM cycle in an application such as inverter control. Furthermore, if CM0n0 to CM0n2 = CM0n3 is set, matching of TM0n and CM0n0 to CM0n2 is detected during down counting by TM0n, so that the F/F remains reset as is, and is not set.

The above explanation applies to an active high case. In an active low case, the levels of positive and negative phases are merely inverted and other operations remain the same.

(b) When BFCMnx > CM0n3 is set in software processing started by INTTM0n

Figure 9-23. Operation Timing in PWM Mode 1 (Asymmetric Triangular Wave, BFCMnx > CM0n3) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Remarks 1. n = 0, 1

2. x = 0 to 2

3. b > CM0n3

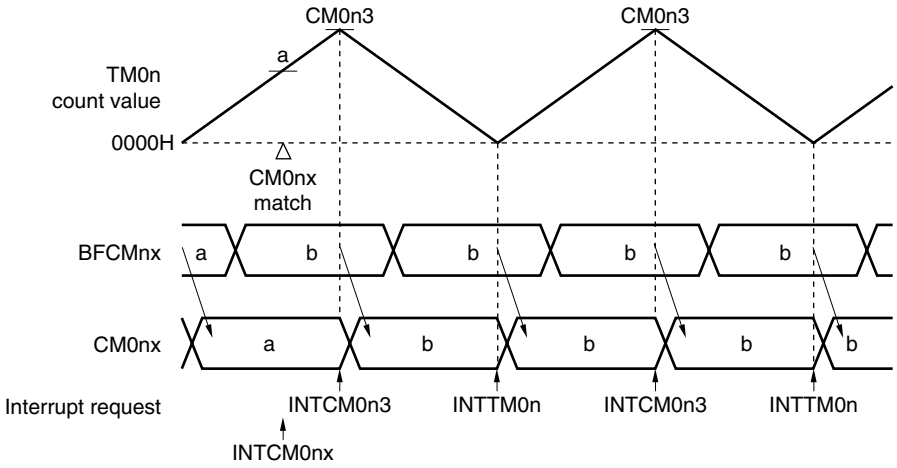
4. t: Dead time = (DTRRn + 1)/f_{CLK} (f_{CLK}: Base clock)

5. The above figure shows an active-high case.

6. INTCM01x is generated on a match between TM01 and CM01x (a in the above figure). INTCM00x is not generated.

Figure 9-23. Operation Timing in PWM Mode 1 (Asymmetric Triangular Wave, BFCMnx > CM0n3) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. n = 0, 1

2. x = 4, 5

3. b > CM0n3

4. INTCM0nx is generated on a match between TM0n and CM0nx (a in the above figure).

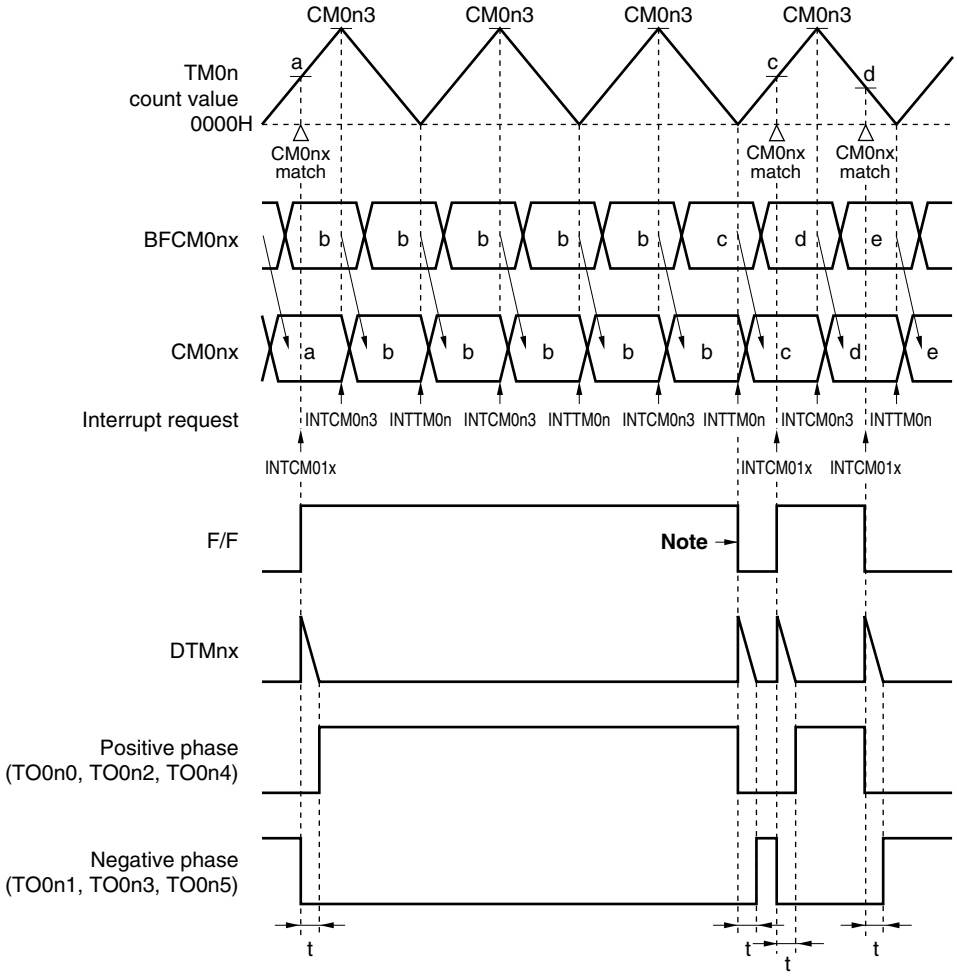
When a value greater than CM0n3 is set to BFCMn0 to BFCMn2, the positive phase side (TO0n0, TO0n2, TO0n4 pins) outputs a high level, and the negative phase side (TO0n1, TO0n3, TO0n5 pins) continues to output a low level. This feature is effective for outputting a low-level or high-level width exceeding the PWM cycle in an application such as inverter control.

The above explanation applies to an active high case. In an active low case, the levels of positive and negative phases are merely inverted and other operations remain the same.

Figure 9-24 shows the change timing from the 100% duty state.

Figure 9-24. Change Timing from 100% Duty State (PWM Mode 1) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Note F/F is reset upon INTTM0n occurrence.

Remarks 1. n = 0, 1

2. x = 0 to 2

3. b > CM0n3

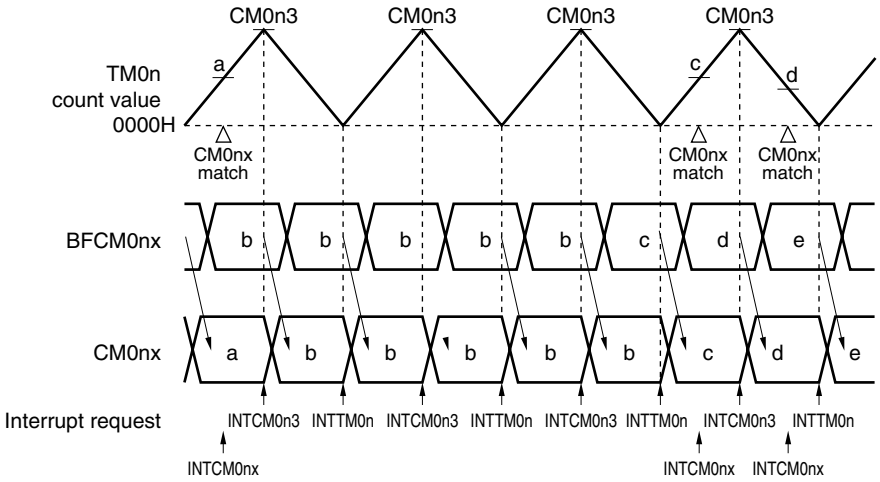
4. t: Dead time = (DTRRn + 1)/f_{CLK} (f_{CLK}: Base clock)

5. The above figure shows an active-high case.

6. INTCM01x is generated on a match between TM01 and CM01x (a to c in the above figure). INTCM00x is not generated.

Figure 9-24. Change Timing from 100% Duty State (PWM Mode 1) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. n = 0, 1

2. x = 4, 5

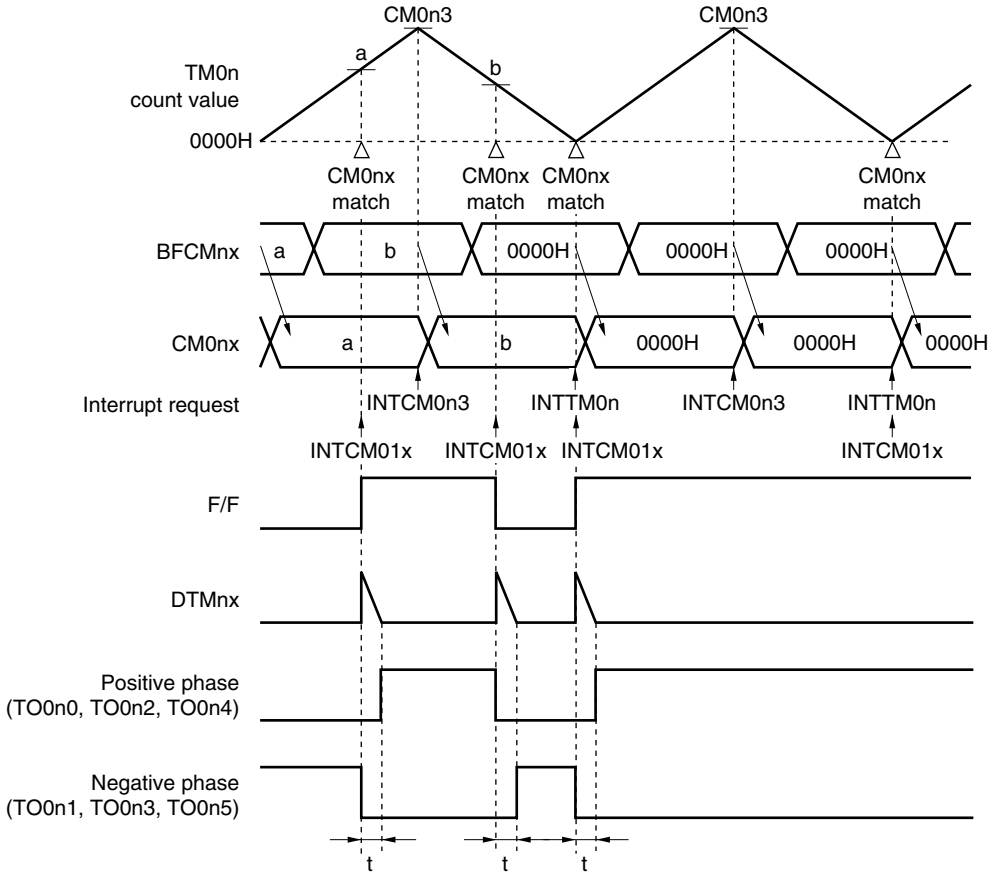
3. b > CM0n3

4. INTCM0nx is generated on a match between TM0n and CM0nx (a to c in the above figure).

(c) When BFCMnx = 0000H is set in software processing started by INTCM0n3

Figure 9-25. Operation Timing in PWM Mode 1 (Asymmetric Triangular Wave, BFCMnx = 0000H) (1) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Remarks 1. n = 0, 1

2. x = 0 to 2

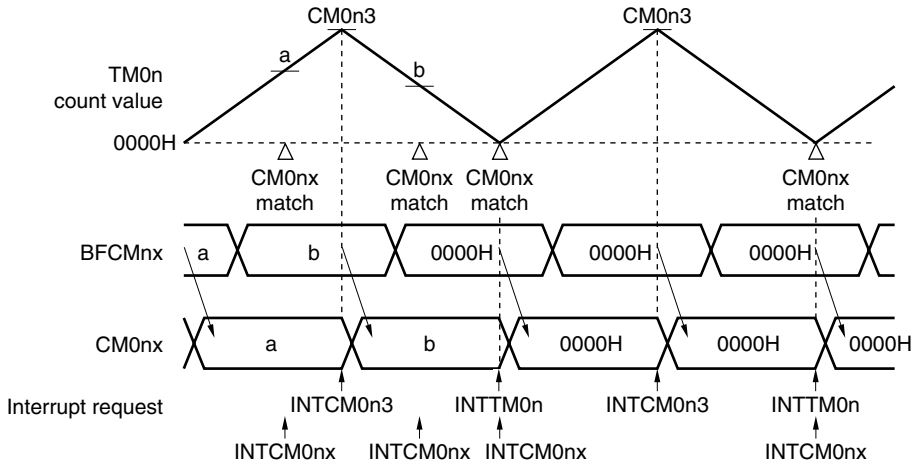
3. t: Dead time = (DTRRn + 1)/f_{CLK} (f_{CLK}: Base clock)

4. The above figure shows an active-high case.

5. INTCM01x is generated on a match between TM01 and CM01x (a and b in the above figure). INTCM00x is not generated.

Figure 9-25. Operation Timing in PWM Mode 1 (Asymmetric Triangular Wave, BFCMnx = 0000H) (1) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. n = 0, 1

2. x = 4, 5

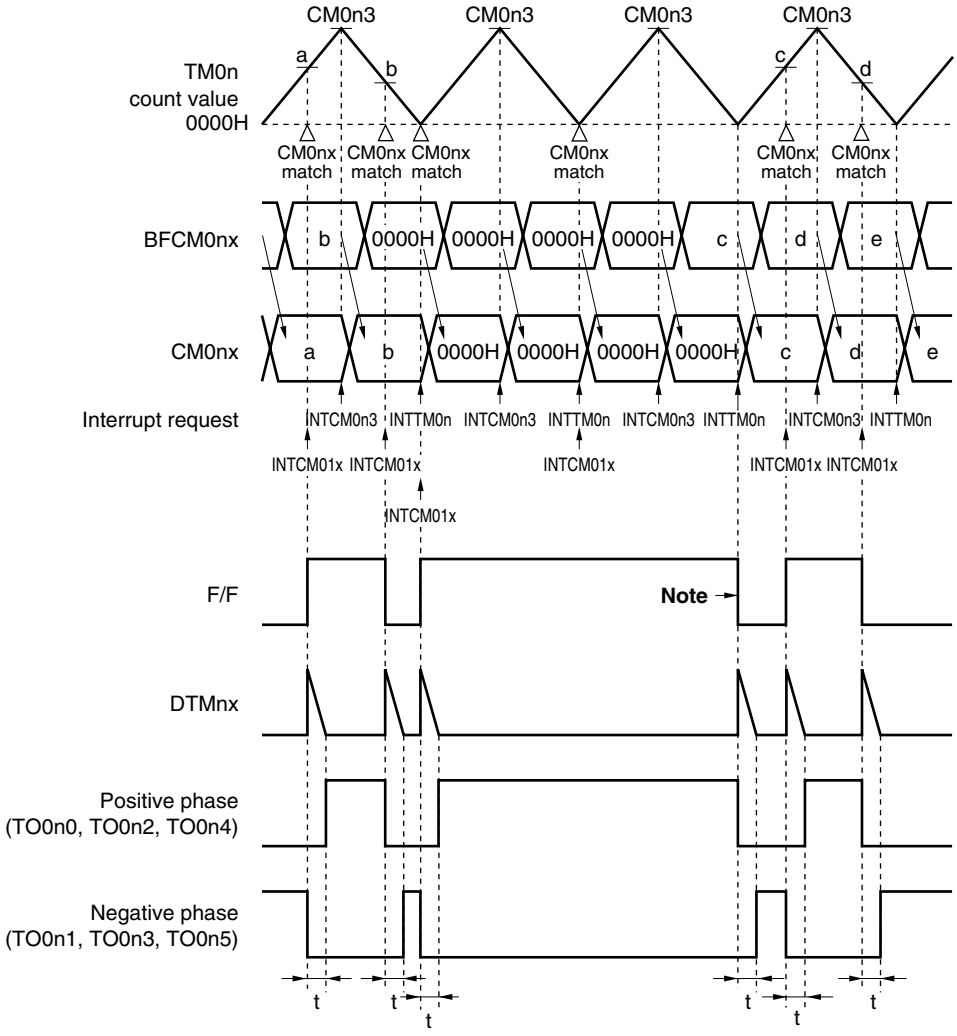
3. INTCM0nx is generated on a match between TM0n and CM0nx (a and b in the above figure).

Since a TM0n = CM0n0 to CM0n2 = 0000H match is detected during up counting by TM0n, the F/F is just set and is not reset. The F/F is also set upon match detection in the cycle when 0000H is transferred to CM0n0 to CM0n2 by INTTM0n interrupt.

Figure 9-26 shows the change timing from the 100% duty state.

Figure 9-26. Change Timing from 100% Duty State (1) (PWM Mode 1) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Note The F/F is reset upon INTTM0n occurrence.

Remarks 1. n = 0, 1

2. x = 0 to 2

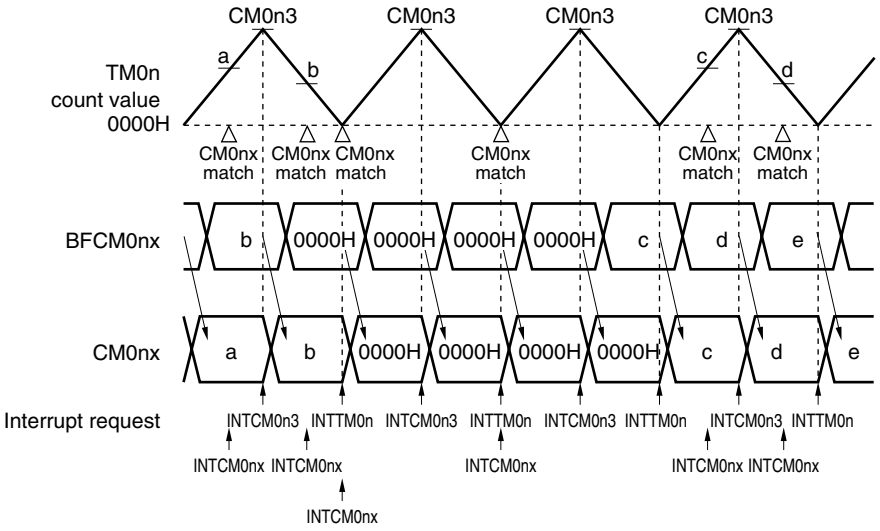
3. t: Dead time = (DTRRn + 1)/f_{CLK} (f_{CLK}: Base clock)

4. The above figure shows an active-high case.

5. INTCM01x is generated on a match between TM01 and CM01x (a to d in the above figure). INTCM00x is not generated.

Figure 9-26. Change Timing from 100% Duty State (1) (PWM Mode 1) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. n = 0, 1

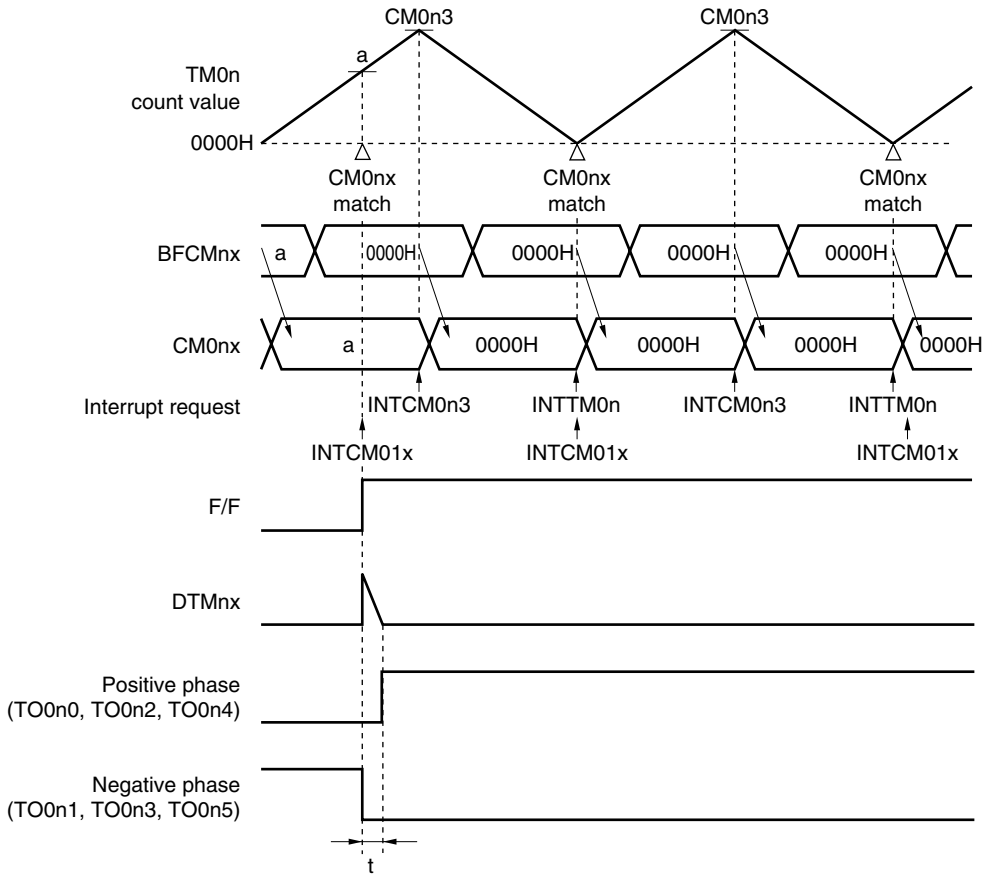
2. x = 4, 5

3. INTCM0nx is generated on a match between TM0n and CM0nx (a to d in the above figure).

(d) When BFCMnx = 0000H is set in software processing started by INTTM0n

Figure 9-27. Operation Timing in PWM Mode 1 (Asymmetric Triangular Wave, BFCMnx = 0000H) (2) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Remarks 1. n = 0, 1

2. x = 0 to 2

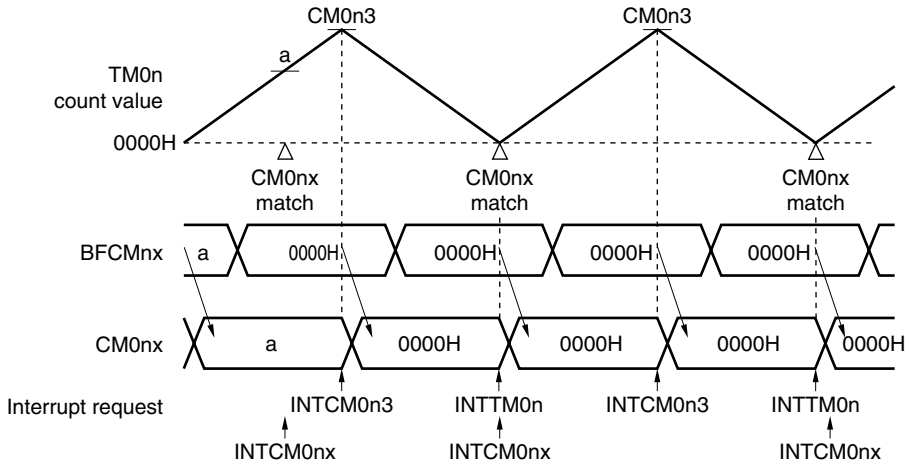
3. t: Dead time = (DTRRn + 1)/f_{CLK} (f_{CLK}: Base clock)

4. The above figure shows an active-high case.

5. INTCM01x is generated on a match between TM01 and CM01x (a in the above figure). INTCM00x is not generated.

Figure 9-27. Operation Timing in PWM Mode 1 (Asymmetric Triangular Wave, BFCMnx = 0000H) (2) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. n = 0, 1

2. x = 4, 5

3. INTCM0nx is generated on a match between TM0n and CM0nx (a in the above figure).

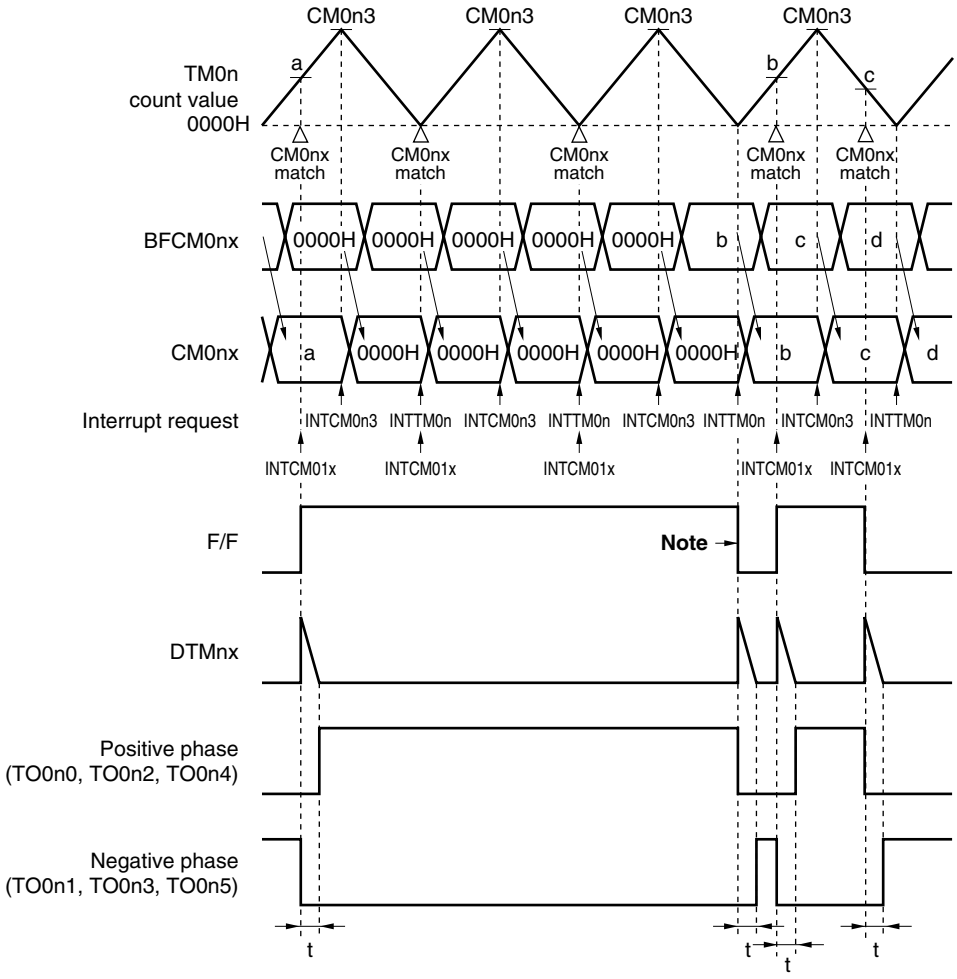
Since TM0n = CM0n0 to CM0n2 = 0000H match is detected during up counting by TM0n, the F/F is just set and is not reset. Therefore, the positive phase side (TO0n0, TO0n2, TO0n4 pins) outputs a high level, and the negative phase side (TO0n1, TO0n3, TO0n5 pins) continues to output a low level.

The above explanation applies to an active high case. In an active low case, the levels of positive and negative phases are merely inverted and other operations remain the same.

Figure 9-28 shows the change timing from the 100% duty state.

Figure 9-28. Change Timing from 100% Duty State (2) (PWM Mode 1) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Note F/F is reset upon INTTM0n occurrence.

Remarks 1. n = 0, 1

2. x = 0 to 2

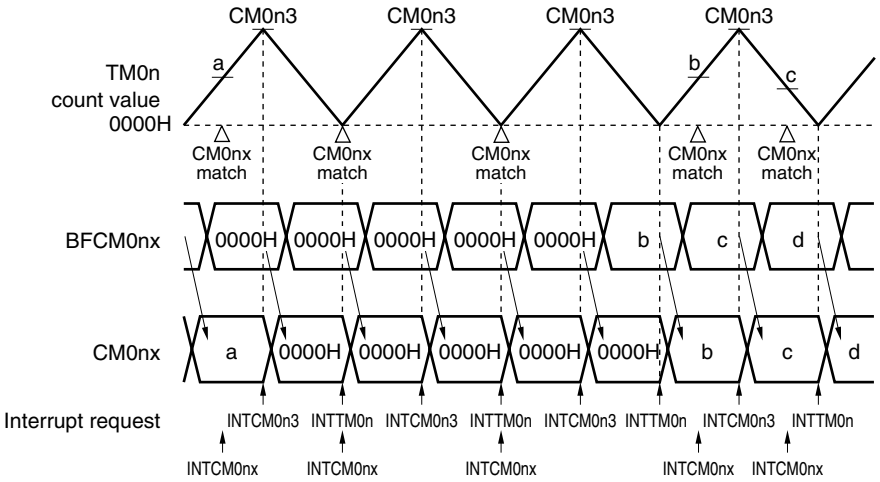
3. t: Dead time = (DTRRn + 1)/f_{CLK} (f_{CLK}: Base clock)

4. The above figure shows an active-high case.

5. INTCM01x is generated on a match between TM01 and CM01x (a to c in the above figure). INTCM00x is not generated.

Figure 9-28. Change Timing from 100% Duty State (2) (PWM Mode 1) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. n = 0, 1

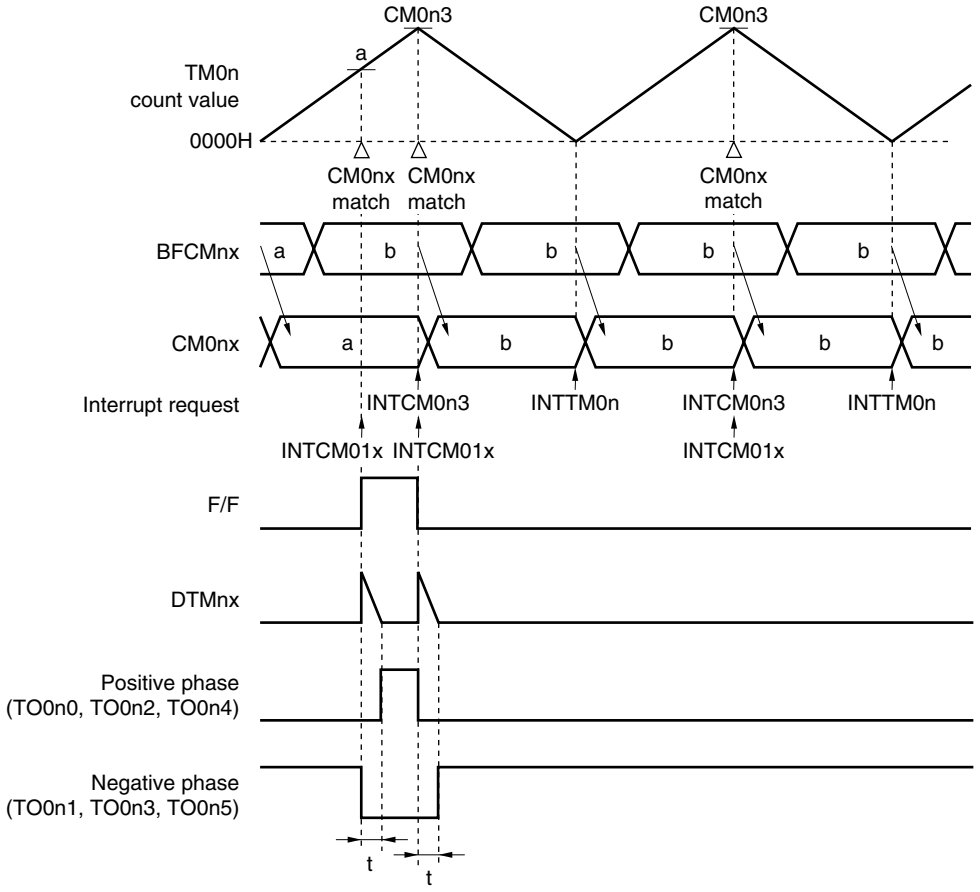
2. x = 4, 5

3. INTCM0nx is generated on a match between TM0n and CM0nx (a to c in the above figure).

(e) When BFCMnx = CM0n3 is set in software processing started by INTTM0n

Figure 9-29. Operation Timing in PWM Mode 1 (Asymmetric Triangular Wave, BFCMnx = CM0n3) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Remarks 1. n = 0, 1

2. x = 0 to 2

3. b = CM0n3

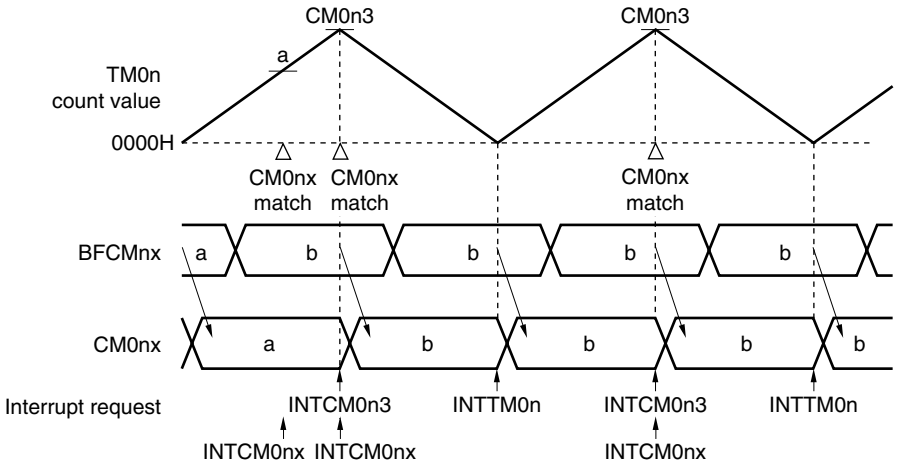
4. t: Dead time = (DTRRn + 1)/f_{CLK} (f_{CLK}: Base clock)

5. The above figure shows an active-high case.

6. INTCM01x is generated on a match between TM01 and CM01x (a in the above figure). INTCM00x is not generated.

Figure 9-29. Operation Timing in PWM Mode 1 (Asymmetric Triangular Wave, BFCMnx = CM0n3) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. n = 0, 1

2. x = 4, 5

3. b = CM0n3

4. INTCM0nx is generated on a match between TM0n and CM0nx (a in the above figure).

Since TM0n and CM0n0 to CM0n2 match is detected during count down of TM0n when BFCMn0 to BFCMn2 = CM0n3 has been set, the F/F remains reset as is and is not set. Therefore, the positive phase side (TO0n0, TO0n2, TO0n4 pins) outputs a low level, and the negative phase side (TO0n1, TO0n3, TO0n5 pins) continues to output a high level. Moreover, the timing of matching with TM0n with CM0n0 to CM0n2 = CM0n3 is the cycle when transfer is performed from BFCMn0 to BFCMn2 to CM0n0 to CM0n2 by INTCM0n3.

The above explanation applies to an active high case. In an active low case, the levels of positive and negative phases are merely inverted and other operations remain the same.

(4) PWM mode 2: Sawtooth wave modulation

[Setting procedure]

(a) Set PWM mode 2 (sawtooth wave) using the MOD01 and MOD00 bits of the TMC0n register. Also set the active level of the TO0n0 to TO0n5 pins using the ALVTO bit of the TOMRn register.

(b) Set the count clock of TM0n using the PRM02 to PRM00 bits of the TMC0n register. The transfer operation from BFCMn3 to CM0n3 is set using the BFTE3 bit, and the transfer operation from BFCMn0 to BFCMn2, BFCMn4, and BFCMn5 to CM0n0 to CM0n2, CM0n4, and CM0n5 is set using the BFTEN bit.

(c) Set the initial values.

(i) Specify the interrupt culling ratio using the CUL02 to CUL00 bits of the TMC0n register.

(ii) Set the cycle width of the PWM cycle in BFCMn3.
- PWM cycle = (BFCMn3 value + 1) × TM0n count clock
  (The TM0n count clock is set by the TMC0n register.)

(iii) Set the dead-time width in DTRRn.
- Dead-time width = (DTRRn + 1)/f_{CLK}
  f_{CLK}: Base clock

(iv) Set the set/reset timing of the F/F used in the PWM cycle in BFCM0n0 to BFCM0n2.

(d) Clear (0) the TM0CEDn bit of the TMC0n register to enable dead-time timer operation. Set TM0CEDn = 1 when not using dead time.

(e) Setting (1) the TM0CEn bit of the TMC0n register starts TM0n counting, and a 6-channel PWM signal is output from pins TO0n0 to TO0n5.

Caution Setting CM0n3 to 0000H is prohibited.

[Operation]

In PWM mode 2, TM0n performs up count operation, and when it matches the value of CM0n3, match interrupt INTCM0n3 is generated and TM0n is cleared (n = 0, 1).

The PWM cycle in this mode is ((BFCMn3 value + 1) × TM0n count clock). Note that the next PWM cycle width is set to BFCMn3.

The data of BFCMn3 is automatically transferred by hardware to CM0n3 upon generation of the INTCM0n3 interrupt. Furthermore, calculation is performed by software processing started by INTCM0n3, and the data for the next cycle is set to BFCMn3.

Data setting to CM0n0 to CM0n2, which control the PWM duty, is explained next.

Setting of data to CM0n0 to CM0n2 consists of setting the duty output from BFCMn0 to BFCMn2.

The values of BFCMn0 to BFCMn2 are automatically transferred by hardware to CM0n0 to CM0n2 upon generation of the INTCM0n3 interrupt. Furthermore, software processing is started up and calculation performed, and reset timing of the F/F for the next cycle is set to BFCMn0 to BFCMn2.

The PWM cycle and the PWM duty are set in the above procedure.

The F/F set/reset conditions upon match of CM0n0 to CM0n2 are as follows.

- Set: TM0n and CM0n3 match detection and rising edge of TM0CEn bit of TMC0n register
- Reset: TM0n and CM0n0 to CM0n2 match detection

The values of DTRRn are transferred to the corresponding dead-time timers (DTMn0 to DTMn2) in synchronization with the set/reset timing of the F/F, and down counting is started. DTMn0 to DTMn2 count down to 000H, and stop when they count down further to FFFH.

DTMn0 to DTMn2 can automatically generate a width at which the active levels of the positive phase (TO0n0, TO0n2, TO0n4) and negative phase (TO0n1, TO0n3, TO0n5) do not overlap (dead time).

In this way, software processing is started by an interrupt (INTCM0n3) that occurs once during every PWM cycle after initial setting has been performed, and by setting the PWM cycle and PWM duty to be used in the next cycle, it is possible to automatically output a PWM waveform to pins TO0n0 to TO0n5 taking into consideration the dead-time width (in the case of an interrupt culling ratio of 1/1).

[Output waveform width with respect to set value]

- PWM cycle = (BFCMn3 + 1) × T_{TM0n}
- Dead time width T_{Dnm} = (DTRRn + 1)/f_{CLK}
- Active width of positive phase (TO0n0, TO0n2, TO0n4 pins)
  = (CM0nX + 1) × T_{TM0n} – T_{Dnm}
- Active width of negative phase (TO0n1, TO0n3, TO0n5 pins)
  = (CM0n3 – CM0nX) × T_{TM0n} – T_{Dnm}

f_{CLK}: Base clock
T_{TM0n}: TM0n count clock
CM0nX: Set value of CM0n0 to CM0n2

The pin level when the TO0n0 to TO0n5 pins are reset is the high impedance state. When the control mode is selected thereafter, the following levels are output until the TM0n is started.

- TO0n0, TO0n2, TO0n4… When active low → High level
  When active high → Low level
- TO0n1, TO0n3, TO0n5… When active low → Low level
  When active high → High level

The active level is set with the ALVTO bit of the TOMRn register. The default is active low.

Caution If a value such that the positive phase or negative phase active width is "0" or a negative value is set in the above formula, the TO0n0 to TO0n5 pins output a waveform fixed to the inactive level waveform with active width "0".

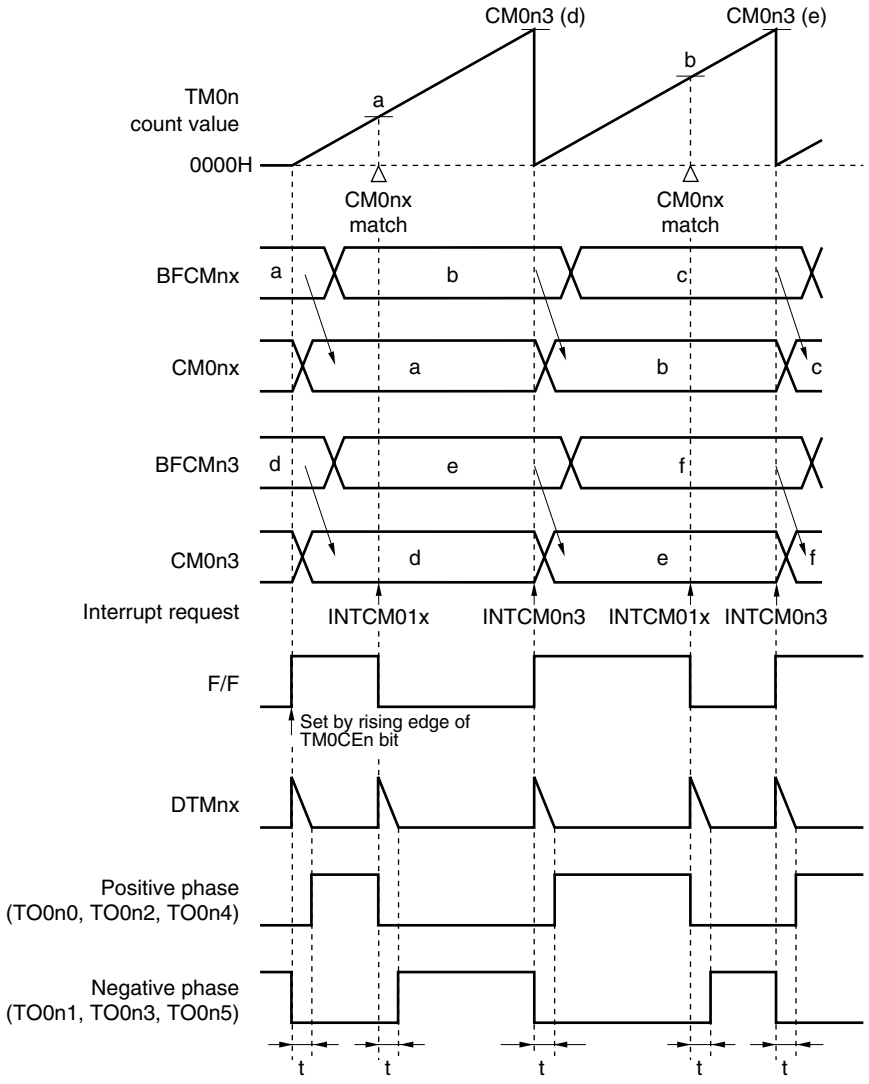
Remarks. 1 m = 0 to 2
n = 0, 1

2. The interrupt request signal occurrence conditions of INTCM010 to INTCM012, INTCM0n4, and INTCM0n5 are shown below.

Setting Condition	INTCM010 to INTCM012, INTCM0n4, INTCM0n5 Signal Occurrence Status
CM010 to CM012, CM0n4, CM0n5 ≤ CM0n3	Occurs
CM010 to CM012, CM0n4, CM0n5 = 0000H	Occurs
CM010 to CM012, CM0n4, CM0n5 > CM0n3	Does not occur

Figure 9-30. Operation Timing in PWM Mode 2 (Sawtooth Wave) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Remarks 1. The above figure shows the timing chart when both BFTE3 and BFTEN of the TMC0n register are 1, and transfer from BFCMn3 to CM0n3, or from BFCMnx to CM0nx is enabled. Transfer is not performed when BFTE3 = 0 or BFTEN = 0.

2. n = 0, 1

3. x = 0 to 2

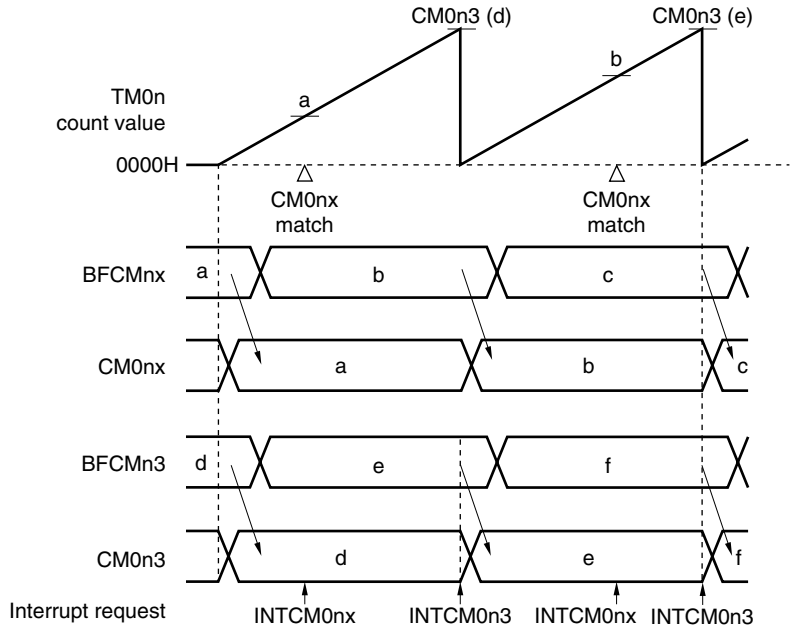
4. t: Dead time = (DTRRn + 1)/f_{CLK} (f_{CLK}: Base clock)

5. The above figure shows an active-high case.

6. INTCM01x is generated on a match between TM01 and CM01x (a and b in the above figure). INTCM00x is not generated.

Figure 9-30. Operation Timing in PWM Mode 2 (Sawtooth Wave) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. The above figure shows the timing chart when both BFTE3 and BFTEN of the TMC0n register are 1, and transfer from BFCMn3 to CM0n3, or from BFCMnx to CM0nx is enabled. Transfer is not performed when BFTE3 = 0 or BFTEN = 0.

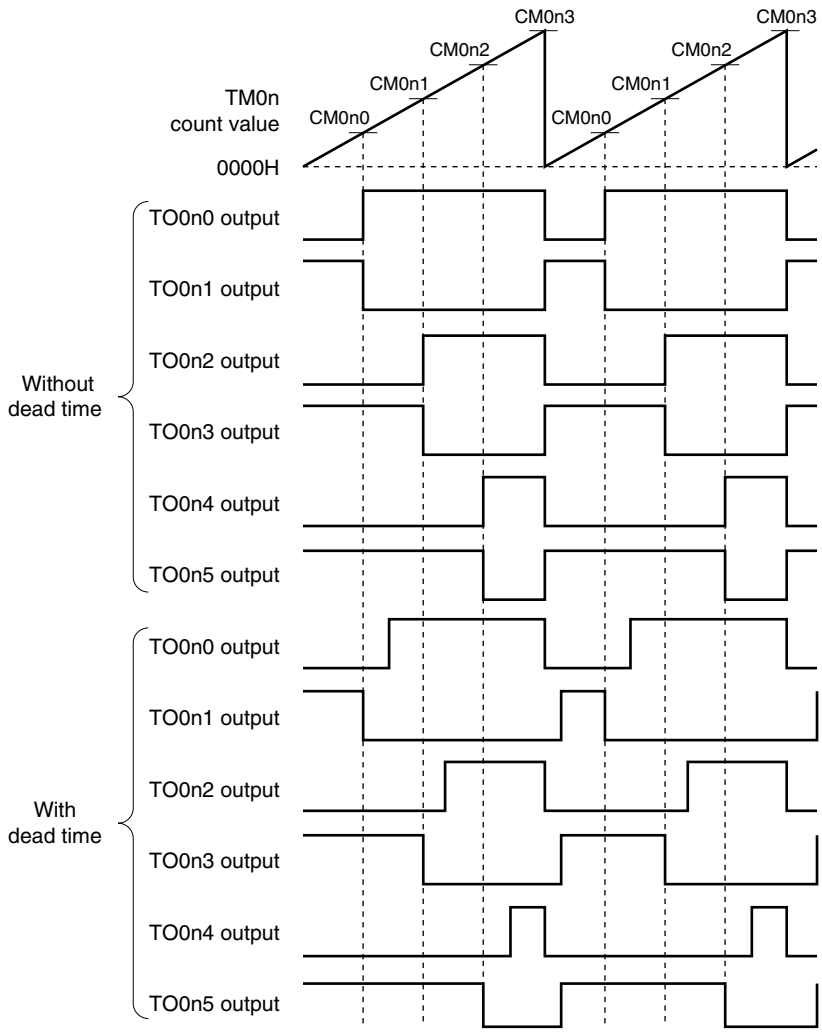
2. n = 0, 1

3. x = 4, 5

4. INTCM0nx is generated on a match between TM0n and CM0nx (a and b in the above figure).

Figure 9-31 shows the overall operation image.

Figure 9-31. Overall Operation Image of PWM Mode 2 (Sawtooth Wave)



Remarks. 1. n = 0, 1

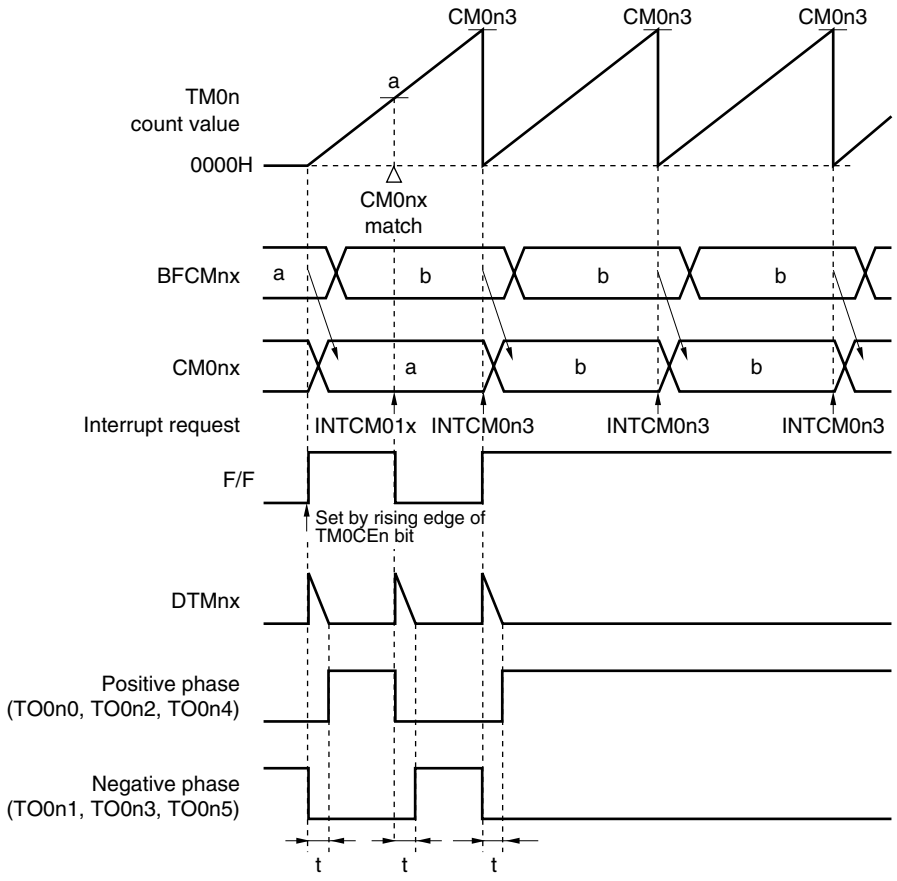
2. The above figure shows an active low case.

Since the F/F is set at the rising edge of the TM0CEn bit of the TMC0n register in the first cycle, the PWM signal can be output.

(a) When BFCMnx > CM0n3 is set

Figure 9-32. Operation Timing in PWM Mode 2 (Sawtooth Wave, BFCMnx > CM0n3) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Remarks 1. n = 0, 1

2. x = 0 to 2

3. b > CM0n3

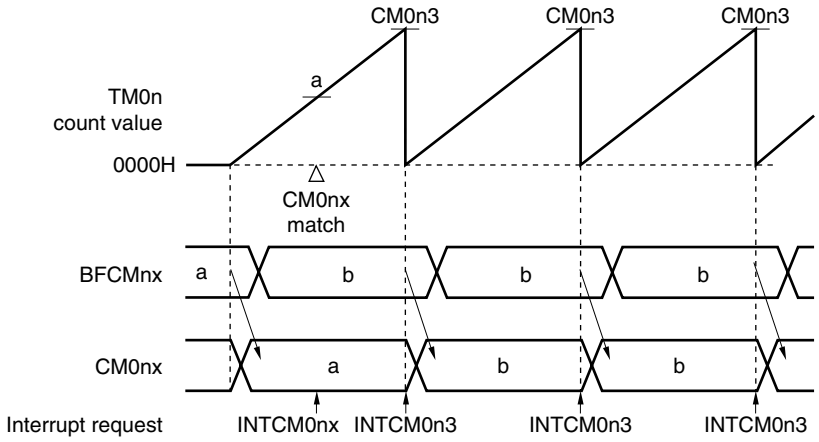
4. t: Dead time = (DTRRn + 1)/f_{CLK} (f_{CLK}: Base clock)

5. The above figure shows an active-high case.

6. INTCM01x is generated on a match between TM01 and CM01x (a in the above figure). INTCM00x is not generated.

Figure 9-32. Operation Timing in PWM Mode 2 (Sawtooth Wave, BFCMnx > CM0n3) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. n = 0, 1

2. x = 4, 5

3. b > CM0n3

4. INTCM0nx is generated on a match between TM0n and CM0nx (a in the above figure).

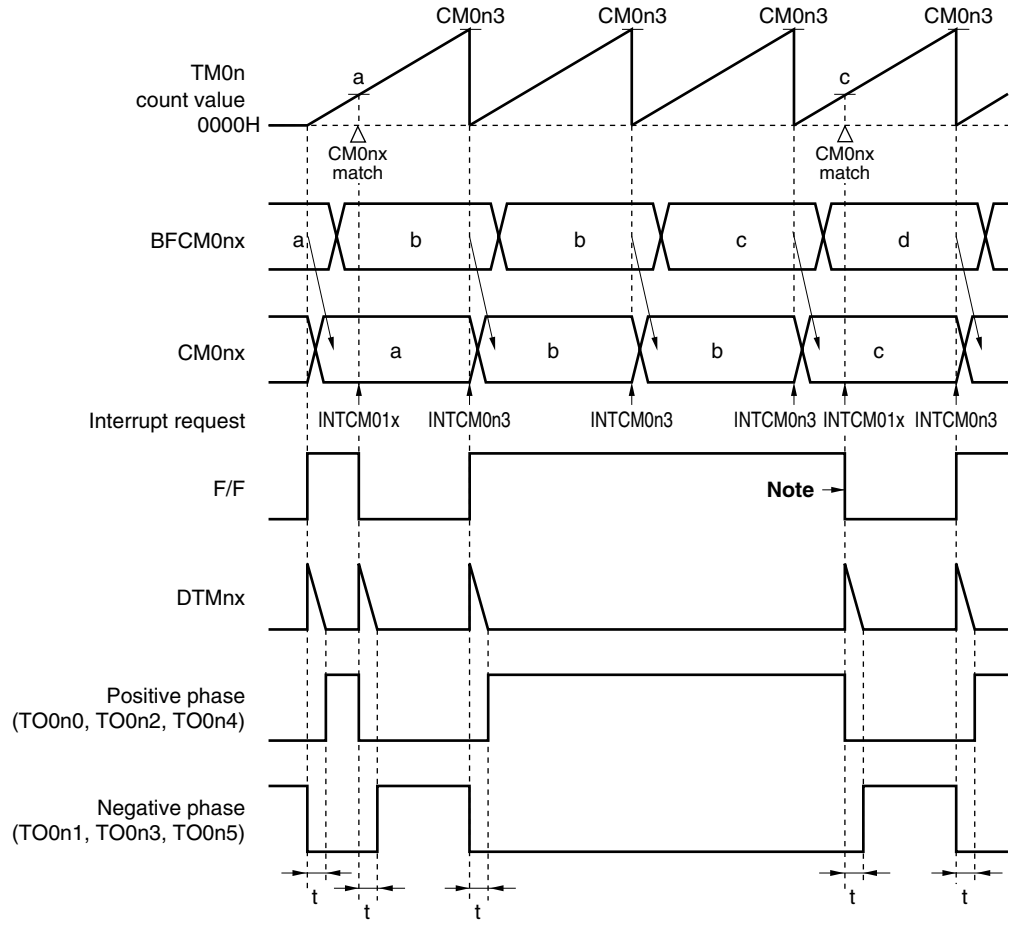
When a value greater than CM0n3 is set to BFCMn0 to BFCMn2, the positive phase side (TO0n0, TO0n2, TO0n4 pins) outputs a high level, and the negative phase side (TO0n1, TO0n3, TO0n5 pins) continues to output a low level. Since TM0n and CM0n0 to CM0n2 match does not occur, the F/F is not reset. This feature is effective for outputting a low-level or high-level width exceeding the PWM cycle in an application such as inverter control.

The above explanation applies to an active high case. In an active low case, the levels of positive and negative phases are merely inverted and other operations remain the same.

Figure 9-33 shows the change timing from the 100% duty state.

Figure 9-33. Change Timing from 100% Duty State (PWM Mode 2) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Note The F/F is reset upon a match with CM0nx.

Remarks 1. n = 0, 1

2. x = 0 to 2

3. b > CM0n3

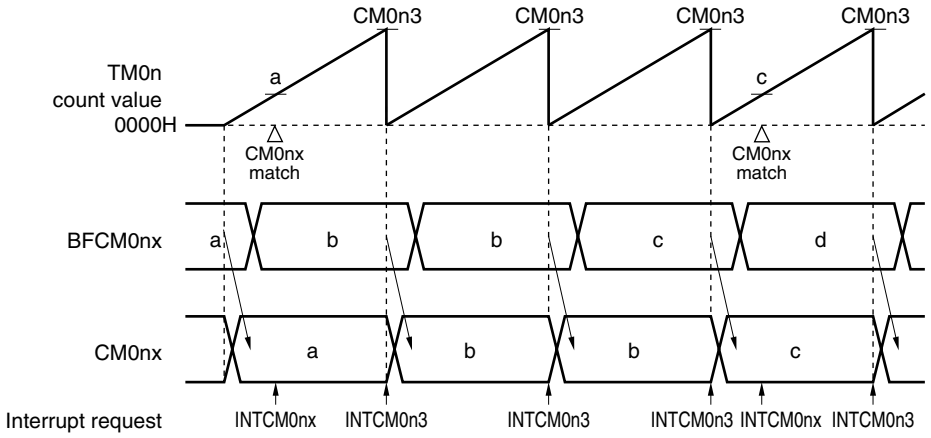
4. t: Dead time = (DTRRn + 1)/f_{CLK} (f_{CLK}: Base clock)

5. The above figure shows an active-high case.

6. INTCM01x is generated on a match between TM01 and CM01x (a and c in the above figure). INTCM00x is not generated.

Figure 9-33. Change Timing from 100% Duty State (PWM Mode 2) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. n = 0, 1

2. x = 4, 5

3. b > CM0n3

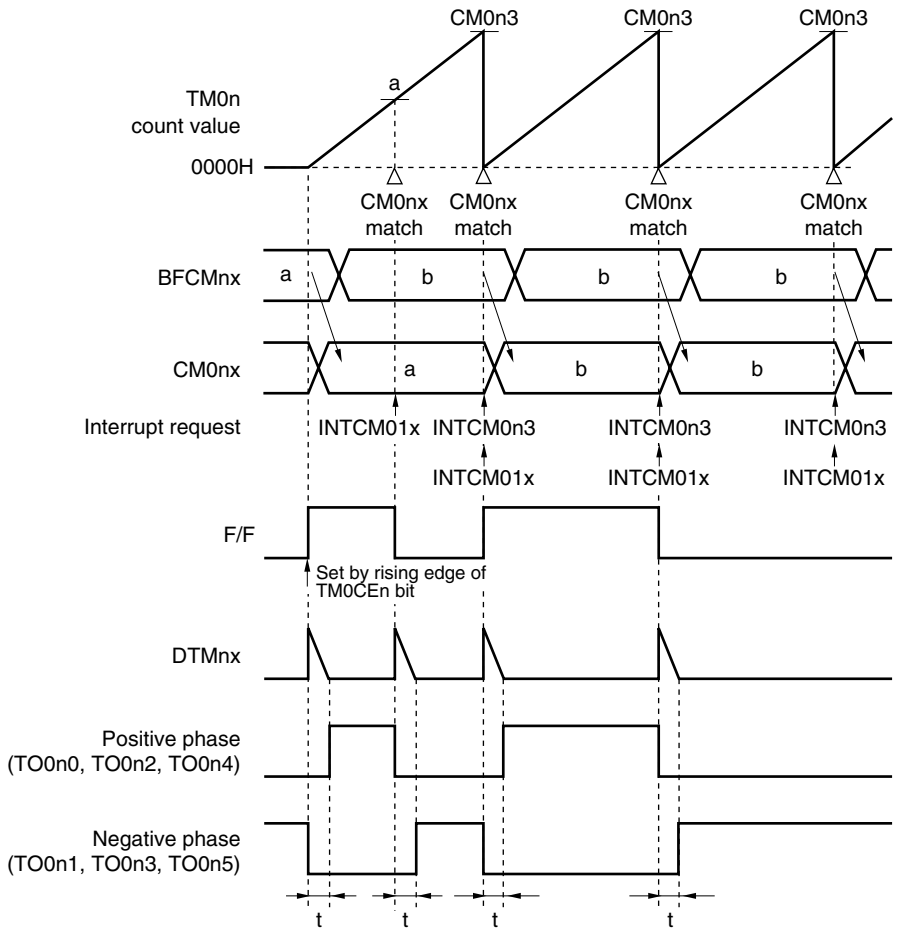
4. INTCM0nx is generated on a match between TM0n and CM0nx (a and c in the above figure).

The timing at which the F/F is reset is upon occurrence of a match with CM0n0 to CM0n2 as usual.

(b) When BFCMnx = CM0n3 is set

Figure 9-34. Operation Timing in PWM Mode 2 (Sawtooth Wave, BFCMnx = CM0n3) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Remarks 1. n = 0, 1

2. x = 0 to 2

3. b = CM0n3

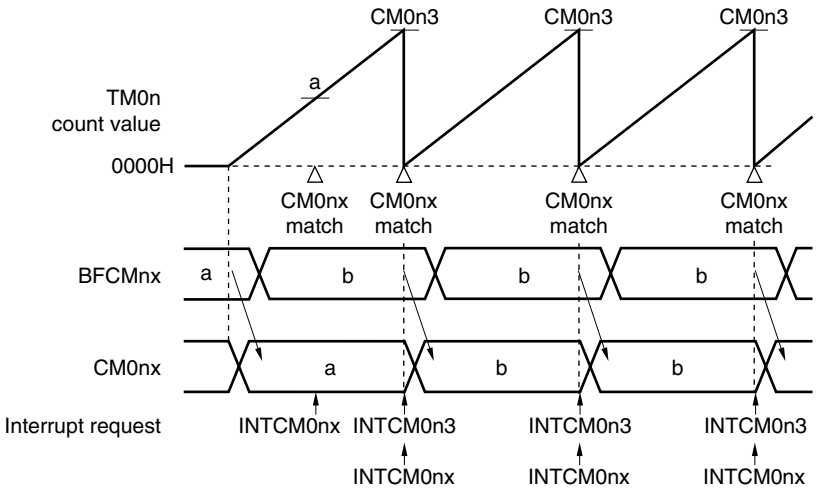
4. t: Dead time = (DTRRn + 1)/f_{CLK} (f_{CLK}: Base clock)

5. The above figure shows an active-high case.

6. INTCM01x is generated on a match between TM01 and CM01x (a in the above figure). INTCM00x is not generated.

Figure 9-34. Operation Timing in PWM Mode 2 (Sawtooth Wave, BFCMnx = CM0n3) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. n = 0, 1

2. x = 4, 5

3. b = CM0n3

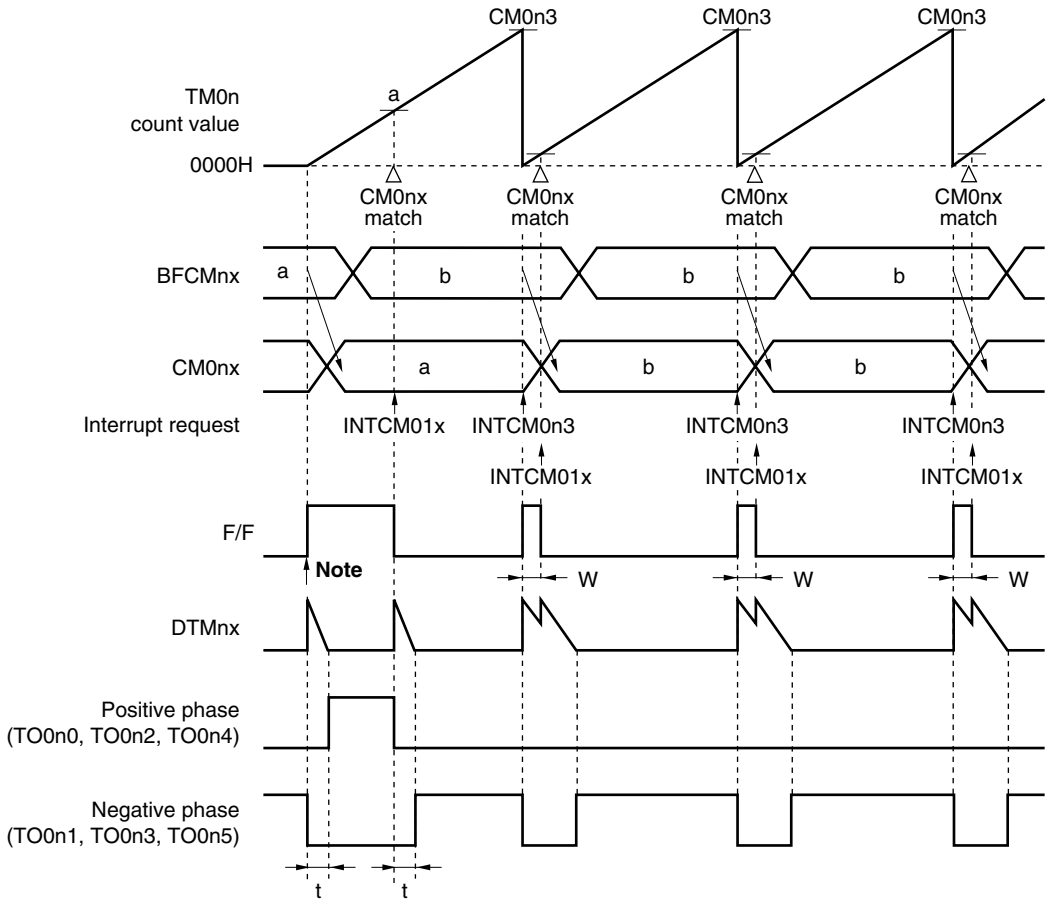
4. INTCM0nx is generated on a match between TM0n and CM0nx (a in the above figure).

If match signal INTCM0n3 for TM0n and CM0n3 and the match signal for TM0n and CM0n0 to CM0n2 conflict, reset of the F/F takes precedence, so that the F/F is not set following a match of CM0n0 to CM0n2 (= CM0n3) and TM0n.

(c) When BFCMnx = 0000H is set

Figure 9-35. Operation Timing in PWM Mode 2 (Sawtooth Wave, BFCMnx = 0000H) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Note Set at the rising edge of the TM0CEn bit.

Remarks 1. n = 0, 1

2. x = 0 to 2

3. t: Dead time = (DTRRn + 1)/f_{CLK} (f_{CLK}: Base clock)

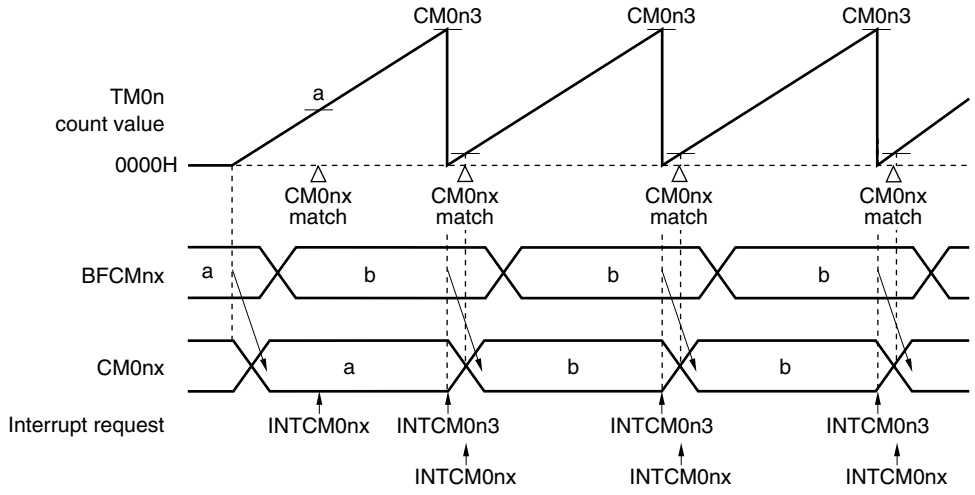
4. The above figure shows an active-high case.

5. W: Width between CM0n3 match and CM0nx match (timer count clock)

6. INTCM01x is generated on a match between TM01 and CM01x (a in the above figure). INTCM00x is not generated.

Figure 9-35. Operation Timing in PWM Mode 2 (Sawtooth Wave, BFCMnx = 0000H) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. n = 0, 1

2. x = 4, 5

3. INTCM0nx is generated on a match between TM0n and CM0nx (a in the above figure).

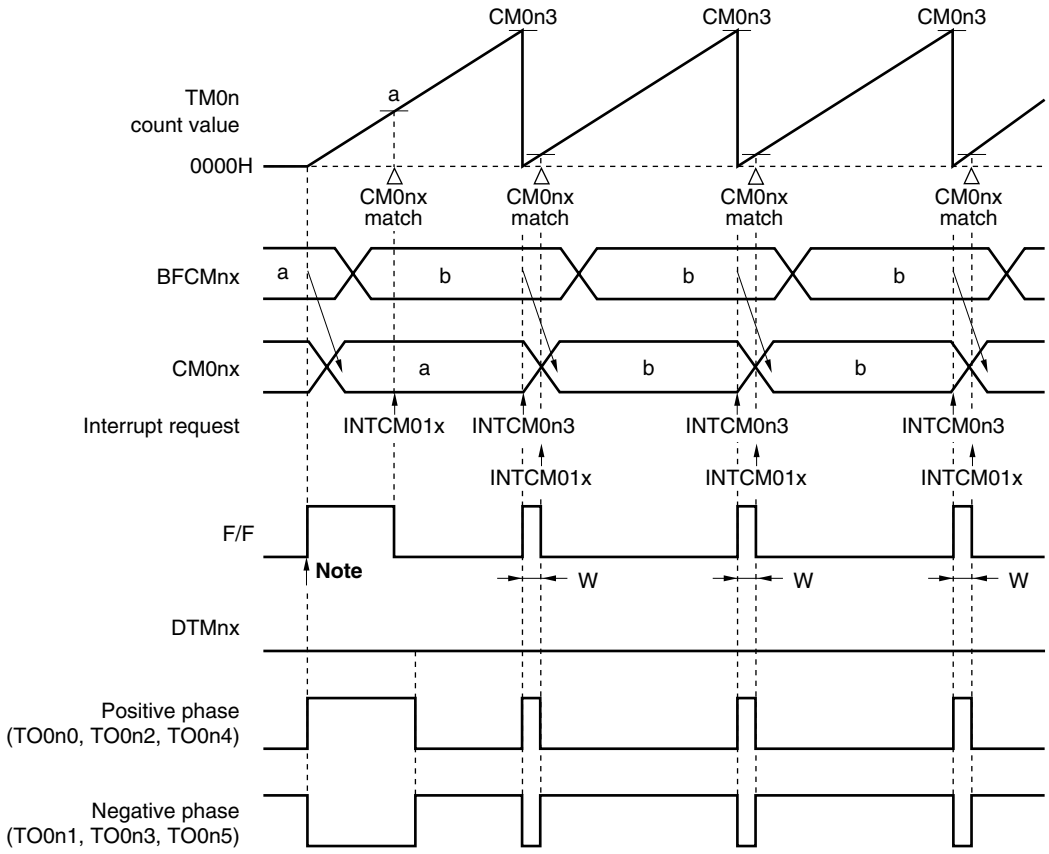
If CM0n0 to CM0n2 = 0000H has been set, the output waveform resulting from the TM0n count clock rate and the DTRRn set value differ.

★ **(d) When BFCMnx = 0000H is set while DTMnx = 000H or TM0CEDn bit = 1**

A pulse equivalent to one count clock of the timer is output.

Figure 9-36. Operation Timing in PWM Mode 2 (Sawtooth Wave, BFCMnx = 0000H) While DTMnx = 000H or TM0CEDn Bit = 1 (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Note Set at the rising edge of the TM0CEn bit.

Remarks 1. n = 0, 1

2. x = 0 to 2

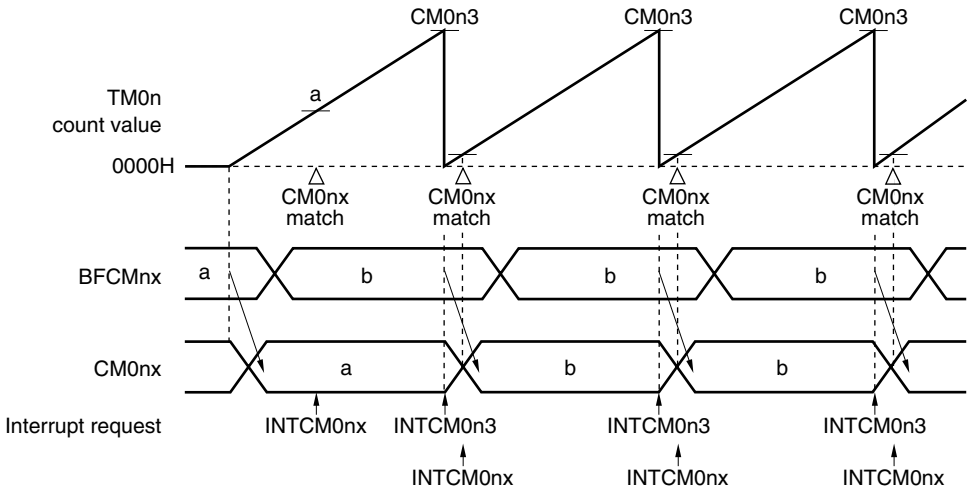
3. The above figure shows an active-high case.

4. W: Width of a pulse equivalent to one count clock of the timer from CM0n3 match

5. INTCM01x is generated on a match between TM01 and CM01x (a in the above figure). INTCM00x is not generated.

Figure 9-36. Operation Timing in PWM Mode 2 (Sawtooth Wave, BFCMnx = 0000H While DTMnx = 000H or TM0CEDn Bit = 1) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. n = 0, 1

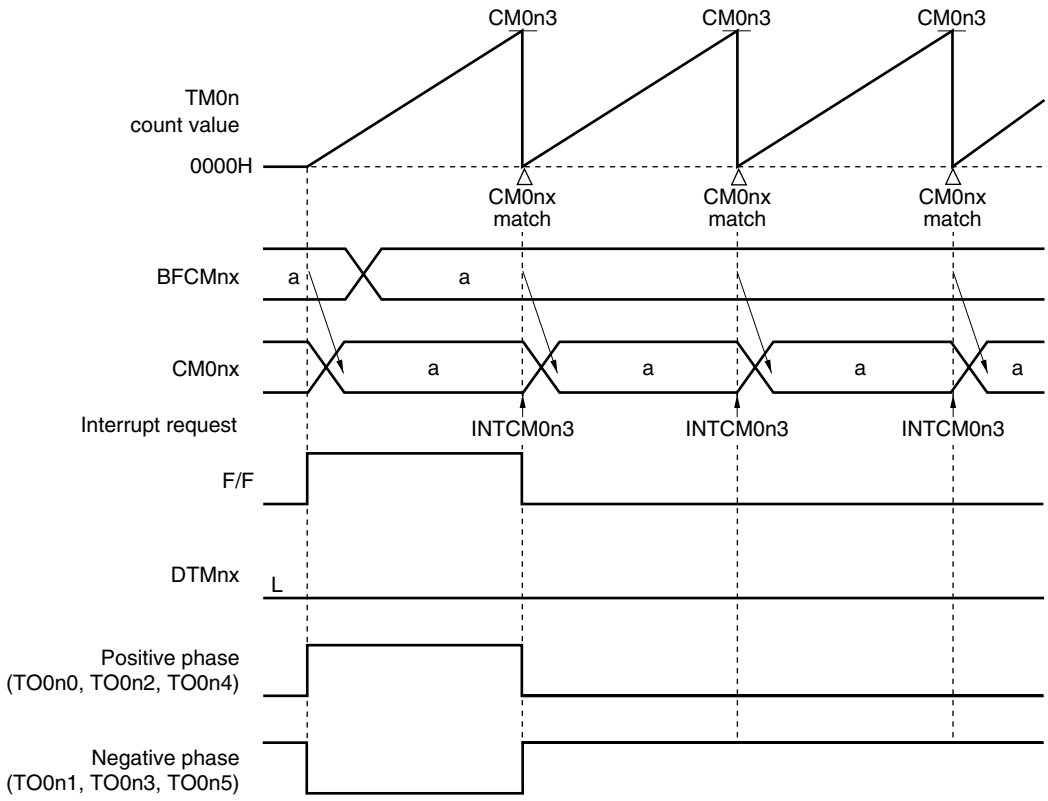
2. x = 4, 5

3. INTCM0nx is generated on a match between TM0n and CM0nx (a in the above figure).

★ **(e) When BFCMnx = CM0n3 = a is set**

Figure 9-37. Operation Timing in PWM Mode 2 (Sawtooth Wave, BFCMnx = CM0n3 = a) (When DTRRn = 0000H, TM0CEDn Bit of TMC0n Register = 1, ALVTO Bit of TOMRn Register = 1 (PWM Driving, Active Level = High) Are Set) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Remarks 1. n = 0, 1

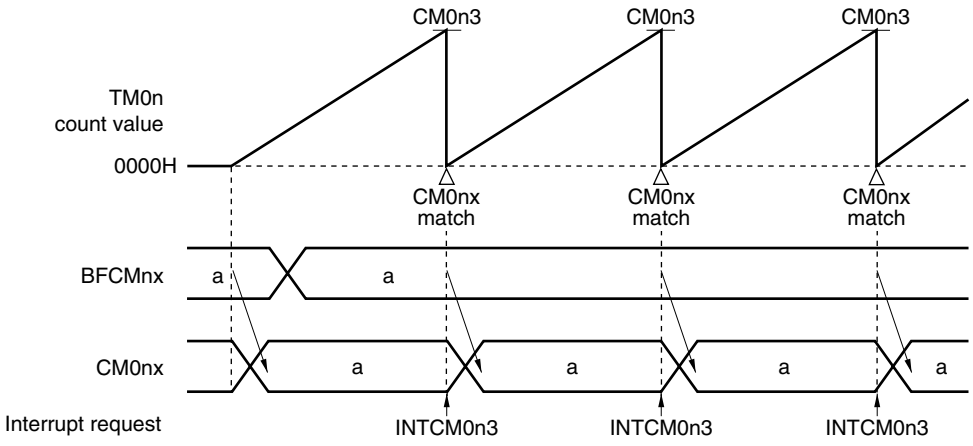
2. x = 0 to 2

3. The above figure shows an active-low case.

4. For the timing including the dead time, refer to **Figure 9-35**.

Figure 9-37. Operation Timing in PWM Mode 2 (Sawtooth Wave, BFCMnx = CM0n3 = a) (When DTRRn = 0000H, TM0CEDn Bit of TMC0n Register = 1, ALVTO Bit of TOMRn Register = 1 (PWM Driving, Active Level = High) Are Set) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



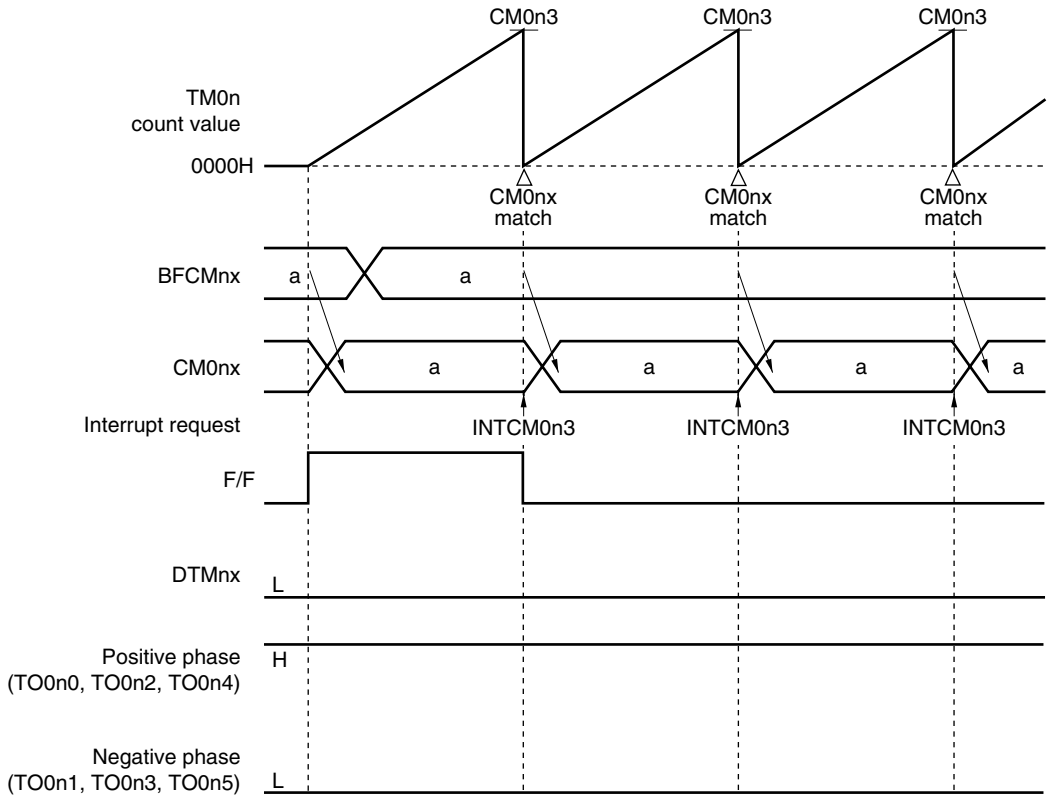
Remarks 1. n = 0, 1

2. x = 4, 5

3. For the timing including the dead time, refer to **Figure 9-35**.

Figure 9-38. Operation Timing in PWM Mode 2 (Sawtooth Wave, BFCMnx = CM0n3 = a) (When DTRRn = 0000H, TM0CEDn Bit of TMC0n Register = 1, ALVTO Bit of TOMRn Register = 0 (PWM Driving, Active Level = Low) Are Set) (1/2)

(a) Operation timing of compare registers 0n0 to 0n2 (CM0n0 to CM0n2)



Remarks 1. n = 0, 1

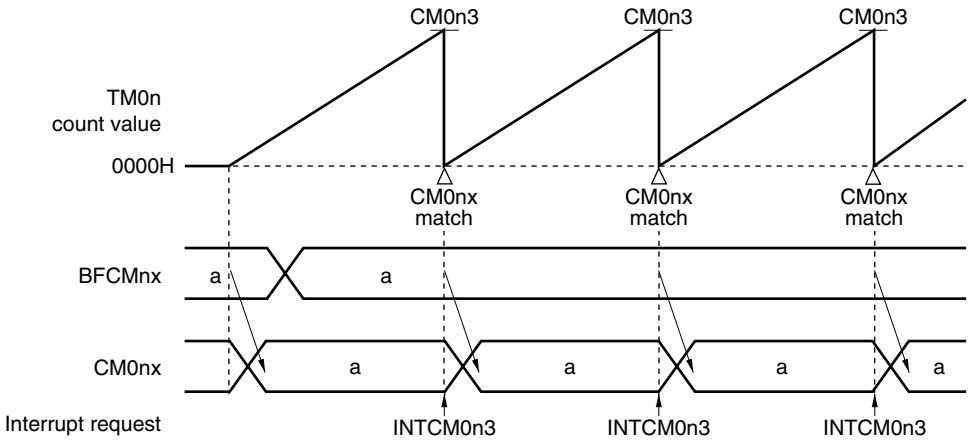
2. x = 0 to 2

3. The above figure shows an active-high case.

4. For the timing including the dead time, refer to **Figure 9-35**.

Figure 9-38. Operation Timing in PWM Mode 2 (Sawtooth Wave, BFCMnx = CM0n3 = a) (When DTRRn = 0000H, TM0CEDn Bit of TMC0n Register = 1, ALVTO Bit of TOMRn Register = 0 (PWM Driving, Active Level = Low) Are Set) (2/2)

(b) Operation timing of compare registers 0n4 and 0n5 (CM0n4, CM0n5)



Remarks 1. n = 0, 1

2. x = 4, 5

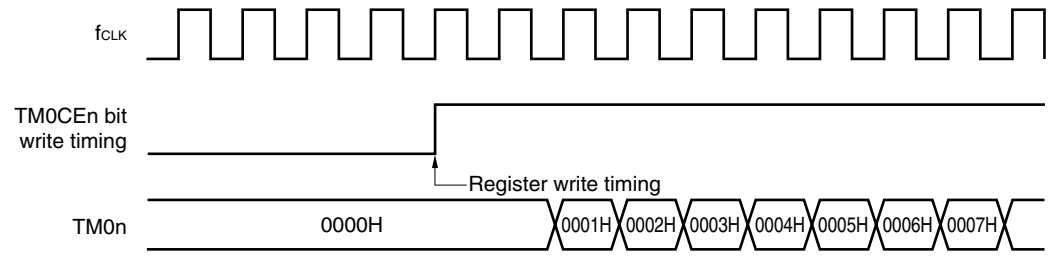
3. For the timing including the dead time, refer to **Figure 9-35**.

9.1.7 Operation timing

(1) TM0CEn bit write and TM0n timer operation timing

Figure 9-39 shows the timing from when the TM0CEn bit of the TMC0n register is written until the TM0n timer starts operating.

Figure 9-39. TM0CEn Bit Write and TM0n Timer Operation Timing



Caution The operation of TM0n starts $2f_{CLK}$ after the register write timing.

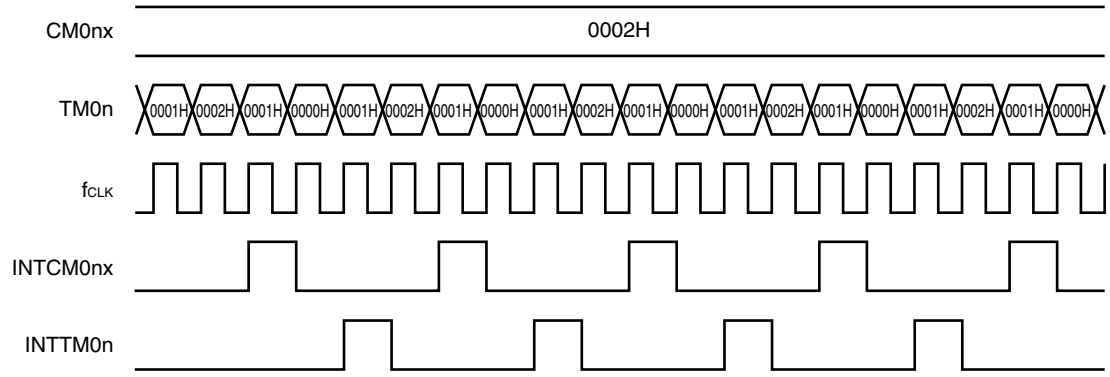
Remark f_{CLK}: Base clock

(2) Interrupt generation timing

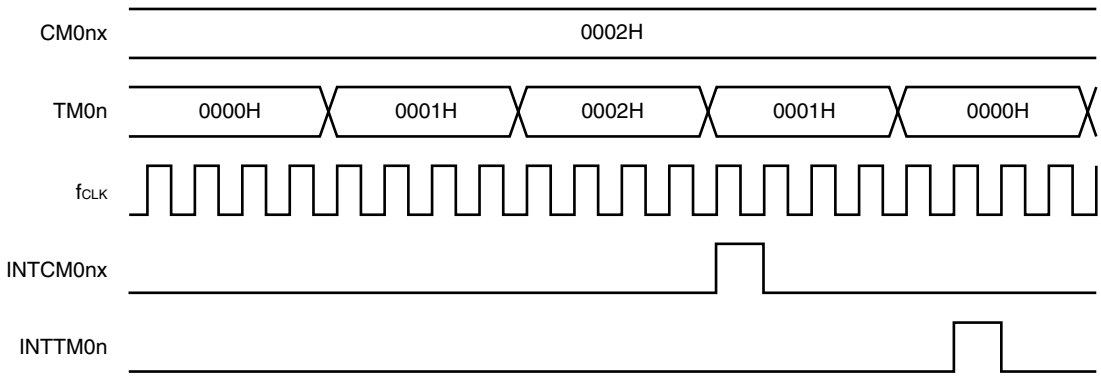
The interrupt generation timing at the TM0n count clock settings (PRM02 to PRM00 bits of the TMC0n register) in the various modes is described below.

Figure 9-40. Interrupt Generation Timing in PWM Mode 0 (Symmetric Triangular Wave), PWM Mode 1 (Asymmetric Triangular Wave)

(a) When count clock = f_{CLK}



(b) When count clock = $f_{CLK}/4$



Cautions 1. INTCM0nx is generated at the next f_{CLK} after detection of a TM0n and CM0nx match.

2. INTTM0n is generated at the next f_{CLK} after detection of a TM0n and 0000H match.

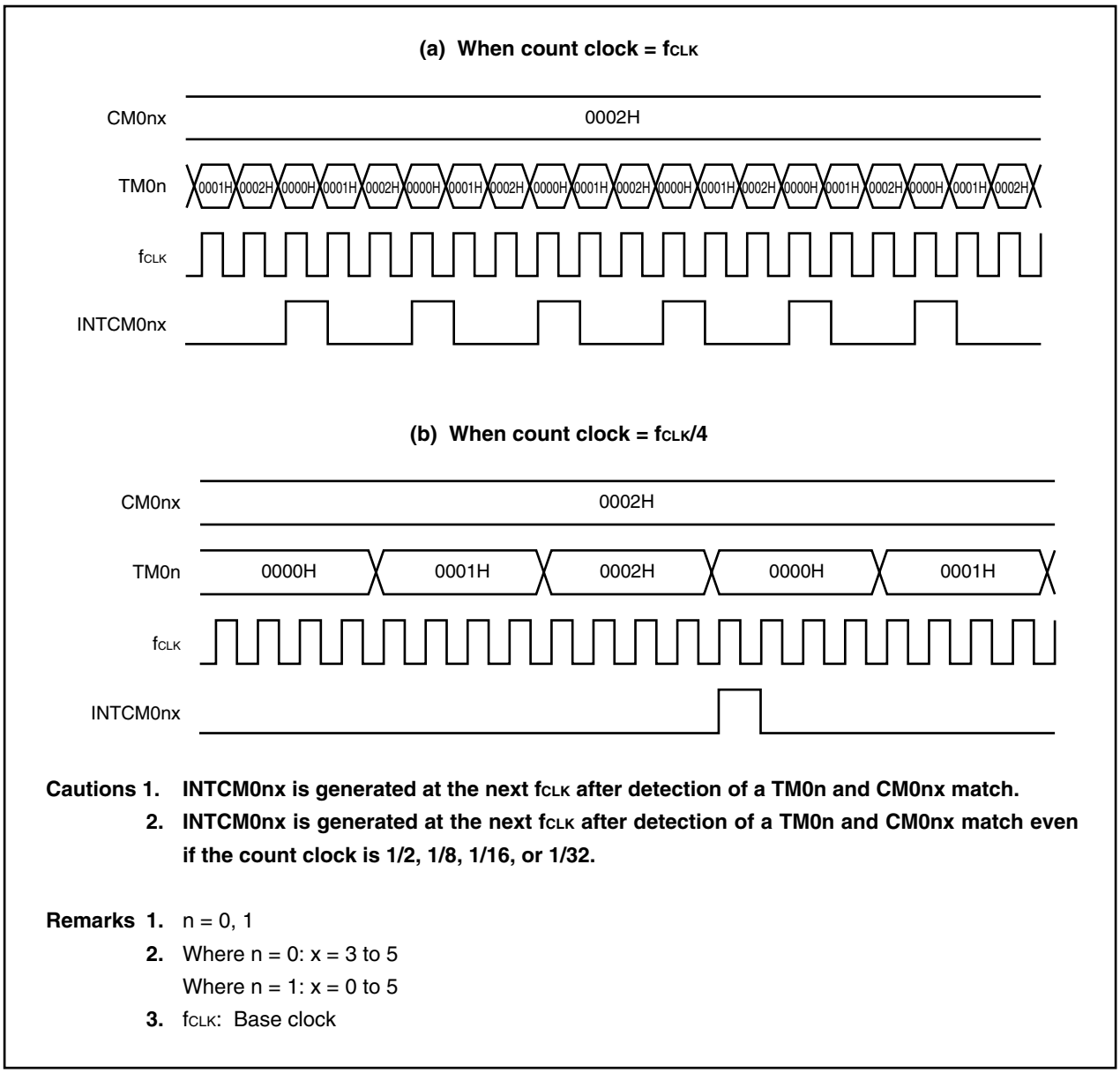
3. INTTM0n is generated at the next f_{CLK} after detection of a TM0n and 0000H match, even if the count clock is 1/2, 1/8, 1/16, or 1/32.

Remarks 1. n = 0, 1

2. Where n = 0: x = 3 to 5
Where n = 1: x = 0 to 5

3. f_{CLK}: Base clock

Figure 9-41. Interrupt Generation Timing in PWM Mode 2 (Sawtooth Wave)



Cautions 1. INTCM0nx is generated at the next f_{CLK} after detection of a TM0n and CM0nx match.

2. INTCM0nx is generated at the next f_{CLK} after detection of a TM0n and CM0nx match even if the count clock is 1/2, 1/8, 1/16, or 1/32.

Remarks 1. n = 0, 1

2. Where n = 0: x = 3 to 5
Where n = 1: x = 0 to 5

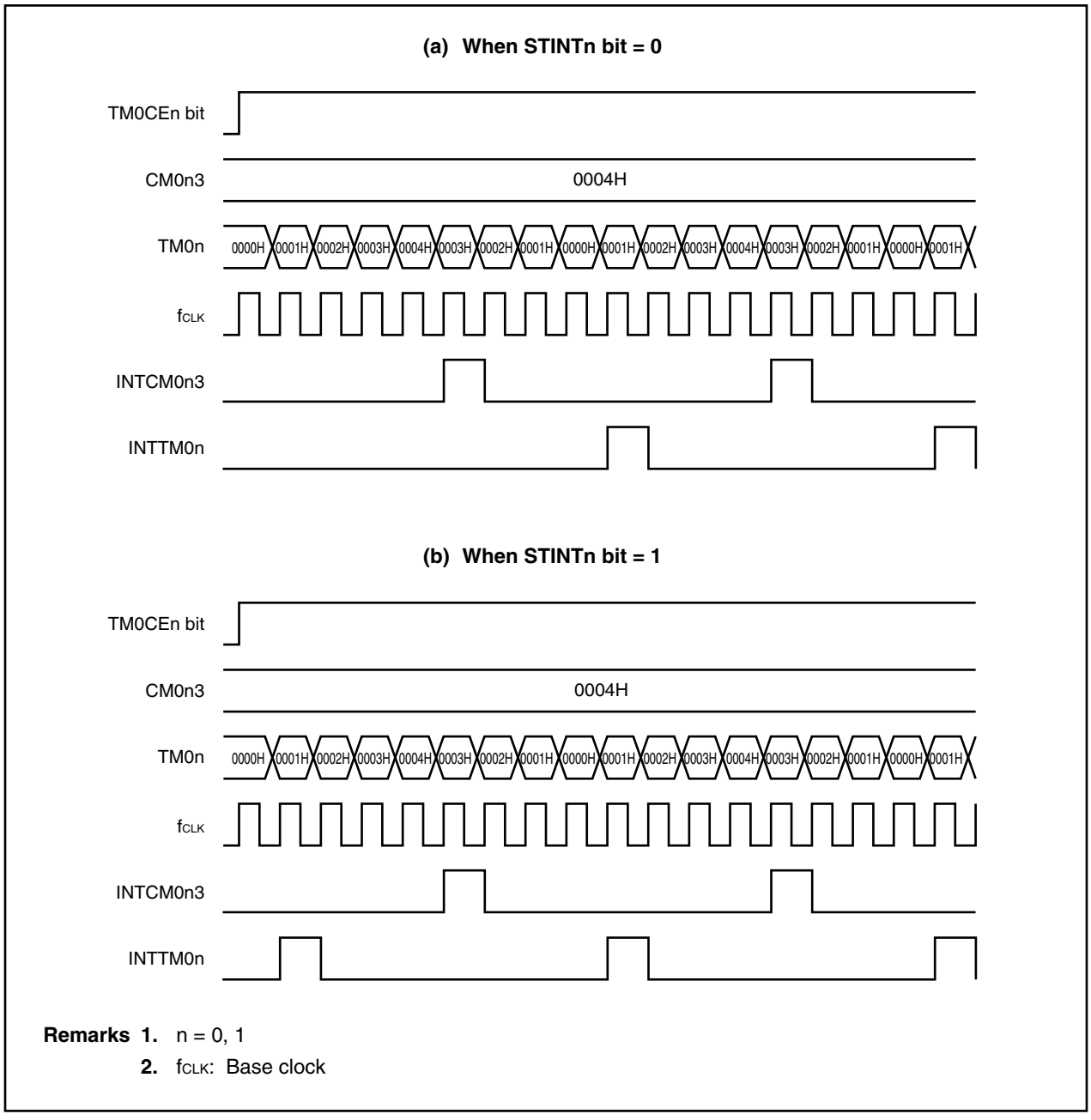
3. f_{CLK}: Base clock

(3) Relationship between interrupt generation and STINTn bit of TMC0n register

The interrupt generation timing for the setting of the STINTn bit of the TMC0n register and the interrupt culling ratio setting (bits CUL02 to CUL00) in the various modes is described below.

If, to realize the INTTM0n and INTCM0n3 interrupt culling function for TM0n, bits CUL02 to CUL00 of the TMC0n register are set for a culling ratio other than 1/1, and count operation is started, the interrupt output order differs according to the setting of the STINTn bit when counting starts.

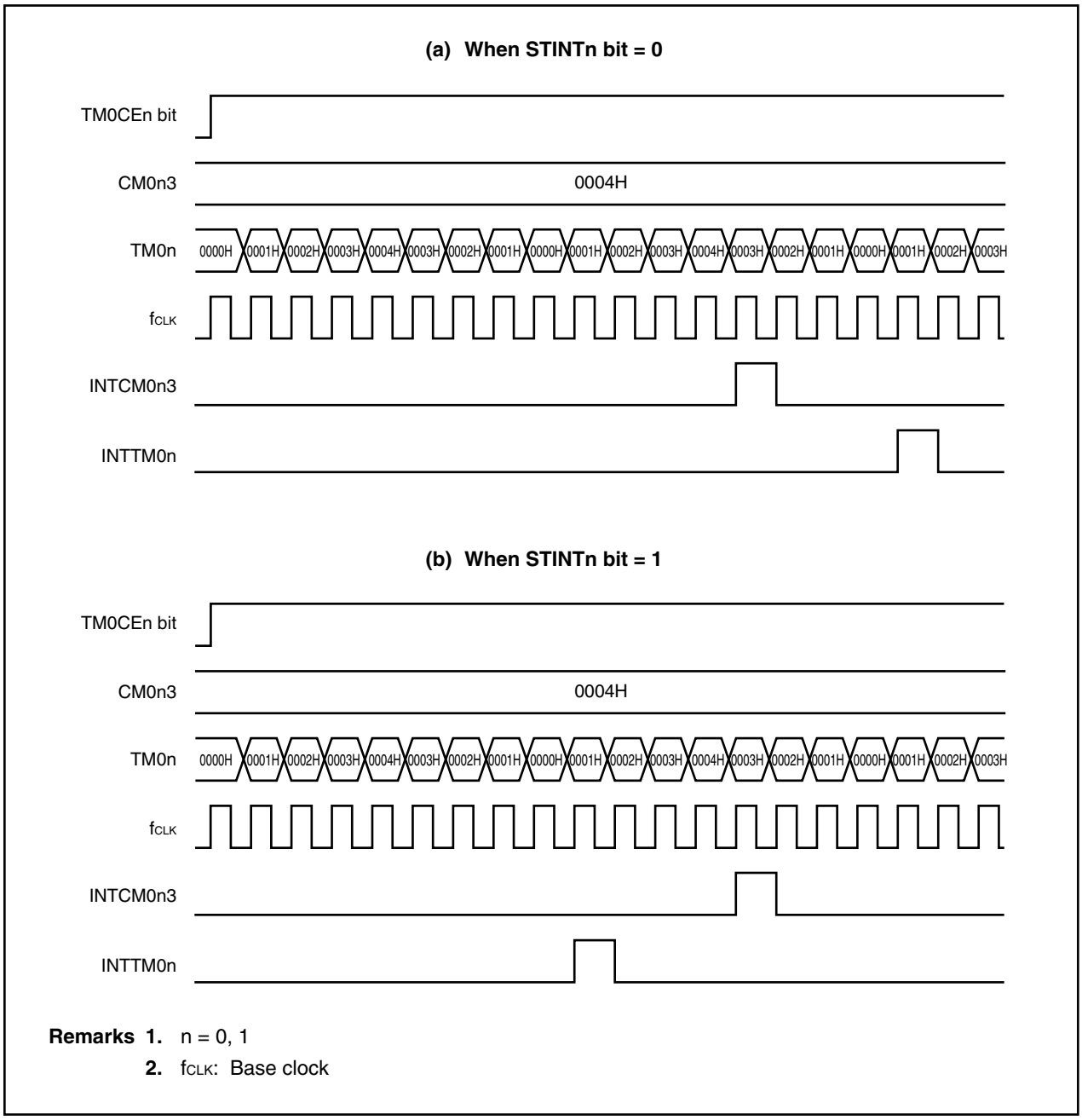
Figure 9-42. Interrupt Generation Timing in PWM Mode 0 (Symmetric Triangular Wave), PWM Mode 1 (Asymmetric Triangular Wave): In Case of Interrupt Culling Ratio of 1/1



Remarks 1. n = 0, 1

2. f_{CLK}: Base clock

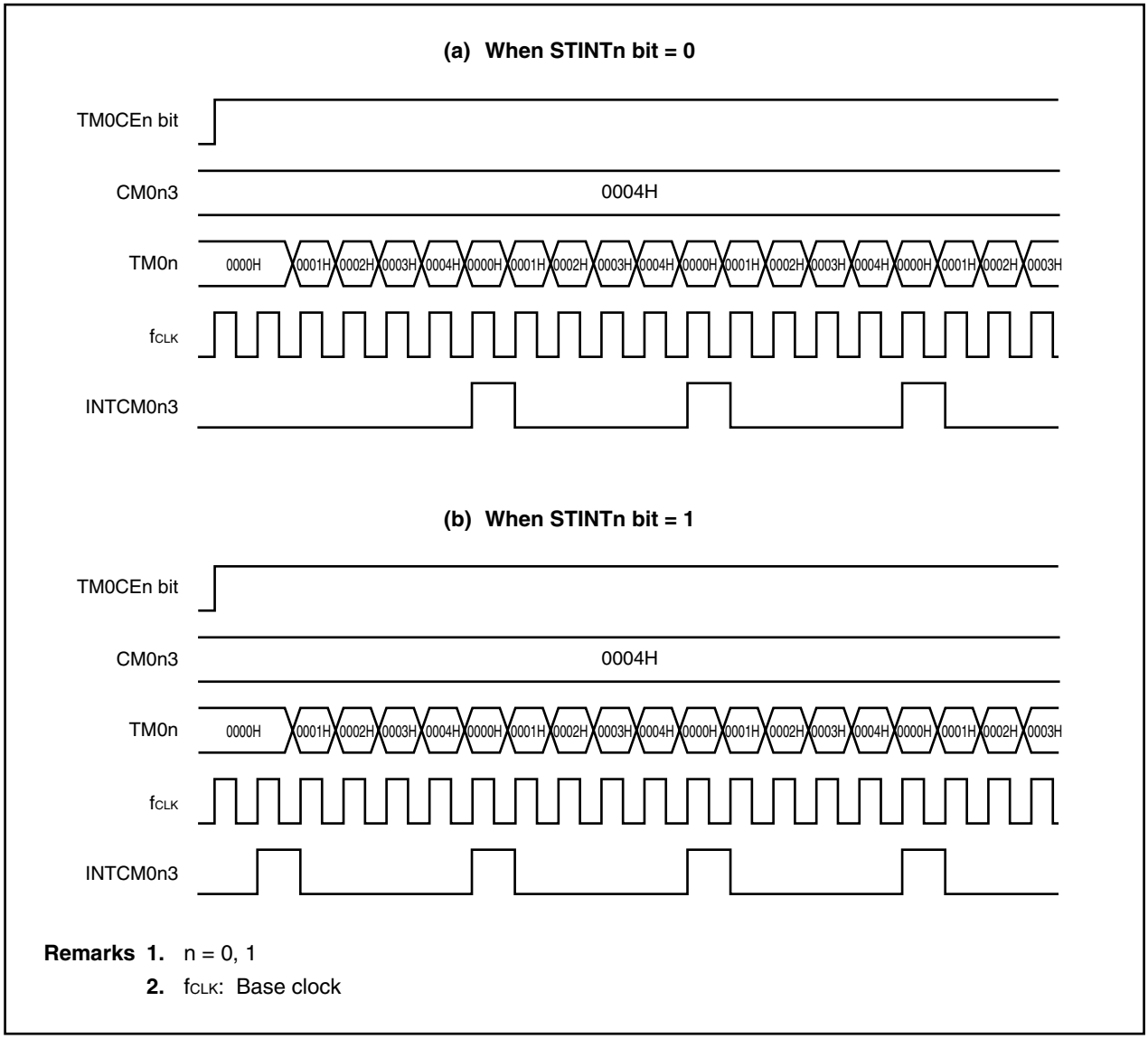
Figure 9-43. Interrupt Generation Timing in PWM Mode 0 (Symmetric Triangular Wave), PWM Mode 1 (Asymmetric Triangular Wave): In Case of Interrupt Culling Ratio of 1/2



Remarks 1. n = 0, 1

2. f_{CLK}: Base clock

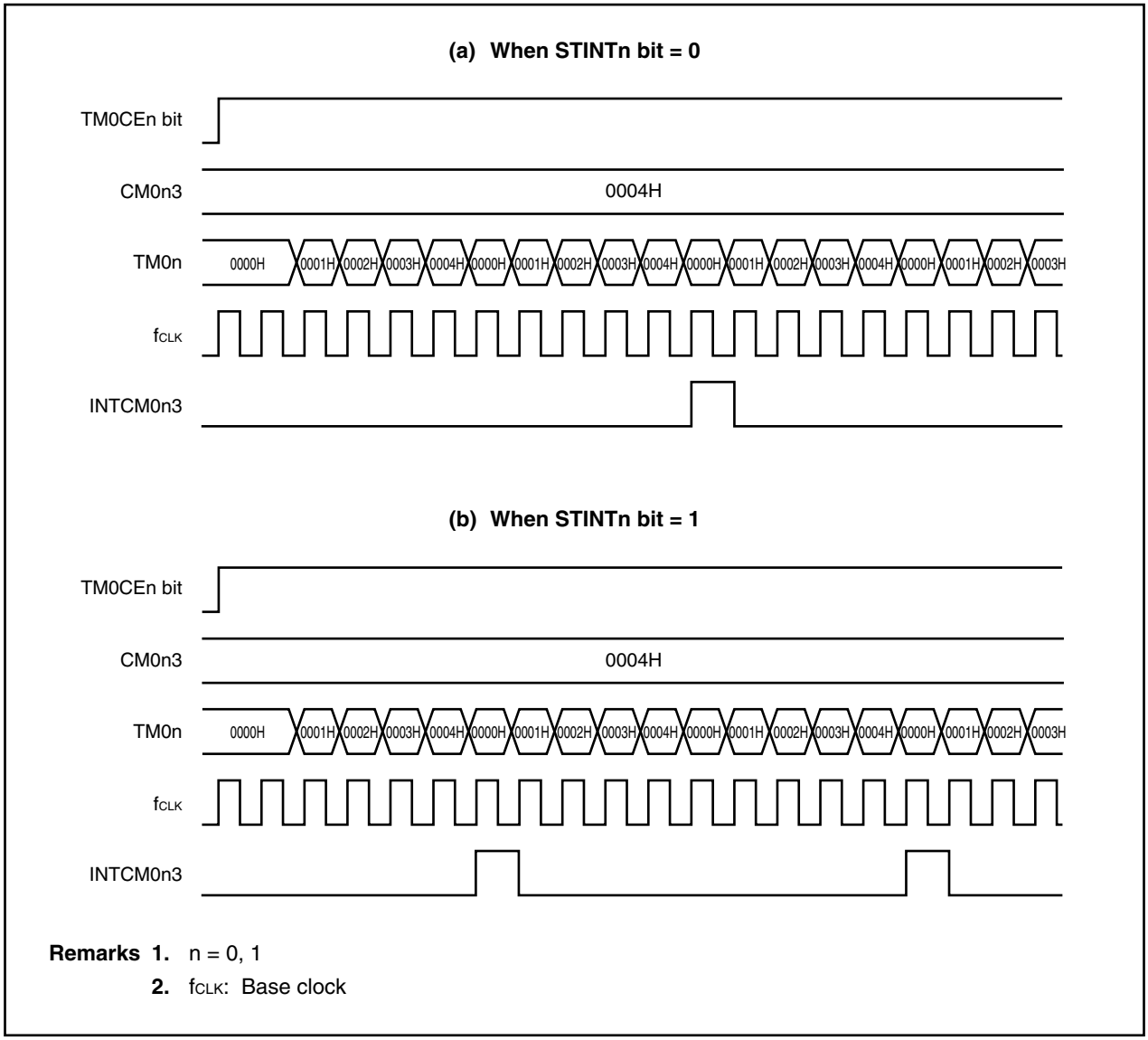
Figure 9-44. Interrupt Generation Timing in PWM Mode 2 (Sawtooth Wave): In Case of Interrupt Culling Ratio of 1/1



Remarks 1. n = 0, 1

2. f_{CLK}: Base clock

Figure 9-45. Interrupt Generation Timing in PWM Mode 2 (Sawtooth Wave): In Case of Interrupt Culling Ratio of 1/2

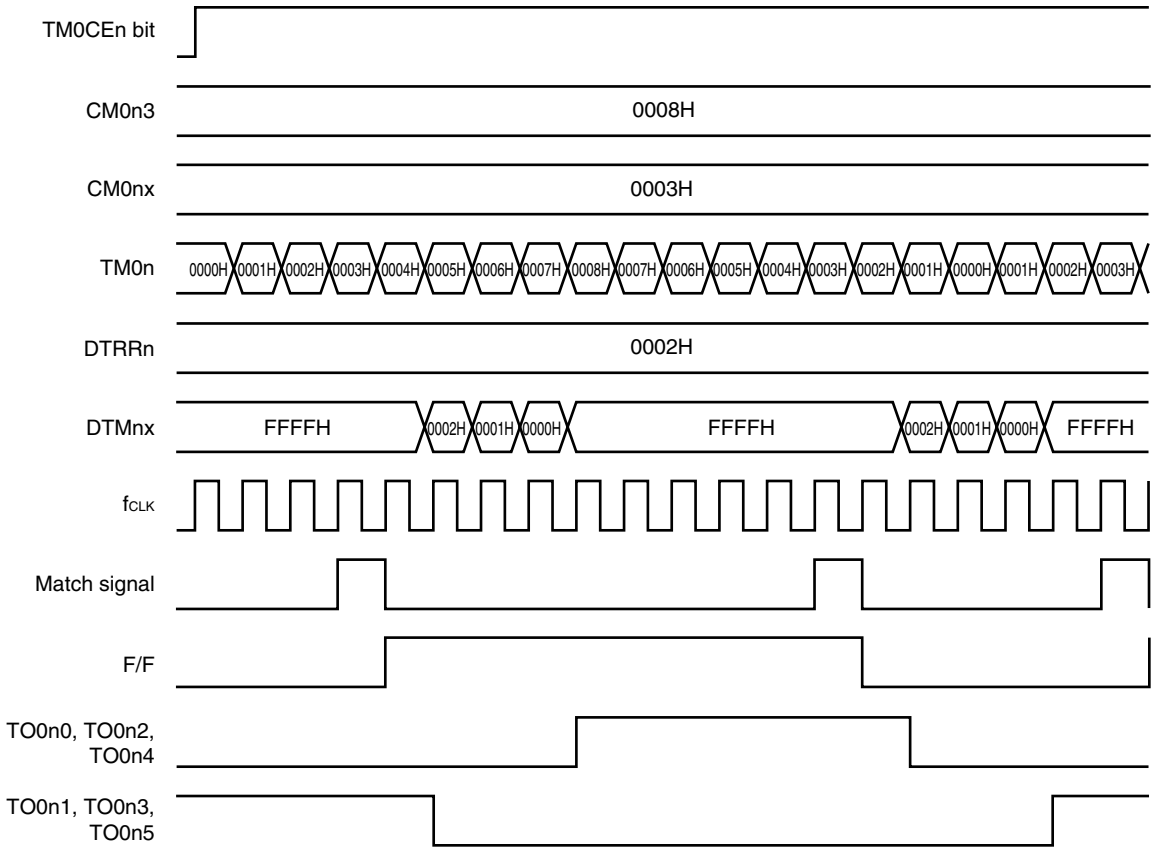


Remarks 1. n = 0, 1

2. f_{CLK}: Base clock

(4) TO0n0 to TO0n5 output timing

Figure 9-46. TO0n0 to TO0n5 Output Timing in PWM Mode 0 (Symmetric Triangular Wave), PWM Mode 1 (Asymmetric Triangular Wave)



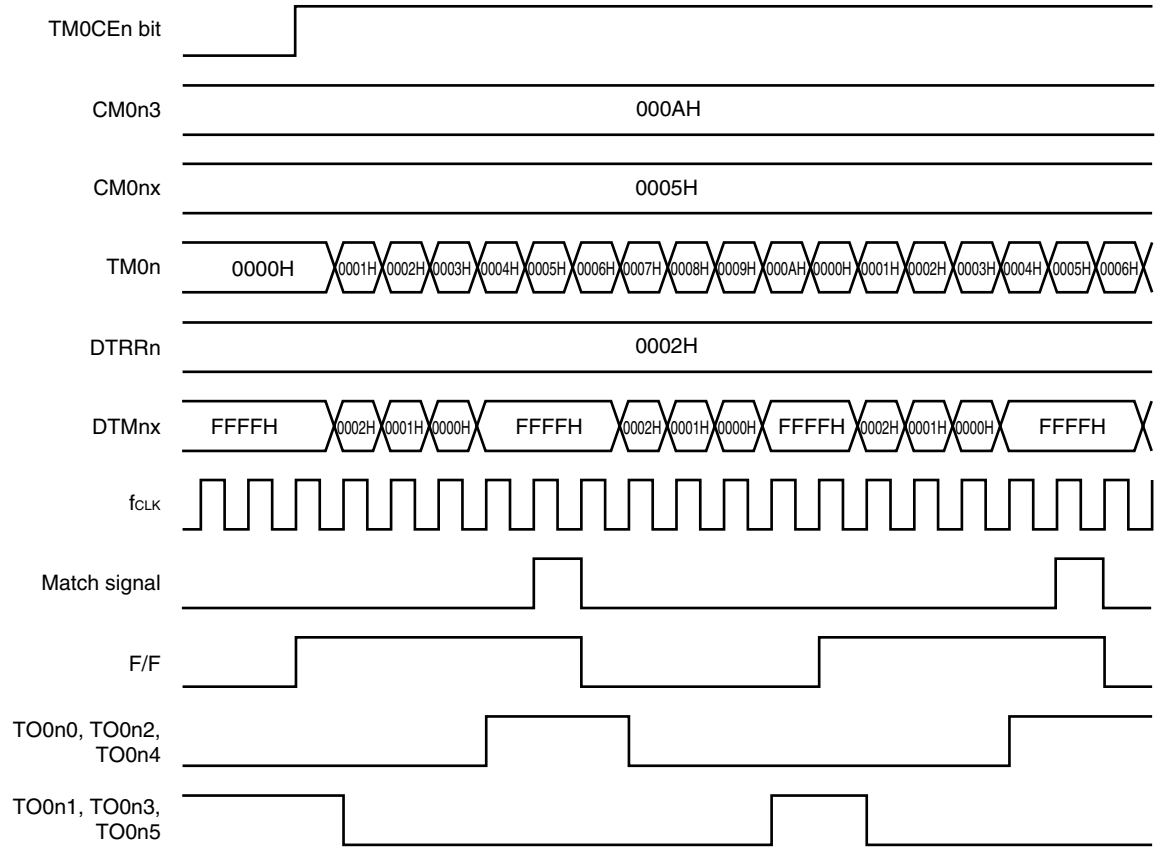
Remarks 1. The above figure shows the timing until the compare register and the TM0n timer match and the TO0n0 to TO0n5 outputs change.

2. x = 0 to 2

3. n = 0, 1

4. f_{CLK}: Base clock

Figure 9-47. TO0n0 to TO0n5 Output Timing in PWM Mode 2 (Sawtooth Wave)



Remarks 1. The above figure shows the timing until the compare register and the TM0n timer match and the TO0n0 to TO0n5 outputs change.

2. x = 0 to 2

3. n = 0, 1

4. f_{CLK}: Base clock

9.2 Timer 1

9.2.1 Features (timer 1)

Timer 10 (TM10) is a 16-bit up/down counter that performs the following operations.

- General-purpose timer mode (See **9.2.5 (1) Operation in general-purpose timer mode**.)
  Free-running timer
  PWM output
- Up/down counter mode (See **9.2.5 (2) Operation in UDC mode**.)
  UDC mode A (mode 1, mode 2, mode 3, mode 4)
  UDC mode B (mode 1, mode 2, mode 3, mode 4)

9.2.2 Function overview (timer 1)

- 16-bit 2-phase encoder input up/down counter & general-purpose timer (TM10)
- Compare registers: 2
- Capture/compare registers: 2
- Interrupt request sources
  - Capture/compare match interrupt: 2 types
  - Compare match interrupt request: 2 types
- Capture request signal: 2 types
  - The TM10 value can be latched using the valid edge of the INTP100 and INTP101 pins corresponding to the capture/compare register as the capture trigger.
- Count clock selectable through division by prescaler (set the frequency of the count clock to 10 MHz or less)
- Base clock (f_{CLK}): 1 type (set f_{CLK} to 20 MHz or less)
  $f_{XX}/2$
- Prescaler division ratio
  The following division ratios can be selected according to the base clock (f_{CLK}).

Division Ratio	Base Clock (f_{CLK})
1/2	$f_{XX}/4$
1/4	$f_{XX}/8$
1/8	$f_{XX}/16$
1/16	$f_{XX}/32$
1/32	$f_{XX}/64$
1/64	$f_{XX}/128$
1/128	$f_{XX}/256$

- PWM output function
  In the general-purpose timer mode, 16-bit resolution PWM can be output from the TO10 pin.
- Timer clear
  The following timer clear operations are performed according to the mode that is used.

  (a) General-purpose timer mode: Timer clear operation is possible upon occurrence of match with CM100 set value.
  (b) Up/down counter mode: The timer clear operation can be selected from among the following four conditions.
    (i) Timer clear performed upon occurrence of match with CM100 set value during TM10 up count operation, and timer clear performed upon occurrence of match with CM101 set value during TM10 down count operation.
    (ii) Timer clear performed only by external input.
    (iii) Timer clear performed upon occurrence of match between TM10 count value and CM100 set value.
    (iv) Timer clear performed upon occurrence of external input and match between TM10 count value and CM100 set value.
- External pulse output (TO10): 1

Remark fxx: Internal system clock

9.2.3 Basic configuration

The basic configuration is shown below.

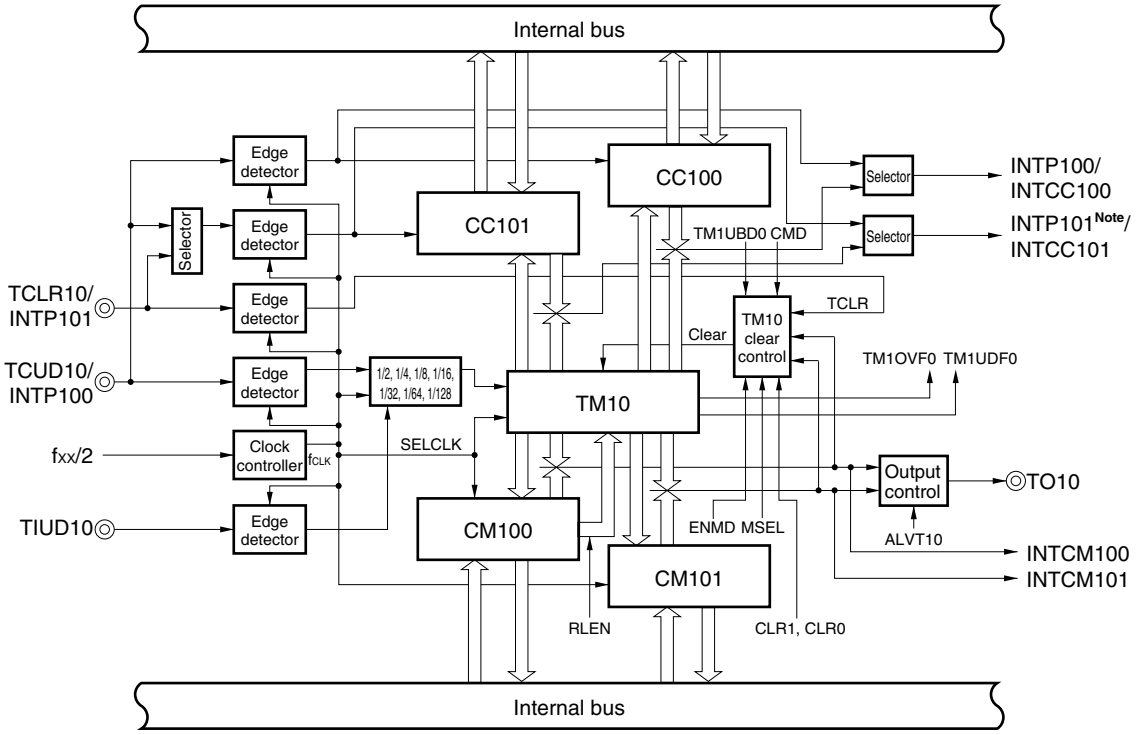
Table 9-5. Timer 1 Configuration List

Timer	Count Clock	Register	Read/Write	Generated Interrupt Signal	Capture Trigger
Timer 1	fxx/4, fxx/8, fxx/16, fxx/32, fxx/64, fxx/128, fxx/256	TM10	Read/write	–	–
		CM100	Read/write	INTCM100	–
		CM101	Read/write	INTCM101	–
		CC100	Read/write	INTCC100	INTP100
		CC101	Read/write	INTCC101	INTP100 or INTP101

Remark fxx: Internal system clock

Figure 9-48 shows the block diagram of timer 1.

Figure 9-48. Block Diagram of Timer 1



Note The INT101 interrupt is the signal of the capture trigger signal from the INTP101 pin or the capture trigger signal from the INTP100 pin, selected by the CSL0 bit of the CSL10 register.

Remarks 1. fxx: Internal system clock

2. fCLK: Base clock (20 MHz (MAX.))

(1) Timer 10 (TM10)

TM10 is a general-purpose timer (in general-purpose mode) and 2-phase encoder input up/down counter (in UDC mode).

This timer counts up in the general-purpose timer mode and counts up/down in the UDC mode.

It can be read/written in 16-bit units.

Cautions 1. Writing to TM10 is enabled only when the TM1CE0 bit of the TMC10 register is 0 (count operation disabled).

2. Continuous reading of TM10 is prohibited. If TM10 is continuously read, the second read value may differ from the actual value. If TM10 must be read twice, be sure to read another register between the first and the second read operation.

★ **3. Writing the same value to the TM10, CC100, and CC101 registers, and the STATUS10 register is prohibited.**

Writing the same value to the CCR0, TUM0, TMC10, SESA10, and PRM10 registers, and CM100 and CM101 registers is permitted (writing the same value is guaranteed even during a count operation).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
TM10																	FFFFF5E0H	0000H

TM10 start and stop is controlled by the TM1CE0 bit of timer control register 10 (TMC10).

The TM10 operation consists of the following two modes.

(a) General-purpose timer mode

In the general-purpose timer mode, TM10 operates as a 16-bit interval timer, free-running timer, or PWM output.

Counting is performed based on the clock selected by software.

Division by the prescaler can be selected for the count clock from among $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/16$, $f_{CLK}/32$, $f_{CLK}/64$, or $f_{CLK}/128$ using the PRM12 to PRM10 bits of prescaler mode register 10 (PRM10). (f_{CLK}: base clock, refer to **9.2.4 (1) Timer 1/timer 2 clock selection register (PRM02)**).

(b) Up/down counter mode (UDC mode)

In the UDC mode, TM10 functions as a 16-bit up/down counter that performs counting based on the TCUD10 and TIUD10 input signals.

This mode is divided into the UDC mode A and UDC mode B, depending on the condition of clearing TM10.

The conditions for clearing TM10 are as follows, depending on the operation mode.

Table 9-6. Timer 1 (TM10) Clear Conditions

Operation Mode	TUM0 Register		TMC10 Register			TM10 Clear
	CMD Bit	MSEL Bit	ENMD Bit	CLR1 Bit	CLR0 Bit	
General-purpose timer mode	0	0	0	×	×	Clearing not performed (free-running timer)
			1	×	×	Cleared upon match with CM100 set value
UDC mode A	1	0	×	0	0	Cleared only by TCLR10 input
			×	0	1	Cleared upon match with CM100 set value during up count operation
			×	1	0	Cleared by TCLR10 input or upon match with CM100 set value during up count operation
			×	1	1	Clearing not performed
UDC mode B	1	1	×	×	×	Cleared upon match with CM100 set value during up count operation or upon match with CM101 set value during down count operation
Other than the above						Setting prohibited

Remark ×: Indicates that the set value of that bit is ignored.

9.2.4 Control registers

(1) Timer 1/timer 2 clock selection register (PRM02)

The PRM02 register is used to select the base clock (f_{CLK}) of timer 1 and timer 2.

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

Cautions 1. Always set 01H to this register before using the timers 1 and 2. Setting to other than 01H is prohibited.

2. Set f_{CLK} to 20 MHz or less.

	7	6	5	4	3	2	1	0	Address	After reset
PRM02	0	0	0	0	0	0	0	PRM2	FFFFF5D8H	00H

Bit position	Bit name	Function
0	PRM2	Specifies the base clock (f_{CLK}) of timer 1 and timer 2. 1: $f_{CLK} = f_{XX}/2$

Remark f_{XX}: Internal system clock

(2) Timer unit mode register 0 (TUM0)

The TUM0 register is an 8-bit register used to specify the TM10 operation mode or to control the operation of the PWM output pin.

TUM0 can be read/written in 8-bit or 1-bit units.

Cautions 1. Changing the value of the TUM0 register during TM10 operation (TM1CE0 bit of TMC10 register = 1) is prohibited.

2. When the CMD bit = 0 (general-purpose timer mode), setting MSEL = 1 (UDC mode B) is prohibited.

	7	6	5	4	3	2	1	0	Address	After reset
TUM0	CMD	0	0	0	TOE10	ALVT10	0	MSEL	FFFFF5EBH	00H

Bit position	Bit name	Function
7	CMD	Specifies TM10 operation mode. 0: General-purpose timer mode (up count) 1: UDC mode (up/down count)
3	TOE10	Specifies timer output (TO10) enable. 0: Timer output disabled 1: Timer output enabled **Caution When CMD bit = 1 (UDC mode), timer output is not performed regardless of the setting of the TOE10 bit. At this time, timer output consists of the negative phase level of the level set by the ALVT10 bit.**
2	ALVT10	Specifies active level of timer output (TO10). 0: Active level is high level 1: Active level is low level **Caution When CMD bit = 1 (UDC mode), timer output is not performed regardless of the setting of the TOE10 bit. At this time, timer output consists of the negative phase level of the level set by the ALVT10 bit.**
0	MSEL	Specifies operation in UDC mode (up/down count) 0: UDC mode A TM10 can be cleared by setting the CLR1, CLR0 bits of the TMC10 register. 1: UDC mode B TM10 is cleared in the following cases. • Upon match with CM100 during TM10 up count operation • Upon match with CM101 during TM10 down count operation When UDC mode B is set, the ENMD, CLR1, and CLR0 bits of the TMC10 register become invalid.

(3) Timer control register 10 (TMC10)

The TMC10 register is used to enable/disable TM10 operation and to set transfer and timer clear operations.

TMC10 can be read/written in 8-bit or 1-bit units.

Caution Changing the values of the TMC10 register bits other than the TM1CE0 bit during TM10 operation (TM1CE0 = 1) is prohibited.

(1/2)

	7	<6>	5	4	3	2	1	0	Address	After reset
TMC10	0	TM1CE0	0	0	RLEN	ENMD	CLR1	CLR0	FFFFF5ECH	00H

Bit position	Bit name	Function
6	TM1CE0	Enables/disables TM10 operation. 0: TM10 count operation disabled 1: TM10 count operation enabled
3	RLEN	Enables/disables transfer from CM100 to TM10. 0: Transfer disabled 1: Transfer enabled **Cautions 1. When RLEN = 1, the value set to CM100 is transferred to TM10 upon occurrence of a TM10 underflow.** **2. The RLEN bit is valid only in UDC mode A (TUM0 register's CMD bit = 1, MSEL bit = 0). In the general-purpose timer mode (CMD bit = 0) and in UDC mode B (CMD bit = 1, MSEL bit =1), a transfer operation is not performed even the RLEN bit is set (1).**
2	ENMD	Enables/disables clearing of TM10 in general-purpose timer mode (CMD bit of TUM0 register = 0). 0: Clear disabled (free-running mode) Clearing is not performed even when TM10 and CM100 values match. 1: Clear enabled Clearing is performed when TM10 and CM100 values match. **Caution The ENMD bit setting becomes invalid in UDC mode (CMD bit of TUM0 register = 1).**

(2/2)

Bit position	Bit name	Function
1, 0	CLR1, CLR0	Controls TM10 clear operation in UDC mode A.

CLR1	CLR0	Specifies TM10 clear source
0	0	Cleared only by external input (TCLR10)
0	1	Cleared upon match of TM10 count value and CM100 set value
1	0	Cleared by TCLR10 input or upon match of TM10 count value and CM100 set value
1	1	Not cleared

Cautions 1. Clearing by match of the TM10 count value and CM100 set value is valid only during a TM10 up count operation (TM10 is not cleared during a TM10 down count operation).

2. The CLR1 and CLR0 bit settings are invalid in general-purpose timer mode (CMD bit of TUM0 register = 0).

3. The CLR1 and CLR0 bit settings are invalid in UDC mode B (MSEL bit of TUM0 register = 1).

4. When clearing by TCLR10 has been enabled by bits CLR1 and CLR0, clearing is performed whether the value of the TM1CE0 bit is 1 or 0.

(4) Capture/compare control register 0 (CCR0)

The CCR0 register specifies the operation mode of the capture/compare registers (CC100, CC101).

CCR0 can be read/written in 8-bit or 1-bit units.

Caution Overwriting the CCR0 register during TM10 operation (TM1CE0 bit = 1) is prohibited.

	7	6	5	4	3	2	1	0	Address	After reset
CCR0	0	0	0	0	0	0	CMS1	CMS0	FFFFF5EAH	00H

Bit position	Bit name	Function
1	CMS1	Specifies operation mode of CC101. 0: Capture register 1: Compare register
0	CMS0	Specifies operation mode of CC100. 0: Capture register 1: Compare register

(5) Signal edge selection register 10 (SESA10)

The SESA10 register is used to specify the valid edge of external interrupt requests from external pins (INTP100, INTP101, TIUD10, TCUD10, TCLR10).

The valid edge (rising edge, falling edge, or both edges) can be specified independently for each pin.

SESA10 can be read/written in 8-bit or 1-bit units.

Cautions 1. Changing the values of the SESA10 register bits during TM10 operation (TM1CE0 = 1) is prohibited.

2. Be sure to set (to 1) the TM1CE0 bit of timer control register 10 (TMC10) even when timer 1 is not used and the TCUD10/INTP100 and TCLR10/INTP101 pins are used as INTP100 and INTP101.

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
SESA10	TESUD01	TESUD00	CESUD01	CESUD00	IES1011	IES1010	IES1001	IES1000	FFFFF5EDH	00H
	TIUD10, TCUD10		TCLR10		INTP101		INTP100			

Bit position	Bit name	Function
7, 6	TESUD01, TESUD00	Specifies valid edge of pins TIUD10, TCUD10.

TESUD01	TESUD00	Valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Cautions 1. The set values of the TESUD01 and TESUD00 bits are only valid in UDC mode A and UDC mode B.

2. If mode 4 is specified as the operation mode of TM10 (specified by the PRM12 to PRM10 bits of the PRM10 register), the valid edge specifications for the TIUD10 and TCUD10 pins (bits TESUD01 and TESUD00) are not valid.

(2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>5, 4</td><td>CESUD01,
CESUD00</td><td>Specifies valid edge of TCLR10 pin.
<table>
<tr><th>CESUD01</th><th>CESUD00</th><th>Valid edge</th></tr>
<tr><td>0</td><td>0</td><td>Falling edge</td></tr>
<tr><td>0</td><td>1</td><td>Rising edge</td></tr>
<tr><td>1</td><td>0</td><td>Low level</td></tr>
<tr><td>1</td><td>1</td><td>High level</td></tr>
</table>
The set values of bits CESUD01 and CESUD00 and the TM10 operation are related as follows.
00: TM10 cleared after detection of falling edge of TCLR10
01: TM10 cleared after detection of rising edge of TCLR10
10: TM10 cleared status held while TCLR10 input is low level
11: TM10 cleared status held while TCLR10 input is high level

Caution The set values of the CESUD01 and CESUD00 bits are valid only in UDC mode A.</td></tr>
<tr><td>3, 2</td><td>IES1011,
IES1010</td><td>Specifies valid edge of the pin (INTP101/INTP100) selected by the CSL0 bit of the CSL10 register.
<table>
<tr><th>IES1011</th><th>IES1010</th><th>Valid edge</th></tr>
<tr><td>0</td><td>0</td><td>Falling edge</td></tr>
<tr><td>0</td><td>1</td><td>Rising edge</td></tr>
<tr><td>1</td><td>0</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>Both rising and falling edges</td></tr>
</table></td></tr>
<tr><td>1, 0</td><td>IES1001,
IES1000</td><td>Specifies valid edge of INTP100 pin.
<table>
<tr><th>IES1001</th><th>IES1000</th><th>Valid edge</th></tr>
<tr><td>0</td><td>0</td><td>Falling edge</td></tr>
<tr><td>0</td><td>1</td><td>Rising edge</td></tr>
<tr><td>1</td><td>0</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>Both rising and falling edges</td></tr>
</table></td></tr>
</table>

(6) Prescaler mode register 10 (PRM10)

The PRM10 register is used to perform the following selections.

- Selection of count clock in general-purpose timer mode (CMD bit of TUM0 register = 0)
- Selection of count operation mode in UDC mode (CMD = 1)

PRM10 can be read/written in 8-bit or 1-bit units.

Cautions 1. Overwriting the PRM10 register during TM10 operation (TM1CE0 bit = 1) is prohibited.

2. Clearing the PRM12 bit to 0 is prohibited in UDC mode (CMD bit of TUM0 register = 1).

3. When TM10 is in mode 4, specification of the valid edge for the TIUD10 and TCUD10 pins is valid.

	7	6	5	4	3	2	1	0	Address	After reset
PRM10	0	0	0	0	0	PRM12	PRM11	PRM10	FFFFF5EEH	07H

Bit position	Bit name	Function
2 to 0	PRM12 to PRM10	Specifies the up/down count operation mode during input of the clock rate when the internal clock of the TM10 is used, or during external clock (TIUD10) input.

PRM12	PRM11	PRM10	CMD = 0	CMD = 1	
			Count clock	Count clock	UDC mode
0	0	0	Setting prohibited	Setting prohibited	
0	0	1	$f_{CLK}/2$		
0	1	0	$f_{CLK}/4$		
0	1	1	$f_{CLK}/8$		
1	0	0	$f_{CLK}/16$	TIUD10	Mode 1
1	0	1	$f_{CLK}/32$		Mode 2
1	1	0	$f_{CLK}/64$		Mode 3
1	1	1	$f_{CLK}/128$		Mode 4

Remark f_{CLK}: Base clock

(a) In general-purpose timer mode (CMD bit of TUM0 register = 0)

The count clock is specified by bits PRM12 to PRM10.

(b) UDC mode (CMD bit of TUM0 register = 1)

The TM10 count triggers in the UDC mode are as follows.

Operation Mode	TM10 Operation
Mode 1	Down count when TCUD10 = high level Up count when TCUD10 = low level
Mode 2	Up count upon detection of valid edge of TIUD10 input Down count upon detection of valid edge of TCUD10 input
★ Mode 3	Up count upon detection of valid edge of TIUD10 input when TCUD10 = high level Down count upon detection of valid edge of TIUD10 input when TCUD10 = low level
Mode 4	Automatic judgment upon detection of both edges of TIUD10 input and both edges of TCUD10 input

(7) Status register 0 (STATUS0)

The STATUS0 register indicates the operating status of TM10.

STATUS0 is read-only, in 8-bit or 1-bit units.

	7	6	5	4	3	<2>	<1>	<0>	Address	After reset
STATUS0	0	0	0	0	0	TM1UDF0	TM1OVF0	TM1UBD0	FFFFF5EFH	00H

Bit position	Bit name	Function
2	TM1UDF0	TM10 underflow flag 0: No TM10 count underflow 1: TM10 count underflow **Caution The TM1UDF0 bit is cleared (to 0) upon completion of a read access to the STATUS0 register from the CPU.**
1	TM1OVF0	TM10 overflow flag 0: No TM10 count overflow 1: TM10 count overflow **Caution The TM1OVF0 bit is cleared (to 0) upon completion of a read access to the STATUS0 register from the CPU.**
0	TM1UBD0	Indicates the operating status of TM10 up/down count. 0: TM10 up count in progress 1: TM10 down count in progress **Caution The state of the TM1UBD0 bit differs according to the mode as follows.** **• The TM1UBD0 bit is fixed to 0 in general-purpose timer mode (CMD bit of TUM0 register = 0).** **• The TM1UBD0 bit indicates the TM10 up-/down-count status in UDC mode (CMD bit of TUM0 register = 1).**

(8) CC101 capture input selection register (CSL10)

The CSL10 register is used to select the INTP101 or INTP100 pin to input a capture signal when the CC101 register is used as a capture register.

CSL10 can be read/written in 8-bit or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
CSL10	0	0	0	0	0	0	0	CSL0	FFFFF5F6H	00H

Bit position	Bit name	Function
0	CSL0	Specifies capture input to CC101. 0: INTP101 1: INTP100

(9) Compare register 100 (CM100)

CM100 is a 16-bit register that always compares its value with the value of TM10. When the value of a compare register matches the value of TM10, an interrupt signal is generated. The interrupt generation timing in the various modes is described below.

- In the general-purpose timer mode (CMD bit of TUM0 register = 0) and UDC mode A (MSEL bit of TUM0 register = 0), an interrupt signal (INTCM100) is generated upon occurrence of a match.
- In UDC mode B (MSEL bit of TUM0 register = 1), an interrupt signal (INTCM100) is generated only upon occurrence of a match during a down count operation.

CM100 can be read/written in 16-bit units.

Caution When the TM1CE0 bit of the TMC10 register is 1, it is prohibited to overwrite the value of the CM100 register.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CM100																	FFFFF5E2H	0000H

(10) Compare register 101 (CM101)

CM101 is a 16-bit register that always compares its value with the value of TM10. When the value of the compare register matches the value of TM10, an interrupt signal is generated. The interrupt generation timing in the various modes is described below.

- In the general-purpose timer mode (CMD bit of TUM0 register = 0) and UDC mode A (MSEL bit of TUM0 register = 0), an interrupt signal (INTCM101) is generated upon occurrence of a match.
- In UDC mode B (MSEL bit of TUM0 register = 1), an interrupt signal (INTCM101) is generated only upon occurrence of a match during a down count operation.

CM101 can be read/written in 16-bit units.

Caution When the TM1CE0 bit of the TMC10 register is "1", it is prohibited to overwrite the value of the CM101 register.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CM101																	FFFFF5E4H	0000H

(11) Capture/compare register 100 (CC100)

CC100 is a 16-bit register. It can be specified as a capture register or as a compare register using capture/compare control register 0 (CCR0). CC100 can be read/written in 16-bit units.

Cautions 1. When used as a capture register (CMS0 bit of CCR0 register = 0), write access is prohibited.

2. When used as a compare register (CMS0 bit of CCR0 register = 1) during TM10 operation (TM1CE0 bit of TMC10 register = 1), overwriting the CC100 register values is prohibited.

3. When TM10 has been stopped (TM1CE0 bit of TMC10 register = 0), the capture trigger is disabled.

4. When the operation mode is changed from capture register to compare register, set a new compare value.

5. Continuous reading of CC100 is prohibited. If CC100 is continuously read, the second read value may differ from the actual value. If CC100 must be read twice, be sure to read another register between the first and the second read operation.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CC100																	FFFFF5E6H	0000H

(a) When set as a capture register

When CC100 is set as a capture register, the valid edge of the corresponding external interrupt signal (INTP100) is detected as the capture trigger. TM10 latches the count value in synchronization with the capture trigger (capture operation). The latched value is held in the capture register until the next capture operation.

The valid edge of external interrupts (rising edge, falling edge, both rising and falling edges) is selected by signal edge selection register 10 (SESA10).

When the CC100 register is specified as a capture register, interrupts are generated upon detection of the valid edge of the INTP100 signal.

(b) When set as a compare register

When CC100 is set as a compare register, it always compares its own value with the value of TM10. If the value of CC100 matches the value of the TM10, CC100 generates an interrupt signal (INTCC100).

(12) Capture/compare register 101 (CC101)

CC101 is a 16-bit register. It can be specified as a capture register or as a compare register using capture/compare control register 0 (CCR0). CC101 can be read/written in 16-bit units.

Cautions 1. When used as a capture register (CMS1 bit of CCR0 register = 0), write access is prohibited.

2. When used as a compare register (CMS1 bit of CCR0 register = 1) during TM10 operation (TM1CE0 bit of TMC10 register = 1), overwriting the CC101 register values is prohibited.

3. When TM10 has been stopped (TM1CE0 bit of TMC10 register = 0), the capture trigger is disabled.

4. When the operation mode is changed from capture register to compare register, newly set a compare value.

5. Continuous reading of CC101 is prohibited. If CC101 is continuously read, the second read value may differ from the actual value. If CC101 must be read twice, be sure to read another register between the first and the second read operation.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CC101																	FFFFF5E8H	0000H

(a) When set as a capture register

When CC101 is set as a capture register, the valid edge of either corresponding external interrupt signal (INTP100 or INTP101) is selected with the selector, and the valid edge of the selected external interrupt signal is detected as the capture trigger. TM10 latches the count value in synchronization with the capture trigger (capture operation). The latched value is held in the capture register until the next capture operation.

The valid edge of external interrupts (rising edge, falling edge, both rising and falling edges) is selected by signal edge selection register 10 (SESA10).

When the CC101 register is specified as a capture register, interrupts are generated upon detection of the valid edge of either the INTP100 or INTP101 signal.

(b) When set as a compare register

When CC101 is set as a compare register, it always compares its own value with the value of TM10. If the value of CC101 matches the value of the TM10, CC101 generates an interrupt signal (INTCC101).

9.2.5 Operation

(1) Operation in general-purpose timer mode

TM10 can perform the following operations in the general-purpose timer mode.

(a) Interval operation (when ENMD bit of TMC10 register = 1)

TM10 and CM100 always compare their values and the INTCM100 interrupt is generated upon occurrence of a match. TM10 is cleared (0000H) at the count clock following the match.

Furthermore, when one more count clock is input, TM10 counts up to 0001H.

The interval time can be calculated with the following formula.

Interval time = (CM100 value + 1) × TM10 count clock rate

(b) Free-running operation (when ENMD bit of TMC10 register = 0)

TM10 performs a full count operation from 0000H to FFFFH, and after the TM1OVF0 bit of the STATUS0 register is set (to 1), TM10 is cleared to 0000H at the next count clock and resumes counting.

The free-running cycle can be calculated by the following formula.

Free-running cycle = 65,536 × TM10 count clock rate

(c) Compare function

TM10 connects two compare register (CM100, CM101) channels and two capture/compare register (CC100, CC101) channels.

When the TM10 count value and the set value of one of the compare registers match, a match interrupt (INTCM100, INTCM101, INTCC100[Note], INTCC101[Note]) is output. Particularly in the case of interval operation, TM10 is cleared upon generation of the INTCM100 interrupt.

Note This match interrupt is generated when CC100 and CC101 are set to the compare register mode.

(d) Capture function

TM10 connects two capture/compare register (CC100, CC101) channels.

When CC100 and CC101 are set to the capture register mode, the value of TM10 is captured in synchronization with the corresponding capture trigger signal.

Furthermore, an interrupt request signal (INTCC100, INTCC101) is generated by the valid edge of the INTP100, INTP101 input signals specified as the capture trigger signals.

Table 9-7. Capture Trigger Signal (TM10) to 16-Bit Capture Register

Capture Register	Capture Trigger Signal
CC100	INTP100
CC101	INTP100 or INTP101

Remark CC100 and CC101 are capture/compare registers. Which of these registers is used is specified by capture/compare control register 0 (CCR0).

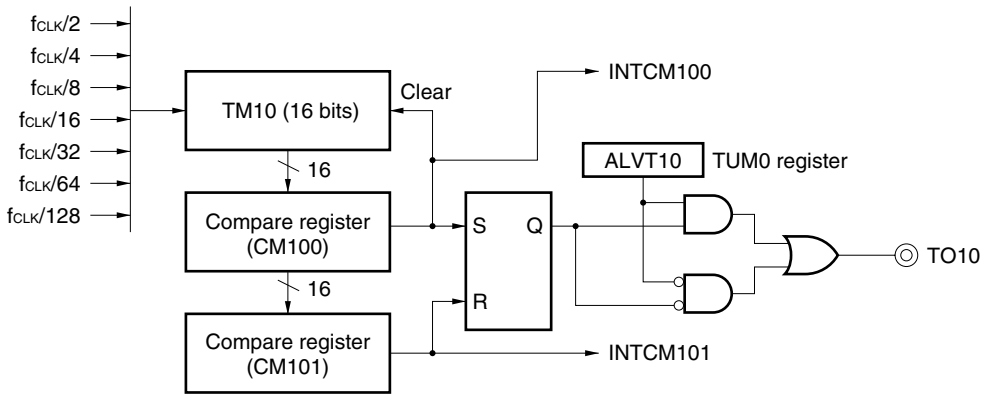
The valid edge of the capture trigger is specified by signal edge selection register 10 (SESA10). If both the rising edge and the falling edge are selected as the capture triggers, it is possible to measure the input pulse width externally. If a single edge is selected as the capture trigger, the input pulse cycle can be measured.

(e) PWM output operation

PWM output operation is performed from the TO10 pin by setting TM10 to the general-purpose timer mode (CMD bit = 0) using timer unit mode register 0 (TUM0).

The resolution is 16 bits, and the count clock can be selected from among seven internal clocks ($f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/16$, $f_{CLK}/32$, $f_{CLK}/64$, $f_{CLK}/128$).

Figure 9-49. TM10 Block Diagram (During PWM Output Operation)



Caution Be sure to set the count clock of TM10 to 10 MHz or lower.

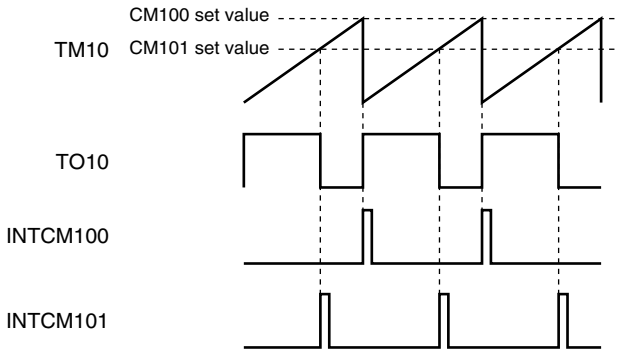
Remark f_{CLK}: Base clock

(i) Description of operation

The CM100 register is a compare register used to set the PWM output cycle. When the value of this register matches the value of TM10, the INTCM100 interrupt is generated. The compare match is saved by hardware, and TM10 is cleared at the next count clock after the match.

The CM101 register is a compare register used to set the PWM output duty. Set the duty required for the PWM cycle.

Figure 9-50. PWM Signal Output Example (When ALVT10 Bit = 0 Is Set)



Cautions 1. Changing the values of the CM100 and CM101 registers is prohibited during TM10 operation (TM1CE0 bit of TMC10 register = 1).

2. Changing the value of the ALVT10 bit of the TUM0 register is prohibited during TM10 operation.

3. PWM signal output is performed from the second PWM cycle after the TM1CE0 bit is set (to 1).

(2) Operation in UDC mode

(a) Overview of operation in UDC mode

The count clock input to TM10 in the UDC mode (CMD bit of TUM0 register = 1) can only be externally input from the TIUD10 and TCUD10 pins. Up/down count judgment in the UDC mode is determined based on the phase difference of the TIUD10 and TCUD10 pin inputs according to the PRM10 register setting (there is a total of four choices).

Table 9-8. List of Count Operations in UDC Mode

PRM10 Register			Operation Mode	TM10 Operation
PRM12	PRM11	PRM10		
1	0	0	Mode 1	Down count when TCUD10 = high level Up count when TCUD10 = low level
1	0	1	Mode 2	Up count upon detection of valid edge of TIUD10 input Down count upon detection of valid edge of TCUD10 input
★ 1	1	0	Mode 3	Up count upon detection of valid edge of TIUD10 input when TCUD10 = high level Down count upon detection of valid edge of TIUD10 input when TCUD10 = low level
1	1	1	Mode 4	Automatic judgment upon detection of both edges of TIUD10 input and both edges of TCUD10 input

The UDC mode is further divided into two modes according to the TM10 clear conditions (a count operation is performed only with TIUD10 and TCUD10 input in both modes).

- **UDC mode A (TUM0 register's CMD bit = 1, MSEL bit = 0)**
  The TM10 clear source can be selected as only external clear input (TCLR10), a match signal between the TM10 count value and the CM100 set value during up count operation, or the logical sum (OR) of the two signals, using bits CLR1 and CLR0 of the TMC10 register. TM10 can transfer the value of CM100 upon occurrence of a TM10 underflow.

- **UDC mode B (TUM0 register's CMD bit = 1, MSEL bit = 1)**
  The status of TM10 after a match of the TM10 count value and CM100 set value is as follows.
  <1> In the case of an up count operation, TM10 is cleared (0000H), and the INTCM100 interrupt is generated.
  <2> In the case of a down count operation, the TM10 count value is decremented (–1).

  The status of TM10 after a match of the TM10 count value and CM101 set value is as follows.
  <1> In the case of an up count operation, the TM10 count value is incremented (+1).
  <2> In the case of a down count operation, TM10 is cleared (0000H), and the INTCM101 interrupt is generated.

(b) Up/down count operation in UDC mode

TM10 up/down count judgment in the UDC mode is determined based on the phase difference of the TIUD10 and TCUD10 pin inputs according to the PRM10 register setting.

(i) Mode 1 (PRM10 register's PRM12 bit = 1, PRM11 bit = 0, PRM10 bit = 0)

In mode 1, the following count operations are performed based on the level of the TCUD10 pin upon detection of the valid edge of the TIUD10 pin.

- TM10 down count operation when TCUD10 pin = high level
- TM10 up count operation when TCUD10 pin = low level

Figure 9-51. Mode 1 (When Rising Edge Is Specified as Valid Edge of TIUD10 Pin)

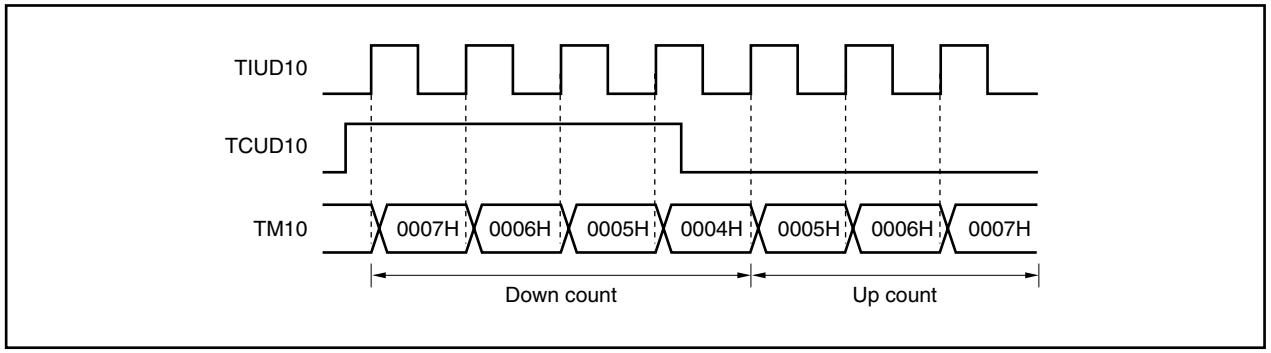
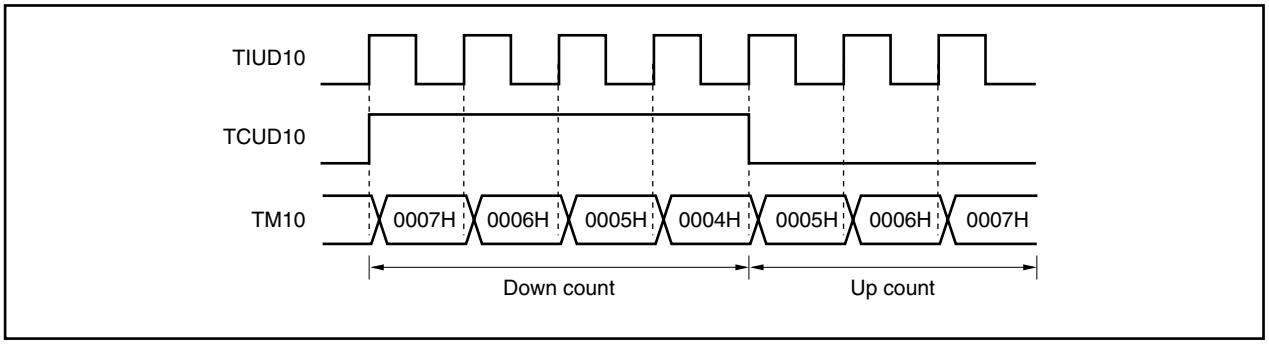


Figure 9-52. Mode 1 (When Rising Edge Is Specified as Valid Edge of TIUD10 Pin): In Case of Simultaneous TCUD10, TCUD10 Pin Edge Timing



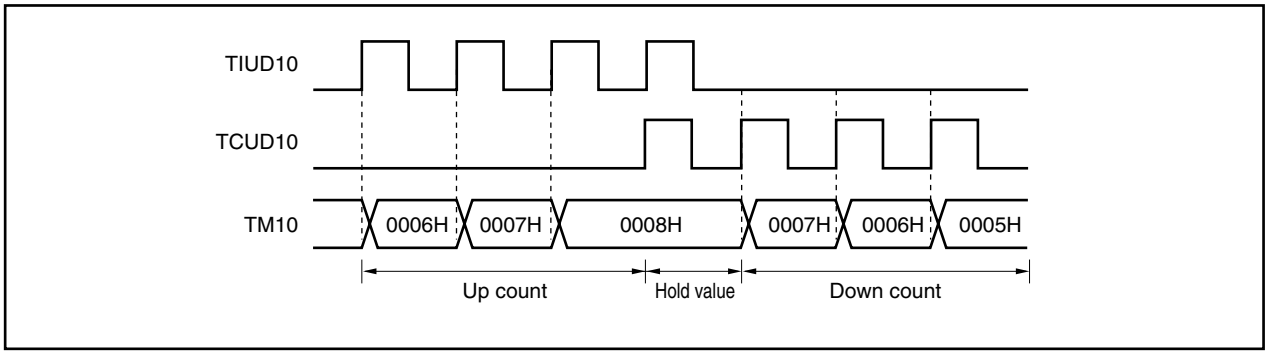
(ii) Mode 2 (PRM10 register's PRM12 bit = 1, PRM11 bit = 0, PRM10 bit = 1)

The count conditions in mode 2 are as follows.

- TM10 up count upon detection of valid edge of TIUD10 pin
- TM10 down count upon detection of valid edge of TCUD10 pin

Caution If the count clock is simultaneously input to the TIUD10 pin and the TCUD10 pin, count operation is not performed and the immediately preceding value is held.

Figure 9-53. Mode 2 (When Rising Edge Is Specified as Valid Edge of TIUD10, TCUD10 Pins)



(iii) Mode 3 (PRM10 register's PRM12 = 1, PRM11 = 1, PRM10 = 0)

In mode 3, when two signals 90 degrees out of phase are input to the TIUD10 and TCUD10 pins, the level of the TCUD10 pin is sampled at the input of the valid edge of the TIUD10 pin (Refer to **Figure 9-54**).

If the TCUD10 pin level sampled at the valid edge input to the TIUD10 pin is low, TM10 counts down when the valid edge is input to the TIUD10 pin.

If the TCUD10 pin level sampled at the valid edge input to the TIUD10 pin is high, TM10 counts up when the valid edge is input to the TIUD10 pin.

Figure 9-54. Mode 3 (When Rising Edge Is Specified as Valid Edge of TIUD10 Pin)

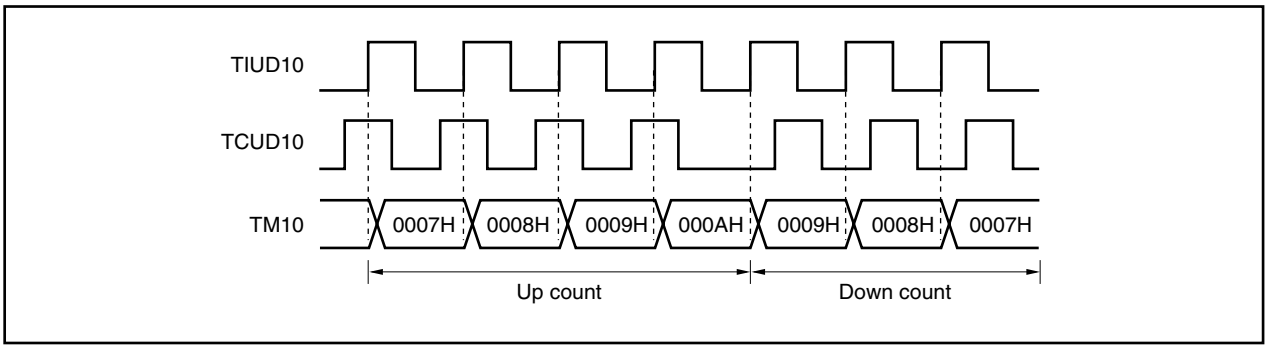
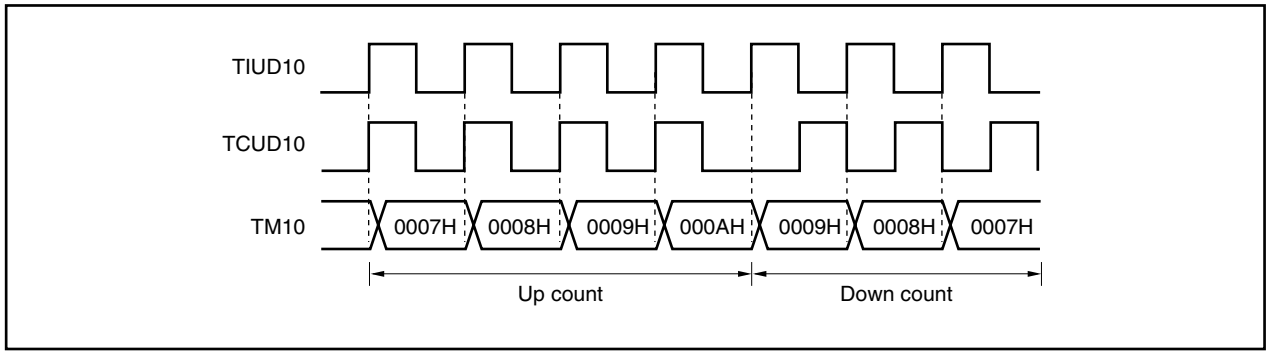


Figure 9-55. Mode 3 (When Rising Edge Is Specified as Valid Edge of TIUD10 Pin): In Case of Simultaneous TIUD10, TCUD10 Pin Edge Timing

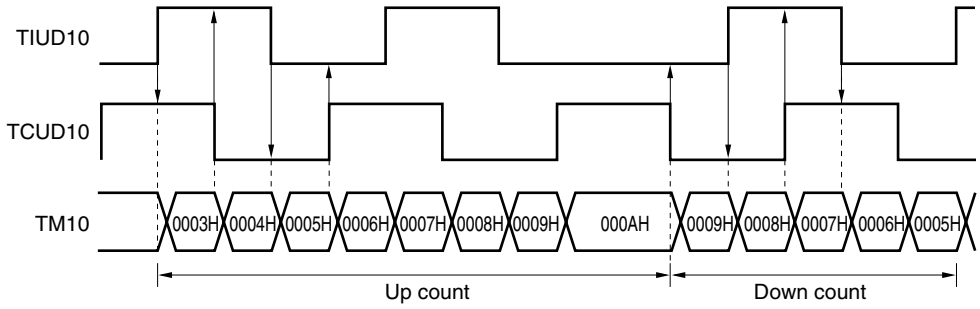


(iv) Mode 4 (PRM10 register's PRM12 = 1, PRM11 = 1, PRM10 = 1)

In mode 4, when two signals out of phase are input to the TIUD10 and TCUD10 pins, up/down operation is automatically judged and counting is performed according to the timing shown in **Figure 9-56**.

In mode 4, counting is executed at both the rising and falling edges of the two signals input to the TIUD10 and TCUD10 pins. Therefore, TM10 counts four times per cycle of an input signal (×4 count).

Figure 9-56. Mode 4



Cautions 1. When mode 4 is specified as the operation mode of TM10, the valid edge specifications for the TIUD10 and TCUD10 pins are not valid.

2. If the TIUD10 pin edge and TCUD10 pin edge are input simultaneously in mode 4, TM10 continues the same count operation (up or down) it was performing immediately before the input.

(c) Operation in UDC mode A

(i) Interval operation

The operations at the count clock following a match of the TM10 count value and the CM100 set value are as follows.

- In case of up count operation: TM10 is cleared (0000H) and the INTCM100 interrupt is generated.
- In case of down count operation: The TM10 count value is decremented (–1) and the INTCM100 interrupt is generated.

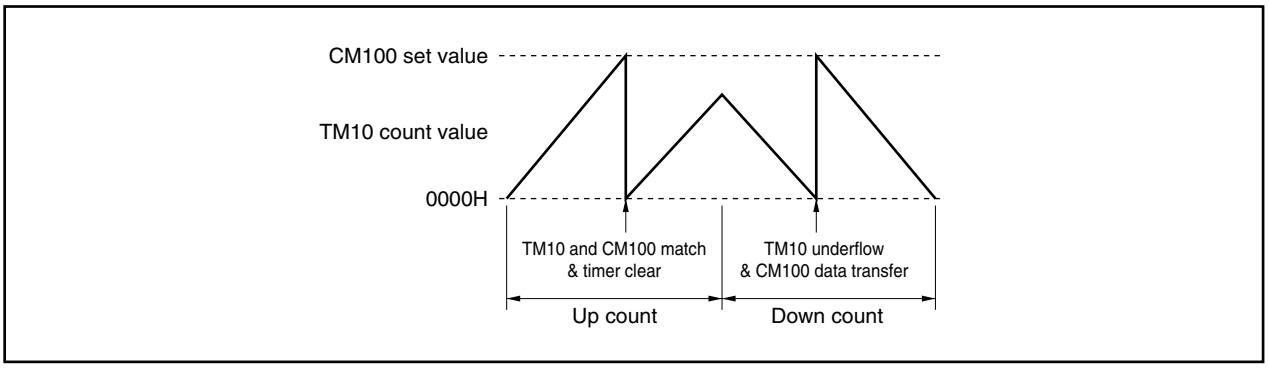
Remark The interval operation can be combined with the transfer operation.

(ii) Transfer operation

If TM10 becomes 0000H during down counting when the RLEN bit of the TMC10 register is 1, the CM100 register set value is transferred to TM10 at the next count clock.

Remarks 1. Transfer enable/disable can be set using the RLEN bit of the TMC10 register.
2. The transfer operation can be combined with the interval operation.

Figure 9-57. Example of TM10 Operation When Interval Operation and Transfer Operation Are Combined



(iii) Compare function

TM10 connects two compare register (CM100, CM101) channels and two capture/compare register (CC100, CC101) channels.

When the TM10 count value and the set value of one of the compare registers match, a match interrupt (INTCM100, INTCM101, INTCC100[Note], INTCC101[Note]) is output.

Note This match interrupt is generated when CC100 and CC101 are set to the compare register mode.

(iv) Capture function

TM10 connects two capture/compare register (CC100, CC101) channels.

When CC100 and CC101 are set to the capture register mode, the value of TM10 is captured in synchronization with the corresponding capture trigger signal. A capture interrupt (INTCC100, INTCC101) is generated upon detection of the valid edge.

(d) Operation in UDC mode B

(i) Basic operation

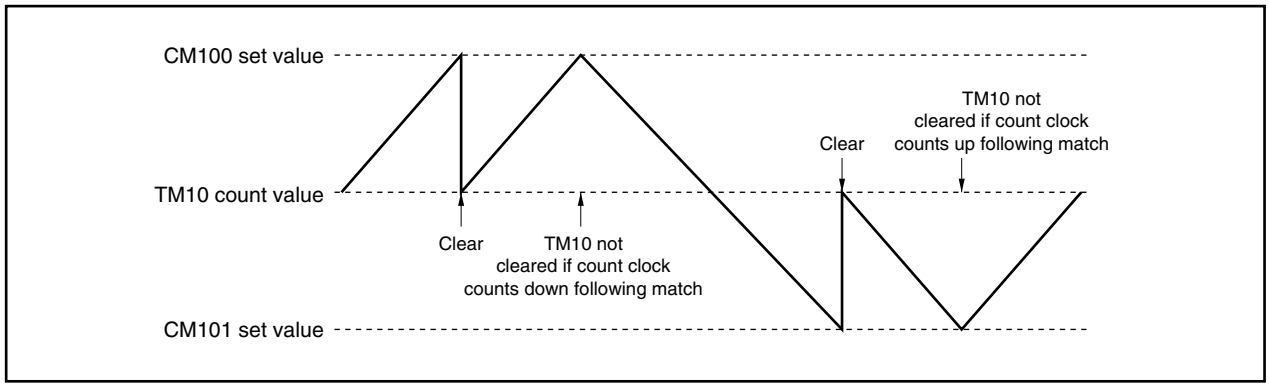
The operations at the next count clock after the count value of TM10 and the CM100 set value match when TM10 is in UDC mode B are as follows.

- In case of up count operation: TM10 is cleared (0000H) and the INTCM100 interrupt is generated.
- In case of down count operation: The TM10 count value is decremented (−1).

The operations at the next count clock after the count value of TM10 and the CM101 set value match when TM10 is in UDC mode B are as follows.

- In case of up count operation: The TM10 count value is incremented (+1).
- In case of down count operation: TM10 is cleared (0000H) and the INTCM101 interrupt is generated.

Figure 9-58. Example of TM10 Operation in UDC Mode



(ii) Compare function

TM10 connects two compare register (CM100, CM101) channels and two capture/compare register (CC100, CC101) channels.

When the TM10 count value and the set value of one of the compare registers match, a match interrupt (INTCM100 (only during up count operation), INTCM101 (only during down count operation), INTCC100[Note], INTCC101[Note]) is output.

Note This match interrupt is generated when CC100 and CC101 are set to the compare register mode.

(iii) Capture function

TM10 connects two capture/compare register (CC100, CC101) channels.

When CC100 and CC101 are set to the capture register mode, the value of TM10 is captured in synchronization with the corresponding capture trigger signal. A capture interrupt (INTCC100, INTCC101) is generated upon detection of the valid edge.

9.2.6 Supplementary description of internal operation

(1) Clearing of count value in UDC mode B

When TM10 is in UDC mode B, the conditions to clear the count value are as follows.

- In case of TM10 up-count operation: TM10 count value is cleared upon match with the CM100 register
- In case of TM10 down-count operation: TM10 count value is cleared upon match with the CM101 register

Figure 9-59. Clear Operation After Match of CM100 Register Set Value and TM10 Count Value

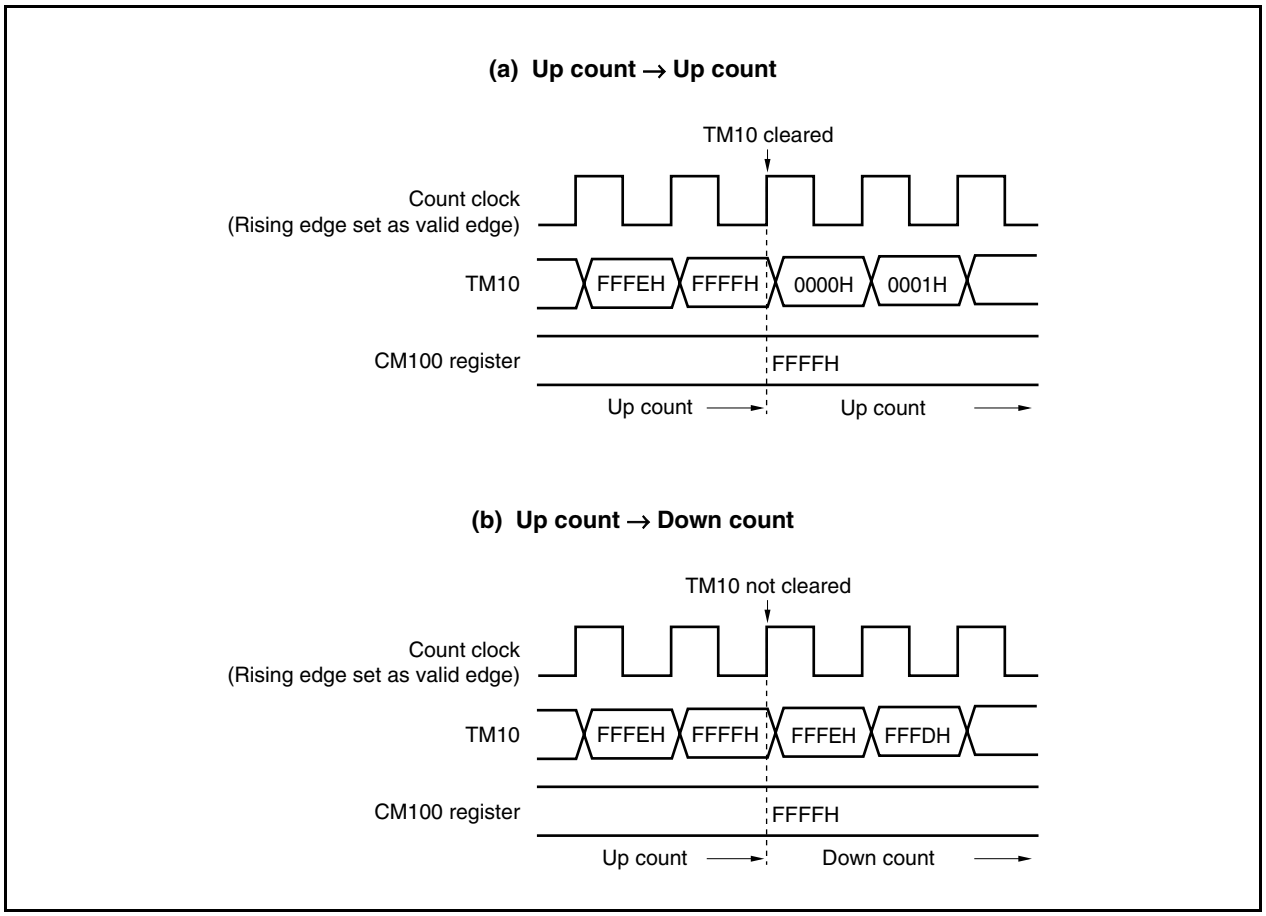
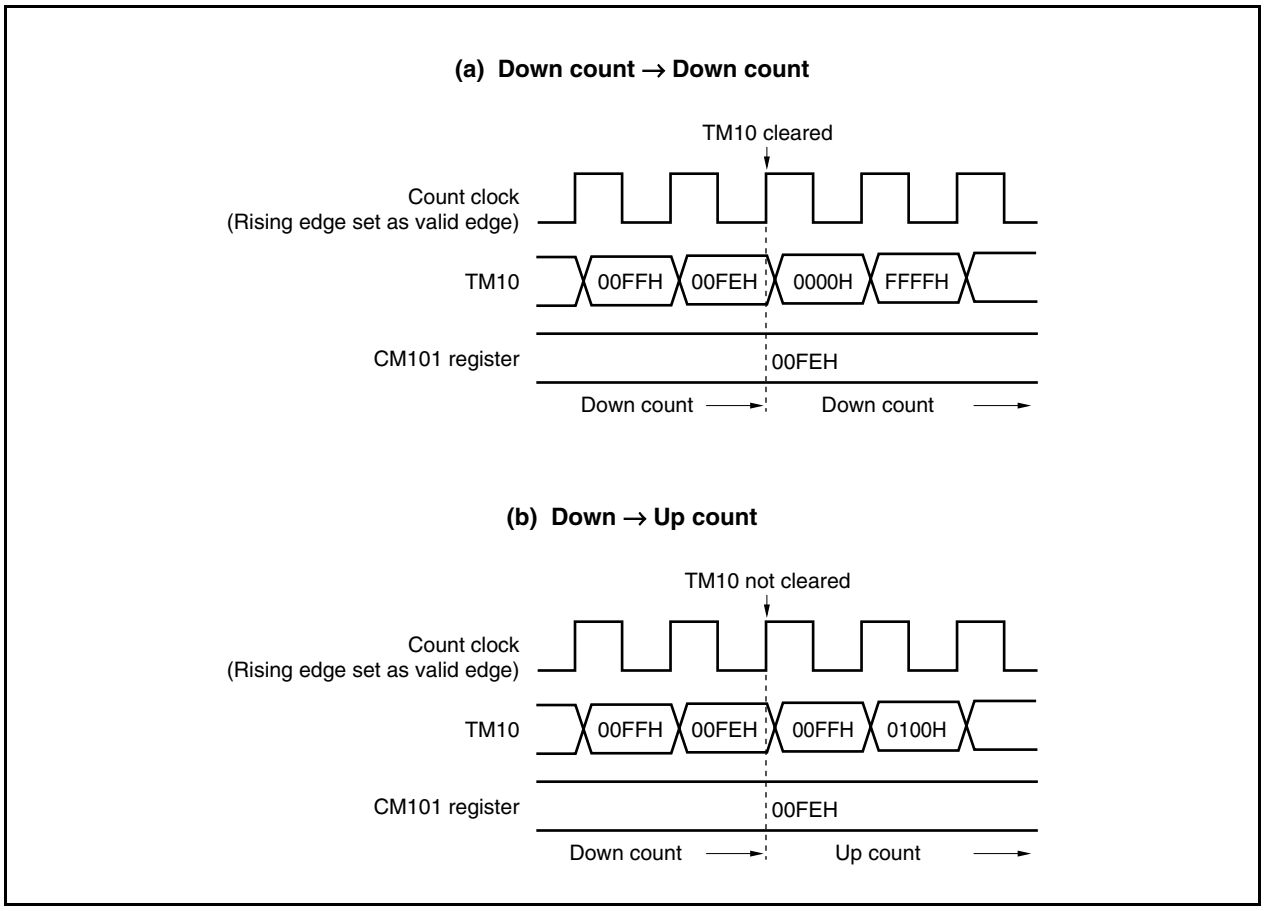


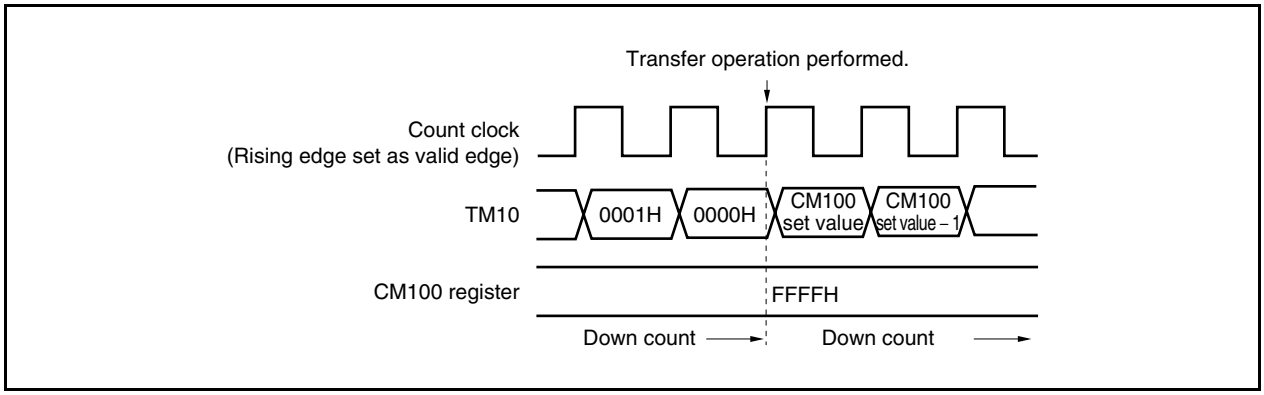
Figure 9-60. Clear Operation After Match of CM101 Register Set Value and TM10 Count Value



(2) Transfer operation

If TM10 becomes 0000H during down counting when the RLEN bit of the TMC10 register is 1 in UDC mode A, the set value of the CM100 register is transferred to TM10 at the next count clock. The transfer operation is not performed during up counting.

Figure 9-61. Internal Operation During Transfer Operation

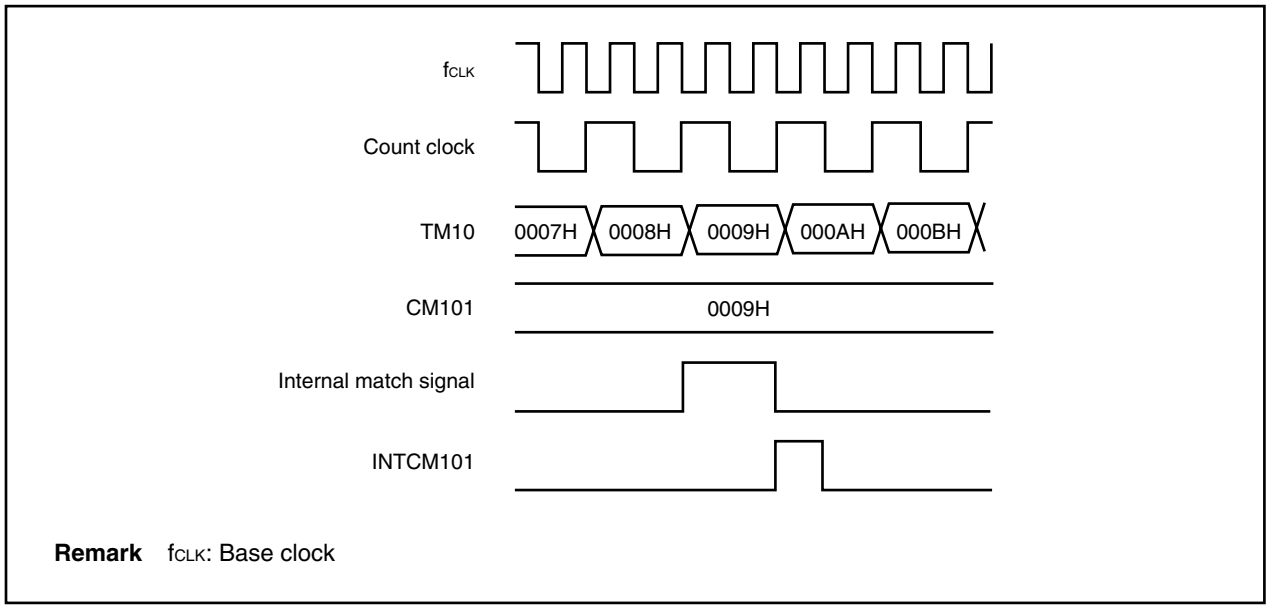


(3) Interrupt signal output upon compare match

An interrupt signal is output when the count value of TM10 matches the set value of the CM100, CM101, CC100[Note], or CC101[Note] register. The interrupt generation timing is as follows.

Note When CC100 and CC101 are set to the compare register mode.

Figure 9-62. Interrupt Output upon Compare Match
(CM101 with Operation Mode Set to General-Purpose Timer Mode and Count Clock Set to f_{CLK}/2)



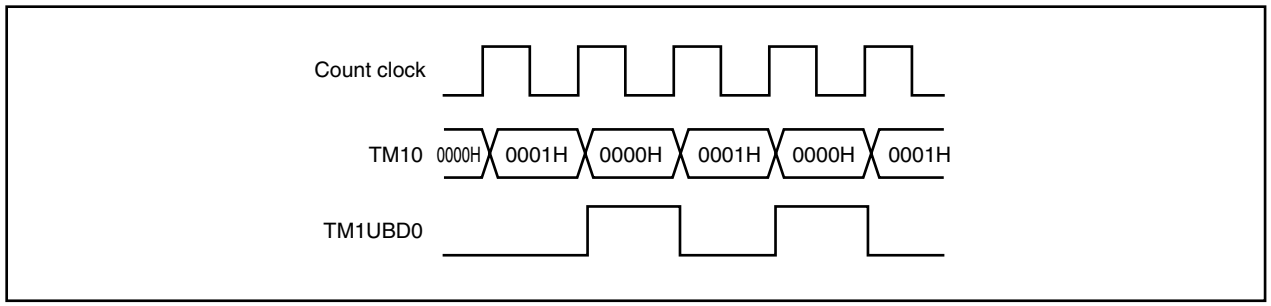
Remark f_{CLK}: Base clock

An interrupt signal such as the one illustrated in Figure 9-62 is output at the next count clock following a match of the TM10 count value and the set value of the corresponding compare register.

(4) TM1UBD0 flag (bit 0 of STATUS0 register) operation

In the UDC mode (CMD bit of TUM0 register = 1), the TM1UBD0 flag changes as follows during TM10 up/down count operation at every internal operation clock.

Figure 9-63. TM1UBD0 Flag Operation



9.3 Timer 2

9.3.1 Features (timer 2)

Timers 20 and 21 (TM20, TM21) are 16-bit general-purpose timer units that perform the following operations.

- Pulse interval or frequency measurement and programmable pulse output
- Interval timer
- PWM output timer
- 32-bit capture timer when 2 timer/counter channels are connected in cascade
  (In this case, four 32-bit capture register channels can be used.)

9.3.2 Function overview (timer 2)

- 16-bit timer/counter (TM20, TM21): 2 channels
- Bit length
  Timer 2 registers (TM20, TM21): 16 bits
  During cascade operation: 32 bits (higher 16 bits: TM21, lower 16 bits: TM20)
- Capture/compare register
  In 16-bit mode: 6
  In 32-bit mode: 4 (capture mode only)
- Count clock division selectable by prescaler (set the frequency of the count clock to 10 MHz or less)
- Base clock (f_{CLK}): 1 type (set f_{CLK} to 20 MHz or less)
  $f_{XX}/2$
- Prescaler division ratio
  The following division ratios can be selected according to the base clock (f_{CLK}).

Division Ratio	Base Clock (f_{CLK})
1/2	$f_{XX}/4$
1/4	$f_{XX}/8$
1/8	$f_{XX}/16$
1/16	$f_{XX}/32$
1/32	$f_{XX}/64$
1/64	$f_{XX}/128$
1/128	$f_{XX}/256$

- Interrupt request sources
  - Compare-match interrupt request: 6 types
    Perform comparison with subchannel n capture/compare register and generate the INTCC2n interrupt upon compare match.
  - Timer/counter overflow interrupt request: 2 types
    The INTTM20 (INTTM21) interrupt is generated when the count value of TM20 (TM21) becomes FFFFH.
- Capture request
  The count values of TM20 and TM21 can be latched using an external pin (INTP2n)[Notes 1, 2], TM10 interrupt signals (INTCM100, INTCM101) and interrupt requests by software as capture triggers.
- PWM output function
  Control of the output of the TO21 to TO24 pins in the compare mode and PWM output can be performed using the compare match timing of subchannels 1 to 4 and the zero count signal of the timer/counter.
- Timer count operation with external clock input[Note 2]
  Timer count operation can be performed using the pin TI2 clock input signal.
- Timer count enable operation[Note 3] with external pin input[Note 2]
  Timer count enable operation can be performed using the TCLR2 pin input signal.
- Timer/counter clear control[Notes 3, 4] with external pin input[Note 2]
  Timer/counter clear operation can be performed using the TCLR2 pin input signal.
- Up/down count control[Notes 3, 5] with external pin input[Note 2]
  Up/down count operation in the compare mode can be controlled using the TCLR2 pin input signal.
- Output delay operation
  A clock-synchronized output delay can be added to the output signal of the TO21 to TO24 pins.
  This is effective as an EMI countermeasure.
- Input filter
  An input filter can be inserted at the input stage of external pins (TI2, INTP20 to INTP25, TCLR2) and the TM10 interrupt signals (refer to **12.5.3 (1) Timer 2 input filter mode registers 0 to 5 (FEM0 to FEM5)**).

Notes 1. For the registers used to specify the valid edge for external interrupt requests (INTP20 to INTP25) to timer 2, refer to **7.3.8 (4) Timer 2 input filter mode registers 0 to 5 (FEM0 to FEM5)**.

2. The pairs TI2 and INTP20, TO21 and INTP21, TO22 and INTP22, TO23 and INTP23, TO24 and INTP24, TCLR2 and INTP25 are alternate function pins.

3. The count enable operation for the timer/counter via external pin input, timer/counter clear operation, and up/down count control cannot be performed all at the same time.

4. In the case of 32-bit cascade connection, a clear operation by external pin input (TCLR2) cannot be performed.

5. Up/down count control using 32-bit cascade connection cannot be performed.

Remark fxx: Internal system clock
n = 0 to 5

9.3.3 Basic configuration

The basic configuration is shown below.

Table 9-9. Timer 2 Configuration List

Timer	Count Clock	Register	Read/Write	Generated Interrupt Signal	Capture Trigger	Other Functions
Timer 2	fxx/4, fxx/8, fxx/16, fxx/32, fxx/64, fxx/128, fxx/256	TM20	–	INTTM20	–	**Note 1**
		TM21	–	INTTM21	–	**Note 1**
		CVSE00	Read/write	INTCC20	INTP20/INTP25	–
		CVSE10	Read/write	INTCC21	INTP21/INTP24	Buffer/**Note 2**
		CVSE20	Read/write	INTCC22	INTP22/INTP23	Buffer/**Note 2**
		CVSE30	Read/write	INTCC23	INTP23/INTP22	Buffer/**Note 2**
		CVSE40	Read/write	INTCC24	INTP24/INTP21	Buffer/**Note 2**
		CVSE50	Read/write	INTCC25	INTP25/INTP20	–
		CVPE40	Read	INTCC24	INTP24/INTP21	**Note 2**
		CVPE30	Read	INTCC23	INTP23/INTP22	**Note 2**
		CVPE20	Read	INTCC22	INTP22/INTP23	**Note 2**
		CVPE10	Read	INTCC21	INTP21/INTP24	**Note 2**

Notes 1. Cascade operation with TM20 and TM21 is possible.

2. Cascade operation using the CVSEn0 and CVPEn0 registers is possible (n = 1 to 4).

Remark fxx: Internal system clock

The following shows the capture/compare operation sources.

Table 9-10. Capture/Compare Operation Sources

Register	Subchannel No.	Timer to Be Captured	Timer to Be Compared	Timer Captured in 32-Bit Cascade Connection
CVSE00	0	TM20	TM20	–
CVPEn0	n	TM21 when BFEEy bit of CMSEm0 register = 0	TM20 when TB1Ey, TB0Ey bits of CMSEm0 register = 01	TM21
CVSEn0	n	TM20 when BFEEy bit of CMSEm0 register = 0	Used as buffer	TM20
CVSE50	5	TM21	TM21	–

Remark n = 1 to 4
m: m = 12 when n = 1, 2, m = 34 when n = 3, 4
y: y = 1, 2 when m = 12, y = 3, 4 when m = 34

The following shows the output level sources during timer output.

Table 9-11. Output Level Sources During Timer Output

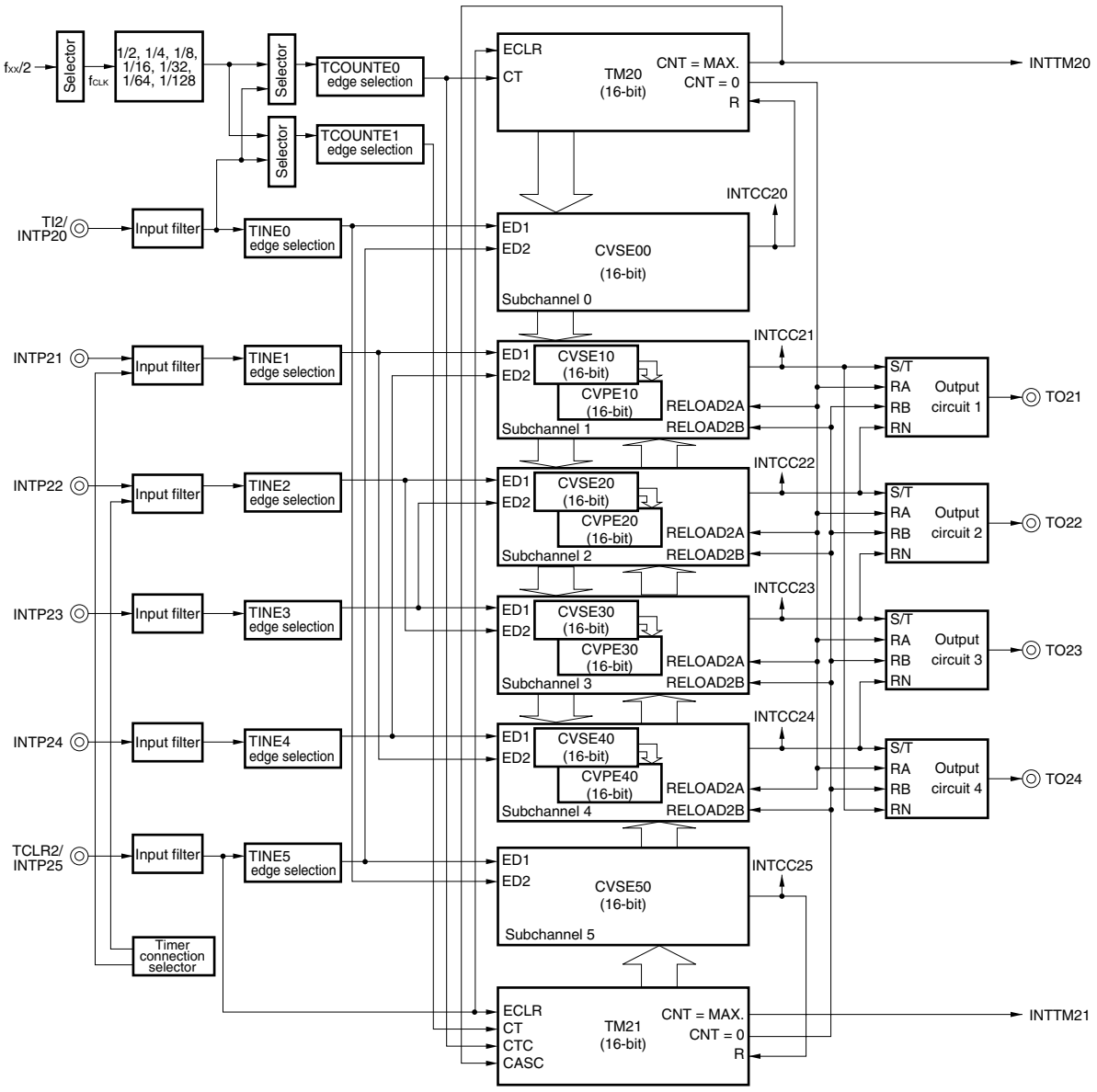
TO2n	Toggle Mode 0 (OTMEn1, OTMEn0 = 00)		Toggle Mode 1 (OTMEn1, OTMEn0 = 01)		Toggle Mode 2 (OTMEn1, OTMEn0 = 10)		Toggle Mode 3 (OTMEn1, OTMEn0 = 11)	
Trigger	Compare match of sub-channel n		Compare match of sub-channel n	TM20 = 0	Compare match of sub-channel n	TM21 = 0	Compare match of sub-channel n	Compare match of sub-channel n + 1
Output level	Active output	Inactive output	Active output	Inactive output	Active output	Inactive output	Active output	Inactive output

Remarks 1. n = 1 to 4

2. OTMEn1, OTMEn0: Bits 13, 12, 9, 8, 5, 4, 1, and 0 of timer 2 output control register 0 (OCTLE0)

Figure 9-64 shows the block diagram of timer 2.

Figure 9-64. Block Diagram of Timer 2



Remark fXX: Internal system clock
fCLK: Base clock (20 MHz (MAX.))

Table 9-12. Meaning of Signals in Block Diagram

Signal Name	Meaning
CASC[Note 1]	TM21 count signal input in 32-bit mode
CNT	Count value of timer 2 (CNT = MAX.: Maximum value count signal output of timer 2 (generated when TM2n = FFFFH), CNT = 0: Zero count signal output of timer 2 (generated when TM2n = 0000H))
CT	TM2n count signal input in 16-bit mode
CTC	TM21 count signal input in 32-bit mode
ECLR	External control signal input from TCLR2 input
ED1, ED2	Capture event signal input from edge selector
R[Note 2]	Compare match signal input (subchannel 0/5)
RA	TM20 zero count signal input (reset signal of output circuit)
RB	TM21 zero count signal input (reset signal of output circuit)
RELOAD2A	TM20 zero count signal input (generated when TM20 = 0000H)
RELOAD2B	TM21 zero count signal input (generated when TM21 = 0000H)
RN	Subchannel x interrupt signal input (reset signal of output circuit)
S/T	Subchannel x interrupt signal input (set signal of output circuit)
TCOUNTE0, TCOUNTE1	Timer 2 count enable signal input
TINEm	Timer 2 subchannel m capture event signal input

Notes 1. TM21 performs a count operation when CASC (CNT = MAX. for TM20) is generated and the rising edge of CTC is detected in the 32-bit mode.

2. TM20/TM21 clear by subchannel 0/5 compare match or count direction can be controlled.

Remark m = 0 to 5
n = 0, 1
x = 1 to 4

(1) Timers 20, 21 (TM20, TM21)

The features of TM2n are listed below.

- Free-running counter that enables counter clearing by compare match of subchannel 0 and subchannel 5
- Can be used as a 32-bit capture timer when TM20 and TM21 are connected in cascade.
- Up/down control, counter clear, and count operation enable/disable can be controlled by external pin (TCLR2)
- Counter up/down and clear operation control method can be set by software.
- Stop upon occurrence of count value 0 and count operation start/stop can be controlled by software.

(2) Timer 2 subchannel 0 capture/compare register (CVSE00)

The CVSE00 register is the 16-bit capture/compare register of subchannel 0.

In the capture register mode, it captures the TM20 count value.

In the compare register mode, it detects a match with TM20.

This register can be read/written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CVSE00																	FFFFF660H	0000H

(3) Timer 2 subchannel n main capture/compare register (CVPEn0) (n = 1 to 4)

The CVPEn0 register is the subchannel n 16-bit main capture/compare register.

In the capture register mode, this register captures the value of TM21 when the BFEEn bit of the CMSEm0 register = 0 (m = 12, 34). When the BFEEn bit = 1, this register holds the value of TM20 or TM21.

In compare register mode, a match between this register and TM2x is detected (TM2x = timer/counter selected by TB1En and TB0En bits).

If the capture register mode is selected in the 32-bit mode (value of TB1En, TB0En bits of CMSEm0 register = 11B), this register captures the contents of TM21 (higher 16 bits).

This register is read-only, in 16-bit units.

Caution When the BFEEn bit = 1, a compare match occurs on starting the timer in the compare register mode because the values of both the TM2x and CVPEn0 registers are 0 after reset (TM2x = timer/counter selected by TB1En and TB0En bits (n = 1 to 4)). After that, the value of the sub register (CVSEn0) is written to the main register (CVPEn0).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CVPE10																	FFFFF652H	0000H
CVPE20																	FFFFF656H	0000H
CVPE30																	FFFFF65AH	0000H
CVPE40																	FFFFF65EH	0000H

(4) Timer 2 subchannel n sub capture/compare register (CVSEn0) (n = 1 to 4)

The CVSEn0 register is the subchannel n 16-bit sub capture/compare register.

In the compare register mode, this register can be used as a buffer. In the capture register mode, this register captures the value of TM20 when the BFEEn bit of the CMSEm0 register = 0 (m = 12, 34).

If the capture register mode is selected in the 32-bit mode (value of TB1En and TB0En bits of CMSEm0 register = 11B), this register captures the contents of TM20 (lower 16 bits).

The CVSEn0 register can be written only in the compare register mode. If this register is written in the capture register mode, the contents written to CVSEn0 register will be lost.

This register can be read/written in 16-bit units.

Caution When the BFEEn bit = 1, a compare match occurs on starting the timer in the compare register mode because the values of both the TM2x and CVPEn0 registers are 0 after reset (TM2x = timer/counter selected by TB1En and TB0En bits (n = 1 to 4)). After that, the value of the sub register (CVSEn0) is written to the main register (CVPEn0).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CVSE10																	FFFFF650H	0000H
CVSE20																	FFFFF654H	0000H
CVSE30																	FFFFF658H	0000H
CVSE40																	FFFFF65CH	0000H

(5) Timer 2 subchannel 5 capture/compare register (CVSE50)

The CVSE50 register is the 16-bit capture/compare register of subchannel 5.

In the capture register mode, it captures the count value of TM21.

In the compare register mode, it detects a match with TM21.

This register can be read/written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CVSE50																	FFFFF662H	0000H

9.3.4 Control registers

(1) Timer 1/timer 2 clock selection register (PRM02)

The PRM02 register is used to select the base clock (f_{CLK}) of timer 1 and timer 2.

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

Cautions 1. Always set this register to 01H before using timer 1 and timer 2. Setting of other than 01H is prohibited.

2. Set f_{CLK} to 20 MHz or less.

	7	6	5	4	3	2	1	0	Address	After reset
PRM02	0	0	0	0	0	0	0	PRM2	FFFFF5D8H	00H

Bit position	Bit name	Function
0	PRM2	Specifies the base clock (f_{CLK}) of timer 1 and timer 2. 1: $f_{CLK} = f_{XX}/2$

Remark f_{XX}: Internal system clock

(2) Timer 2 clock stop register 0 (STOPTE0)

The STOPTE0 register is used to stop the operation clock input to timer 2.

This register can be read/written in 16-bit units.

When the higher 8 bits of the STOPTE0 register are used as the STOPTE0H register, and the lower 8 bits are used as the STOPTE0L register, the STOPTE0H register can be read/written in 8-bit or 1-bit units, and the STOPTE0L register is read-only, in 8-bit units.

Cautions 1. Initialize timer 2 when the STFTE bit = 0. Timer 2 cannot be initialized when the STFTE bit = 1.

2. If, following initialization, the value of the STFTE bit is made "1", the initialized state is maintained.

	<15>	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
STOPTE0	STFTE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FFFFF640H	0000H

Bit position	Bit name	Function
15	STFTE	Stops the operation clock to timer 2. 0: Normal operation 1: Stop operation clock to timer 2

(3) Timer 2 count clock/control edge selection register 0 (CSE0)

The CSE0 register is used to specify the TM2n count clock and the control valid edge (n = 0, 1).

This register can be read/written in 16-bit units.

When the higher 8 bits of the CSE0 register are used as the CSE0H register, and the lower 8 bits are used as the CSE0L register, they can be read/written in 8-bit or 1-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CSE0	0	0	0	0	TES1E1	TES1E0	TES0E1	TES0E0	CESE1	CESE0	CSE12	CSE11	CSE10	CSE02	CSE01	CSE00	FFFFF642H	0000H

Bit position	Bit name	Function
11, 10, 9, 8	TESnE1, TESnE0	Specifies the valid edge of the TM2n internal count clock (TCOUNTEn) signal.
7, 6	CESE1, CESE0	Specifies the valid edge of the TM2n external clear input (TCLR2).
5 to 3, 2 to 0	CSEn2, CSEn1, CSEn0	Selects internal count clock (TCOUNTEn) of TM2n.

TESnE1	TESnE0	Valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges[Note]

CESE1	CESE0	Valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Through input (no clear operation)
1	1	Both rising and falling edges

CSEn2	CSEn1	CSEn0	Count clock
0	0	0	$f_{CLK}/2$[Note]
0	0	1	$f_{CLK}/4$
0	1	0	$f_{CLK}/8$
0	1	1	$f_{CLK}/16$
1	0	0	$f_{CLK}/32$
1	0	1	$f_{CLK}/64$
1	1	0	$f_{CLK}/128$
1	1	1	Selects input signal from external clock input pin (TI2) as clock.

Note Setting TESnE1, TESnE0 = 11B and CSEn2 to CSEn0 = 000B at the same time is prohibited.

Remark n = 0, 1

f_{CLK}: Base clock

(4) Timer 2 subchannel input event edge selection register 0 (SESE0)

The SESE0 register specifies the valid edge of the external capture signal input (TINEn) for the subchannel n capture/compare register performing capture (n = 0 to 5).

This register can be read/written in 16-bit units.

When the higher 8 bits of the SESE0 register are used as the SESE0H register, and the lower 8 bits are used as the SESE0L register, they can be read/written in 8-bit or 1-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
SESE0	0	0	0	0	IESE51	IESE50	IESE41	IESE40	IESE31	IESE30	IESE21	IESE20	IESE11	IESE10	IESE01	IESE00	FFFFF644H	0000H

Bit position	Bit name	Function
11 to 0	IESEn1, IESEn0	Specifies the valid edge of external capture signal input (TINEn) for subchannel n capture/compare register performing capture.

IESEn1	IESEn0	Valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Remark n = 0 to 5

(5) Timer 2 time base control register 0 (TCRE0)

The TCRE0 register controls the operation of TM2n (n = 0, 1).

This register can be read/written in 16-bit units.

When the higher 8 bits of the TCRE0 register are used as the TCRE0H register, and the lower 8 bits are used as the TCRE0L register, they can be read/written in 8-bit or 1-bit units.

Cautions 1. If ECREn = 1 and ECEEn = 1 have been set, it is not possible to input an external clear signal (TCLR2) for TM2n. In this case, first set CLREn = 1, and then clear TM2n by software (n = 0, 1).

2. When clearing is performed using the ECLR signal, the TM2n counter is cleared with a delay of (1 internal count clock set with bits CSEn2 to CSEn0 of the CSE0 register) + 2 base clocks. Therefore, if external clock input is selected as the internal count clock, the counter is not cleared until the external clock (TI2) is input.

3. The ECREn bit and the ECEEn bit cannot be set to 1.

4. If the ECEEn bit is set to 1 and the ECREn bit is set to 0, a down count operation cannot be performed.

5. When UDSEn1, UDSEn0 = 01 and OSTEn = 1, the counter does not count up when the counter value is 0. Therefore, when the counter value is 0, set OSTEn = 0, and after the value of the counter ceases to be 0, set OSTEn = 1. Also, on the application, change the value of OSTEn from 0 to 1 using the subchannels 0 and 5 interrupt signals.

6. When the TM2n count value is cleared (0) by setting CLREn to 1, the CLREn = 1 setting must be held for at least one of the internal count clocks set by the CSEn2 to CSEn0 bits of the CSE0 register.

Example When timer 20 (TM20) is cleared (0)

<1> Select f_{CLK}/2 as TM20 internal count clock

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSE0	0	0	0	0	×	×	×	×	×	×	×	×	×	0	0	0

<2> Clear (0) the TM20 count value

	7	6	5	4	3	2	1	0
TCRE0L	0	1	0	0	0	×	×	×

<3> Set the conditions required for the TM20 count clock

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSE0	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×

<4> Start the TM20 count operation

	7	6	5	4	3	2	1	0
TCRE0L	0	0	1	0	0	×	×	×

(1/2)

	15	<14>	<13>	12	11	10	9	8	7	<6>	<5>	4	3	2	1	0	Address	After reset
TCRE0	CASE1	CLRE1	CEE1	ECRE1	ECEE1	OSTE1	UDSE11	UDSE10	0	CLRE0	CEE0	ECRE0	ECEE0	OSTE0	UDSE01	UDSE00	FFFFF646H	0000H

Bit position	Bit name	Function
15	CASE1	Specifies 32-bit cascade operation mode for TM21 (TM21 counts upon overflow of TM20 (carry count)). 0: Not connected in cascade[Note 1] 1: 32-bit cascade operation mode[Notes 2, 3] **Notes 1.** TM21 counts at CT signal input in the count enabled state. **2.** TM21 counts at CTC and CASC signal inputs in the count enabled state. **3.** Only the capture register mode can be used for the capture/compare register. **Cautions 1. When CASE1 = 1, set the TByE1 and TByE0 bits of the CMSEx0 register to 11 (x = 12, 34, y: When x = 12, y = 1, 2, and when x = 34, y = 3, 4).** **2. When CASE1 = 0, TCOUNTE1 is selected as the count of TM21. When CASE1 = 1, TCOUNTE0 and the TM20 overflow signal are selected as the count of TM21.**
14, 6	CLREn	Specifies software clear for TM2n. 0: TM2n operation continued 1: TM2n count value cleared (0) **Caution Do not perform the software clear and hardware clear operations simultaneously.**
13, 5	CEEn	Specifies TM2n count operation enable/disable. 0: Count operation stopped 1: Count operation enabled
12, 4	ECREn	Specifies TM2n external clear (TCLR2) operation enable/disable via ECLR signal input. 0: TM2n external clear (TCLR2) operation not enabled 1: TM2n external clear (TCLR2) operation enabled **Cautions 1. In the 32-bit cascade operation mode (CASE1 = 1), the TM2n external clear operation is not performed.** **2. When the count value is cleared by inputting the ECLR signal while ECREn = 1, the ECREn = 1 setting must be held for at least one of the internal count clocks set by the CSEn2 to CSEn0 bits of the CSE0 register.** **3. In the 32-bit cascade operation mode (CASE1 = 1), only TM21 is affected by the ECREn bit setting.**

Remark n = 0, 1

(2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>11, 3</td><td>ECEEn</td><td>Specifies TM2n count operation enable/disable through ECLR signal input.
0: TM2n count operation not enabled
1: TM2n count operation enabled

Cautions 1. In the 32-bit cascade operation mode (CASE1 = 1), the TM2n count operation using ECLR signal input is not performed.
2. When the ECEEn bit = 1, always set the CESE1 and CESE0 bits of the CSE0 register to 10 (through input).
3. In the 32-bit cascade operation mode (CASE1 = 1), only TM21 is affected by the ECEEn bit setting.</td></tr>
<tr><td>10, 2</td><td>OSTEn</td><td>Specifies stop mode.
0: TM2n count stopped when count value is 0.
1: TM2n count not stopped when count value is 0.

Caution When the TM2n count stop is cancelled when the OSTE1n bit = 1 (TM2n count is stopped when the count value is 0), TM2n counts up except when the UDSEn1, UDSEn0 bits = 10. The count direction when the UDSEn1 and UDSEn0 bits = 10 is determined by the value of ECLR.</td></tr>
<tr><td>9, 8, 1, 0</td><td>UDSEn1,
UDSEn0</td><td>Specifies TM2n up/down count.
<table><tr><th>UDSEn1</th><th>UDSEn0</th><th>Count</th></tr><tr><td>0</td><td>0</td><td>Perform only up count.
Clear TM2n with compare match signal.</td></tr><tr><td>0</td><td>1</td><td>Count up after TM2n has become 0, and count down after a compare match occurs for subchannels 0, 5 (triangular wave up/down count).</td></tr><tr><td>1</td><td>0</td><td>Selects up/down count according to the ECLR signal input.
Up count when ECLR = 1
Down count when ECLR = 0</td></tr><tr><td>1</td><td>1</td><td>Setting prohibited</td></tr></table>Cautions 1. In the 32-bit cascade operation mode (CASE1 bit = 1), set the UDSEn1 and UDSEn0 bits to 00.
2. When the UDSEn1 and UDSEn0 bits = 10, be sure to set the CESE1 and CESE0 bits of the CSE0 register to 10 (through input).
3. When the UDSEn1 and UDSEn0 bits = 10, compare match between TM2n and CVSEx0 has no effect on the TM2n count operation (x: 0 when n = 0, 5 when n = 1).</td></tr>
</table>

Remark n = 0, 1

(6) Timer 2 output control register 0 (OCTLE0)

The OCTLE0 register controls timer output from the TO2n pin (n = 1 to 4).

This register can be read/written in 16-bit units.

When the higher 8 bits of the OCTLE0 register are used as a OCTLE0H register, and the lower 8 bits are used as a OCTLE0L register, they can be read/written in 8-bit or 1-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
OCTLE0	SWFE 4	ALVE 4	OTME 41	OTME 40	SWFE 3	ALVE 3	OTME 31	OTME 30	SWFE 2	ALVE 2	OTME 21	OTME 20	SWFE 1	ALVE 1	OTME 11	OTME 10	FFFFF648H	0000H

Bit position	Bit name	Function
15, 11, 7, 3	SWFEn	Fixes the TO2n pin output level according to the setting of ALVEn bit. 0: Output level not fixed. 1: When ALVEn = 0, output level fixed to low level. When ALVEn = 1, output level fixed to high level.
14, 10, 6, 2	ALVEn	Specifies the active level of the TO2n pin output. 0: Active level is high level 1: Active level is low level
13, 12, 9, 8, 5, 4, 1, 0	OTMEn1, OTMEn0	Specifies toggle mode.

OTMEn1	OTMEn0	Toggle mode
0	0	Toggle mode 0: Reverse output level of TO2n output every time a subchannel n compare match occurs.
0	1	Toggle mode 1: Upon subchannel n compare match, set TO2n output to active level, and when TM20 is “0”, set TO2n output to inactive level.
1	0	Toggle mode 2: Upon subchannel n compare match, set TO2n output to active level, and when TM21 is “0”, set TO2n output to inactive level.
1	1	Toggle mode 3: Upon subchannel n compare match, set TO2n output to active level, and upon subchannel n + 1 compare match, set TO2n output to inactive level (when n = “4”, n + 1 becomes “1”).

Cautions 1. When the OTMEn1 and OTMEn0 bits = 11 (toggle mode 3), if the same output delay operation settings are made when setting the ODLEn2 to ODLEn0 bits of the ODELE0 register, two outputs change simultaneously upon 1 subchannel n compare match.

2. If two or more signals are input simultaneously to the same output circuit, S/T signal input has a higher priority than RA, RB, and RN signal inputs.

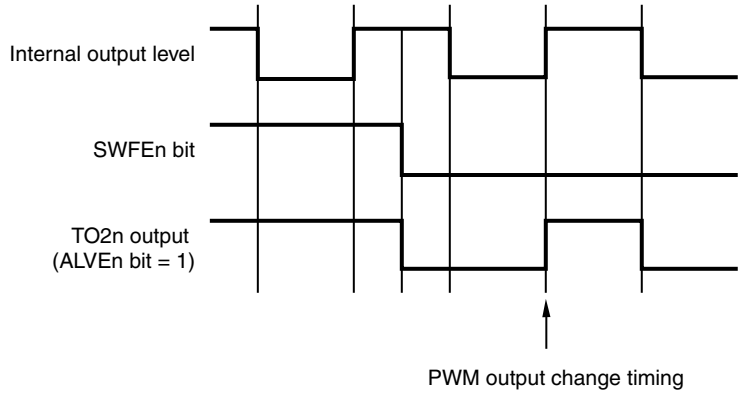
Remark n = 1 to 4

★ **(a) Caution for PWM output change timing**

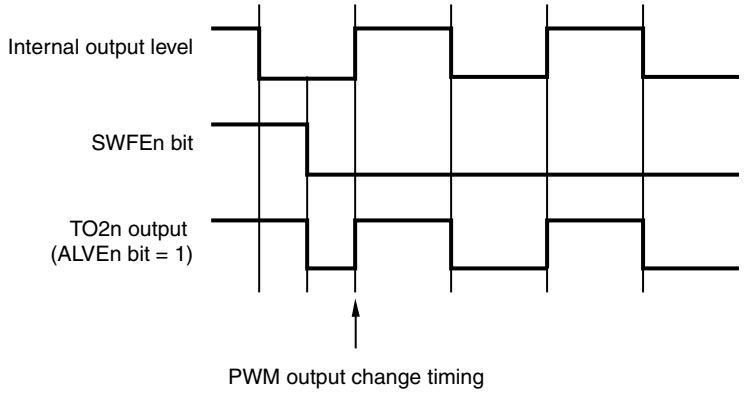
If the SWFEn bit is changed from 1 to 0 when the timer is operating while the internal PWM output operation is being performed, then the output level becomes active. After that, PWM output from the TO2n pin is performed upon a compare match at subchannel n. However, the first PWM output change timing varies as follows, depending on the internal output level and the SWFEn bit clear timing.

Figure 9-65. PWM Output Change Timing

(i) Example 1



(ii) Example 2



Remark n = 1 to 4

(7) Timer 2 subchannel 0, 5 capture/compare control register (CMSE050)

The CMSE050 register controls the timer 2 subchannel 0 capture/compare register (CVSE00) and the timer 2 subchannel 5 capture/compare register (CVSE50).

This register can be read/written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CMSE050	0	0	EEVE5	0	LNKE5	CCSE5	0	0	0	0	EEVE0	0	LNKE0	CCSE0	0	0	FFFFF64AH	0000H

Bit position	Bit name	Function
13, 5	EEVEn	Enables/disables event detection by subchannel n capture/compare register. 0: ED1 and ED2 signal inputs ignored (nothing is done even if these signals are input). 1: Operation caused by ED1 and ED2 signal inputs enabled.
11, 3	LNKEn	Specifies capture event signal input from edge selection to ED1 or ED2. 0: In capture register mode, ED1 signal input selected. In compare register mode, LNKEn bit has no influence. 1: In capture register mode, ED2 signal input selected. In compare register mode, LNKEn bit has no influence.
10, 2	CCSEn	Selects capture/compare register operation mode. 0: Operates in capture register mode. The TM20 and TM21 count statuses can be read with subchannel 0 and subchannel 5, respectively. 1: Operates in compare register mode. TM2m is cleared upon detection of match between subchannel n and TM2m.

Remark m = 0, 1
n = 0, 5

(8) Timer 2 subchannel 1, 2 capture/compare control register (CMSE120)

The CMSE120 register controls the timer 2 subchannel n sub capture/compare register (CVSEn0) and the timer 2 subchannel n main capture/compare register (CVPEn0) (n = 1, 2).

This register can be read/written in 16-bit units.

(1/2)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CMSE120	0	0	EEVE2	BFEE2	LNKE2	CCSE2	TB1E2	TB0E2	0	0	EEVE1	BFEE1	LNKE1	CCSE1	TB1E1	TB0E1	FFFFF64CH	0000H

Bit position	Bit name	Function
13, 5	EEVEn	Enables/disables event detection for CMSE120 register. 0: ED1 and ED2 signal inputs ignored (nothing is done even if these signals are input). 1: Operation caused by ED1 and ED2 signal inputs enabled.
12, 4	BFEEn	Specifies the buffer operation of subchannel n sub capture/compare register (CVSEn0). 0: Subchannel n sub capture/compare register (CVSEn0) not used as buffer. 1: Subchannel n sub capture/compare register (CVSEn0) used as buffer. **Caution When the BFEEn bit = 1, a compare match occurs on starting the timer in the compare register mode because the values of both the TM2x and CVPEn0 registers are 0 after reset (TM2x = timer/counter selected by TB1En and TB0En bits (n = 1 to 4)). After that, the value of the sub register (CVSEn0) is written to the main register (CVPEn0).** **Remarks 1.** The operations in the capture register mode and compare register mode when the subchannel n sub capture/compare register (CVSEn0) is not used as a buffer are shown below. • In capture register mode: The CPU can read both the master register (CVPEn0) and slave register (CVSEn0). The next event is ignored until the CPU finishes reading the master register. TM20 capture is performed by the slave register, and TM21 capture is performed by the master register. • In compare register mode: The CPU writes to the slave register (CVSEn0), and immediately after, the same contents as those of the slave register are written to the master register (CVPEn0). **2.** The operations in the capture register mode and compare register mode when the subchannel n sub capture/compare register (CVSEn0) is used as a buffer are shown below. • In capture register mode: When the CPU reads the master register (CVPEn0), the master register updates the value held by the slave register (CVSEn0) immediately before the CPU read operation. When a capture event occurs, the timer/counter value at that time is always saved in the slave register. • In compare register mode: The CPU writes to the slave register (CVSEn0) and these contents are transferred to the master register (CVPEn0) set by the LNKEn bits.

Remark n = 1, 2

(2/2)

Bit position	Bit name	Function
11, 3	LNKEn	Selects capture event signal input from edge selection and specifies transfer operation in compare register mode. 0: ED1 signal input selected in capture register mode. In the compare register mode, the data of the CVSEn0 register is transferred to the CVPEn0 register upon occurrence of a TM2x compare match (TM2x = timer/counter selected by bits TB1En, TB0En). 1: ED2 signal input selected in capture register mode. In the compare register mode, the data of the CVSEn0 register is transferred to the CVPEn0 register when the TM2x count value becomes 0 (TM2x = timer/counter selected by bits TB1En, TB0En).
10, 2	CCSEn	Selects capture/compare register operation mode. 0: Capture register mode 1: Compare register mode
9, 8, 1, 0	TB1En, TB0En	Sets subchannel n timer/counter.

TB1En	TB0En	Subchannel n timer/counter
0	0	Subchannel n not used.
0	1	TM20 set to subchannel n.
1	0	TM21 set to subchannel n.
1	1	32-bit mode[Note] (both TM20 and TM21 selected)

Note In the 32-bit mode, the effect of the BFEEn bit is ignored. Also, the CVSEn0 register cannot be used as a buffer in this mode.

Caution When the TB1En, TB0En bits are set to 11, set the CASE1 bit of the TCRE0 register to 1.

Remark n = 1, 2

(9) Timer 2 subchannel 3, 4 capture/compare control register (CMSE340)

The CMSE340 register controls the timer 2 subchannel n sub capture/compare register (CVSEn0) and the timer 2 subchannel n main capture/compare register (CVPEn0).

This register can be read/written in 16-bit units.

(1/2)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CMSE340	0	0	EEVE4	BFEE4	LNKE4	CCSE4	TB1E4	TB0E4	0	0	EEVE3	BFEE3	LNKE3	CCSE3	TB1E3	TB0E3	FFFFF64EH	0000H

Bit position	Bit name	Function
13, 5	EEVEn	Enables/disables event detection by CMSE340 register. 0: ED1 and ED2 signal inputs ignored (nothing is done even if these signals are input). 1: Operation caused by ED1 and ED2 signal inputs enabled.
12, 4	BFEEn	Specifies the subchannel n sub capture/compare register (CVSEn0) buffer operation. 0: Subchannel n sub capture/compare register (CVSEn0) not used as buffer 1: Subchannel n sub capture/compare register (CVSEn0) used as buffer **Caution When the BFEEn bit = 1, a compare match occurs on starting the timer in the compare register mode because the values of both the TM2x and CVPEn0 registers are 0 after reset (TM2x = timer/counter selected by TB1En and TB0En bits (n = 1 to 4)). After that, the value of the sub register (CVSEn0) is written to the main register (CVPEn0).** **Remarks 1.** The operations in the capture register mode and compare register mode when the subchannel n sub capture/compare register (CVSEn0) is not used as a buffer are shown below. • In capture register mode: The CPU can read both the master register (CVPEn0) and slave register (CVSEn0). The next event is ignored until the CPU finishes reading the master register. TM20 capture is performed by the slave register, and TM21 capture is performed by the master register. • In compare register mode: The CPU writes to the slave register (CVSEn0), and immediately after, the same contents as those of the slave register are written to the master register (CVPEn0). **2.** The operations in the capture register mode and compare register mode when the subchannel n sub capture/compare register (CVSEn0) is used as a buffer are shown below. • In capture register mode: When the CPU reads the master register (CVPEn0), the master register updates the value held by the slave register (CVSEn0) immediately before the CPU read operation. When a capture event occurs, the timer/counter value at that time is always saved in the slave register. • In compare register mode: The CPU writes to the slave register (CVSEn0) and these contents are transferred to the master register (CVPEn0) set by the LNKEn bits.

Remark n = 3, 4

(2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>11, 3</td><td>LNKEn</td><td>Selects capture event signal input from edge selection and specifies transfer operation in compare register mode.
0: ED1 signal input selected in capture register mode.
In the compare register mode, the data of the CVSEn0 register is transferred to the CVPEn0 register upon occurrence of a TM2x compare match (TM2x = timer/ counter selected with bits TB1En, TB0En).
1: ED2 signal input selected in capture register mode.
In the compare register mode, the data of the CVSEn0 register is transferred to the CVPEn0 register when the TM2x count value becomes 0 (TM2x = timer/ counter selected by bits TB1En, TB0En).</td></tr>
<tr><td>10, 2</td><td>CCSEn</td><td>Selects capture/compare register operation mode.
0: Capture register mode
1: Compare register mode</td></tr>
<tr><td>9, 8, 1, 0</td><td>TB1En,
TB0En</td><td>Sets subchannel n timer/counter.
<table><tr><th>TB1En</th><th>TB0En</th><th>Subchannel n timer/counter</th></tr><tr><td>0</td><td>0</td><td>Subchannel n not used</td></tr><tr><td>0</td><td>1</td><td>TM20 set to subchannel n.</td></tr><tr><td>1</td><td>0</td><td>TM21 set to subchannel n.</td></tr><tr><td>1</td><td>1</td><td>32-bit mode^{Note} (both TM20 and TM21 selected)</td></tr></table>Note In the 32-bit mode, the effect of the BFEEn bit is ignored. Also, the CVSEn register cannot be used as a buffer in this mode.

Caution When the TB1En, TB0En bits are set to 11, set the CASE1 bit of the TCRE0 register to 1.</td></tr>
</table>

Remark n = 3, 4

(10) Timer 2 time base status register 0 (TBSTATE0)

The TBSTATE0 register indicates the status of TM2n (n = 0, 1).

This register can be read/written in 16-bit units.

When the higher 8 bits of the TBSTATE0 register are used as the TBSTATE0H register, and the lower 8 bits are used as the TBSTATE0L register, they can be read/written in 8-bit or 1-bit units.

Caution The ECFEn, RSFEn, and UDFEn bits are read-only bits.

	15	14	13	12	<11>	<10>	<9>	<8>	7	6	5	4	<3>	<2>	<1>	<0>	Address	After reset
TBSTATE0	0	0	0	0	OVFE1	ECFE1	RSFE1	UDFE1	0	0	0	0	OVFE0	ECFE0	RSFE0	UDFE0	FFFFF664H	0101H

Bit position	Bit name	Function
11, 3	OVFEn	Indicates TM2n overflow status. 0: No overflow 1: Overflow **Caution If write access to the TBSTATE0 register is performed when an overflow has not been detected, the OVFEn bit is cleared (0).**
10, 2	ECFEn	Indicates the ECLR signal input status. 0: Low level 1: High level
9, 1	RSFEn	Indicates the TM2n count status. 0: TM2n is not counting. 1: TM2n is counting (either up or down)
8, 0	UDFEn	Indicates the TM2n up/down count status. 0: TM2n is in the down count mode. 1: TM2n is in the up count mode.

Remark n = 0, 1

(11) Timer 2 capture/compare 1 to 4 status register 0 (CCSTATE0)

The CCSTATE0 register indicates the status of the timer 2 subchannel sub capture/compare register (CVSEn0) and the timer 2 subchannel main capture/compare register (CVPEn0) (n = 1 to 4).

This register can be read/written in 16-bit units.

When the higher 8 bits of the CCSTATE0 register are used as the CCSTATE0H register, and the lower 8 bits are used as the CCSTATE0L register, they can be read/written in 8-bit or 1-bit units.

Caution The BFFEn1 and BFFEn0 bits are read-only bits.

	15	<14>	13	12	11	<10>	9	8	7	<6>	5	4	3	<2>	1	0	Address	After reset
CCSTATE0	0	CEFE4	BFFE41	BFFE40	0	CEFE3	BFFE31	BFFE30	0	CEFE2	BFFE21	BFFE20	0	CEFE1	BFFE11	BFFE10	FFFFF666H	0000H

Bit position	Bit name	Function
14, 10, 6, 2	CEFEn	Indicates the capture/compare event occurrence status. 0: In capture register mode: No capture operation has occurred. In compare register mode: No compare match has occurred. 1: In capture register mode: At least one capture operation has occurred. In compare register mode: At least one compare match has occurred. **Caution The CEFEn bit can be cleared (0) by performing a write access to the CCSTATE0 register when no capture operation or compare match has occurred. When bit manipulation is performed on the CEFE1 (CEFE3) and CEFE2 (CEFE4) bits, both bits are cleared.**
13, 12, 9, 8, 5, 4, 1, 0	BFFEn1, BFFEn0	Indicates the capture buffer status. BFFEn1 / BFFEn0 / Capture buffer status 0 / 0 / No value in buffer 0 / 1 / Subchannel n master register (CVPEn0) contains a capture value. Slave register (CVSEn0) does not contain a value. 1 / 0 / Both subchannel n master register (CVPEn0) and slave register (CVSEn0) contain a capture value. 1 / 1 / Unused **Caution The BFFEn1 and BFFEn0 bits return a value only when subchannel n sub capture/compare register (CVSEn0) buffer operation (bit BFEEn of CMSEm0 register = 1) is selected or when capture register mode (bit CCSEn of CMSEm0 register = 0) is selected. 0 is read when the compare register mode (CCSEn bit = 1) is selected.**

Remark m = 12, 34
n = 1 to 4

(12) Timer 2 output delay register 0 (ODELE0)

The ODELE0 register sets the output delay operation synchronized with the clock to the TO2n pin's output delay circuit (n = 1 to 4).

This register can be read/written in 16-bit units.

When the higher 8 bits of the ODELE0 register are used as the ODELE0H register, and the lower 8 bits are used as the ODELE0L register, they can be read/written in 8-bit or 1-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
ODELE0	0	ODLE42	ODLE41	ODLE40	0	ODLE32	ODLE31	ODLE30	0	ODLE22	ODLE21	ODLE20	0	ODLE12	ODLE11	ODLE10	FFFFF668H	0000H

Bit position	Bit name	Function
14 to 12, 10 to 8, 6 to 4, 2 to 0	ODLEn2, ODLEn1, ODLEn0	Specifies output delay operation.

ODLEn2	ODLEn1	ODLEn0	Set output delay operation
0	0	0	Output delay operation not performed.
0	0	1	Sets output delay of 1 system clock.
0	1	0	Sets output delay of 2 system clocks.
0	1	1	Sets output delay of 3 system clocks.
1	0	0	Sets output delay of 4 system clocks.
1	0	1	Sets output delay of 5 system clocks.
1	1	0	Sets output delay of 6 system clocks.
1	1	1	Sets output delay of 7 system clocks.

Remark The ODLEn2, ODLEn1, and ODLEn0 bits are used for EMI countermeasures.

Remark n = 1 to 4

(13) Timer 2 software event capture register (CSCE0)

The CSCE0 register sets capture operation by software in the capture register mode.

This register can be read/written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CSCE0	0	0	0	0	0	0	0	0	0	0	SEVE5	SEVE4	SEVE3	SEVE2	SEVE1	SEVE0	FFFFF66AH	0000H

Bit position	Bit name	Function
5 to 0	SEVEn	Specifies capture operation by software in capture register mode. 0: Normal operation continued. 1: Capture operation performed. **Cautions 1. The SEVEn bit ignores the settings of the EEVEn and the LNKEn bits of the CMSEm0 register.** **2. The SEVEn bit is automatically cleared (0) at the end of an event.** **3. The SEVEn bit ignores all the internal limitation statuses of the timer 2 unit.**

Remark m = 12, 34, 05

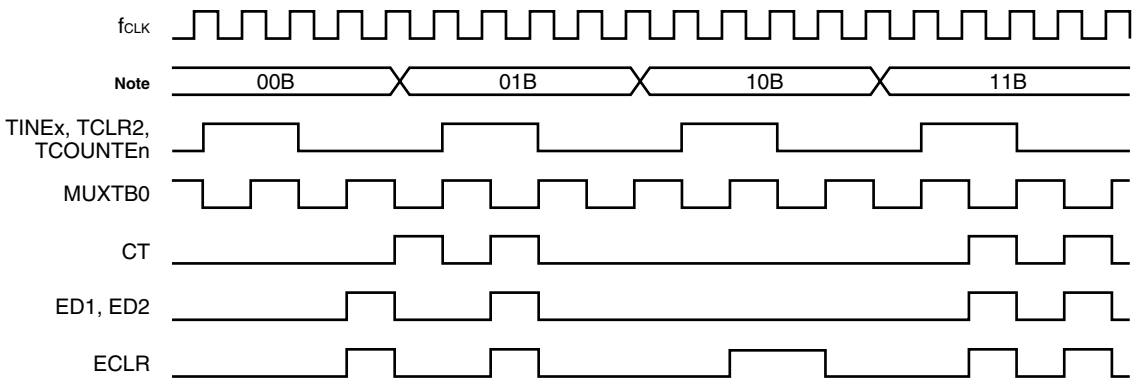
n = 0 to 5

9.3.5 Operation

(1) Edge detection

The edge detection timing is shown below.

Figure 9-66. Edge Detection Timing



Note The set values of the TESnE1 and TESnE0 bits and the CESE1 and CESE0 bits of the CSE0 register, and the IESEx1 and IESEx0 bits of the SESE0 register are shown.

Remarks 1. f_{CLK}: Base clock

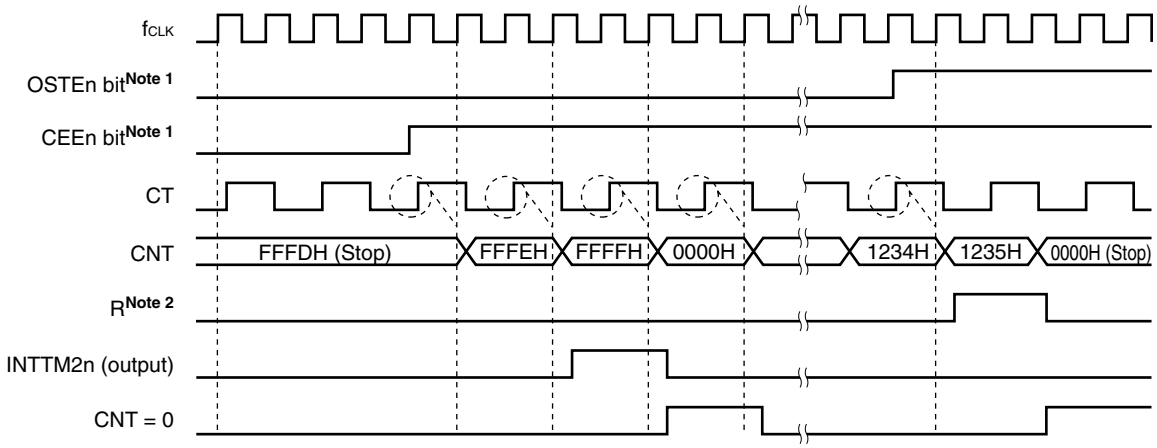
2. CT: TM2n count signal input in the 16-bit mode
ECLR: External control signal input from TCLR2 input
ED1, ED2: Capture event signal input from edge selector
MUXTB0: TM20 multiplex signal
TCOUNTEn: Timer 2 count enable signal input
TINEx: Timer 2 subchannel x capture event signal input (x = 0 to 5)

3. n = 0, 1
x = 0 to 5

(2) Basic operation of timer 2

Figures 9-67 to 9-70 show the basic operation of timer 2.

Figure 9-67. Timer 2 Up Count Timing (When TCRE0 Register's UDSEn1, UDSEn0 Bits = 00B, ECEEn Bit = 0, ECREn Bit = 0, CLREn Bit = 0, CASE1 Bit = 0)



Notes 1. Bits OSTE, CEE of TCRE0 register

2. Can control TM20/TM21 clear by subchannel 0/5 compare match or count direction.

Remarks 1. f_{CLK}: Base clock

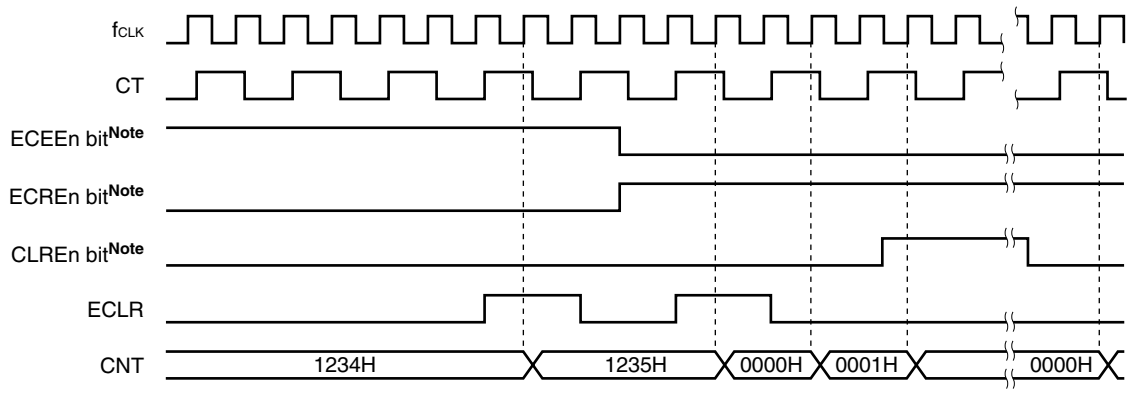
2. CNT: Count value of timer 2

CT: TM2n count signal input in 16-bit mode

R: Compare match signal input (subchannel 0/5)

3. n = 0, 1

Figure 9-68. External Control Timing of Timer 2 (When TCRE0 Register's UDSEn1, UDSEn0 Bits = 00B, OSTEn Bit = 0, CEEn Bit = 1, CASE1 Bit = 0)



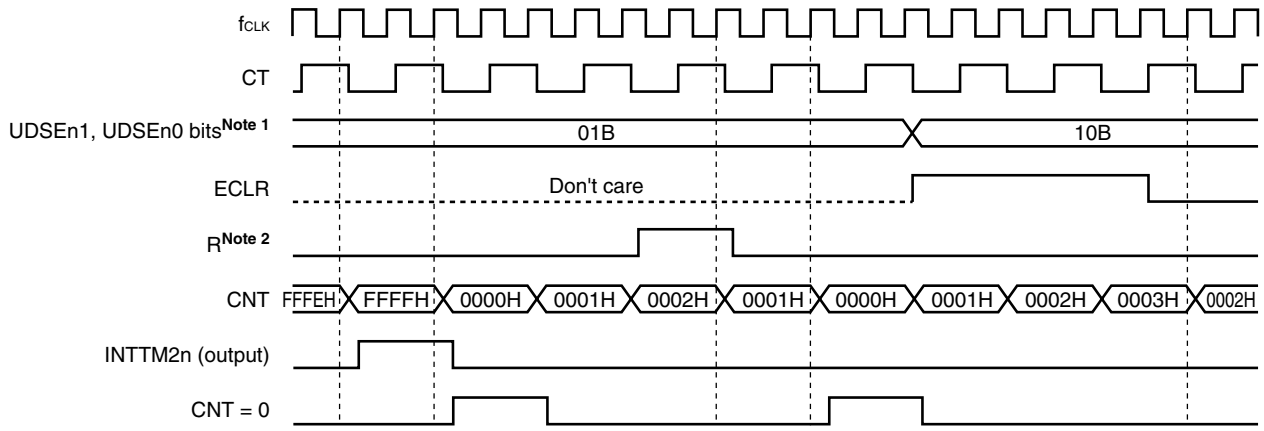
Note Bits ECEEn, ECREn, CLREn of TCRE0 register

Remarks 1. f_{CLK}: Base clock

2. CNT: Count value of timer 2
CT: TM2n count signal input in 16-bit mode
ECLR: External control signal input from TCLR2 pin input

3. n = 0, 1

Figure 9-69. Operation in Timer 2 Up/Down Count Mode (When TCRE0 Register's ECEEn bit = 0, ECREn Bit = 0, CLREn Bit = 0, OSTEn Bit = 0, CEEn Bit = 1, CASE1 Bit = 0)



Notes 1. UDSEn1, UDSEn0 bits of TCRE0 register

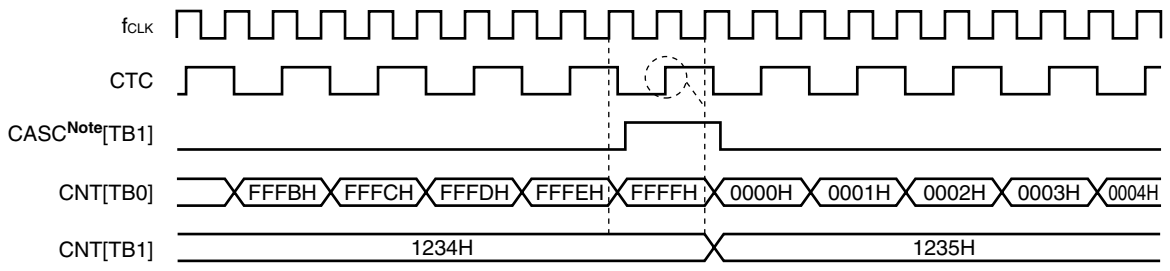
2. Can control TM20/TM21 clear by subchannel 0/5 compare match or count direction.

Remarks 1. f_{CLK}: Base clock

2. CNT: Count value of timer 2
CT: TM2n count signal input in 16-bit mode
ECLR: External control signal input from TCLR2 pin input
R: Compare match signal input (subchannel 0/5)

3. n = 0, 1

Figure 9-70. Timing in 32-Bit Cascade Operation Mode (When TCRE0 Register's UDSEn1, UDSEn0 Bits = 00B, ECEEn Bit = 0, ECREn Bit = 0, CLREn Bit = 0, OSTEn Bit = 0, CEEn Bit = 1, CASE1 Bit = 1)



Note If, in the 32-bit mode, CASC (CNT = MAX. for TM20) is input to TM21 and the CTC rising edge is detected, TM21 performs a count operation.

Remarks 1. f_{CLK}: Base clock

2. CASC: TM21 count signal input in 32-bit mode
CNT: Count value of timer 2
CTC: TM21 count signal input in 32-bit mode
TB0: Count value of TM20
TB1: Count value of TM21

3. n = 0, 1

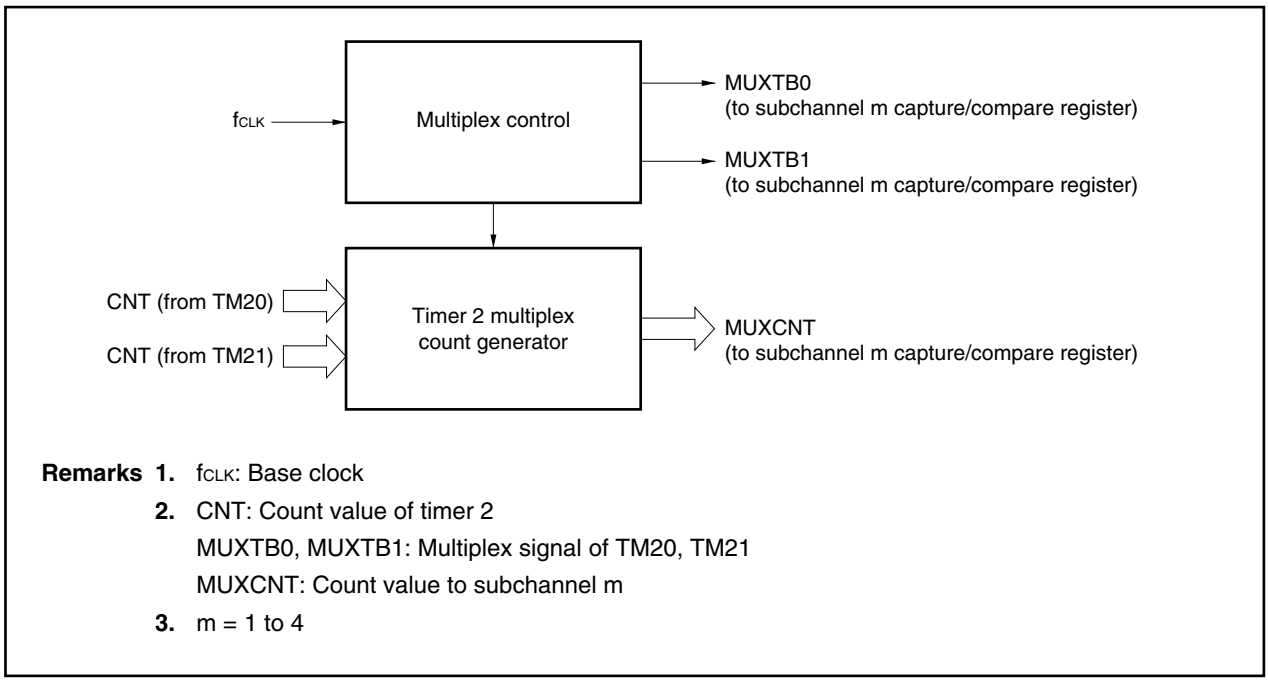
(3) Operation of capture/compare register (subchannels 1 to 4)

Subchannels 1 to 4 receive the count value of the timer 2 multiplex count generator.

The multiplex count generator is an internal unit of TM2n that supplies the multiplex count value MUXCNT to subchannels 1 to 4. The count value of TM20 is output to subchannels 1 to 4 at the rising edge of MUXTB0, and the count value of TM21 is output to subchannels 1 to 4 at the rising edge of MUXTB1.

Figure 9-71 shows the block diagram of the timer 2 multiplex count generator, and Figure 9-72 shows the multiplex count timing.

Figure 9-71. Block Diagram of Timer 2 Multiplex Count Generator

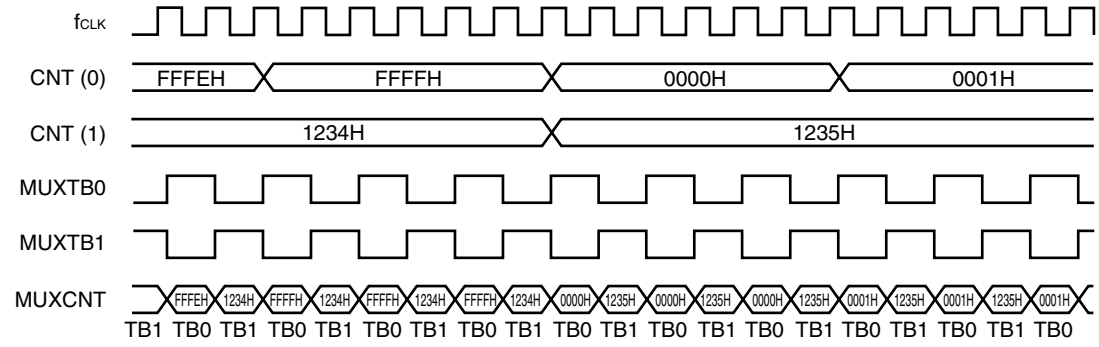


Remarks 1. f_{CLK}: Base clock

2. CNT: Count value of timer 2
MUXTB0, MUXTB1: Multiplex signal of TM20, TM21
MUXCNT: Count value to subchannel m

3. m = 1 to 4

Figure 9-72. Multiplex Count Timing

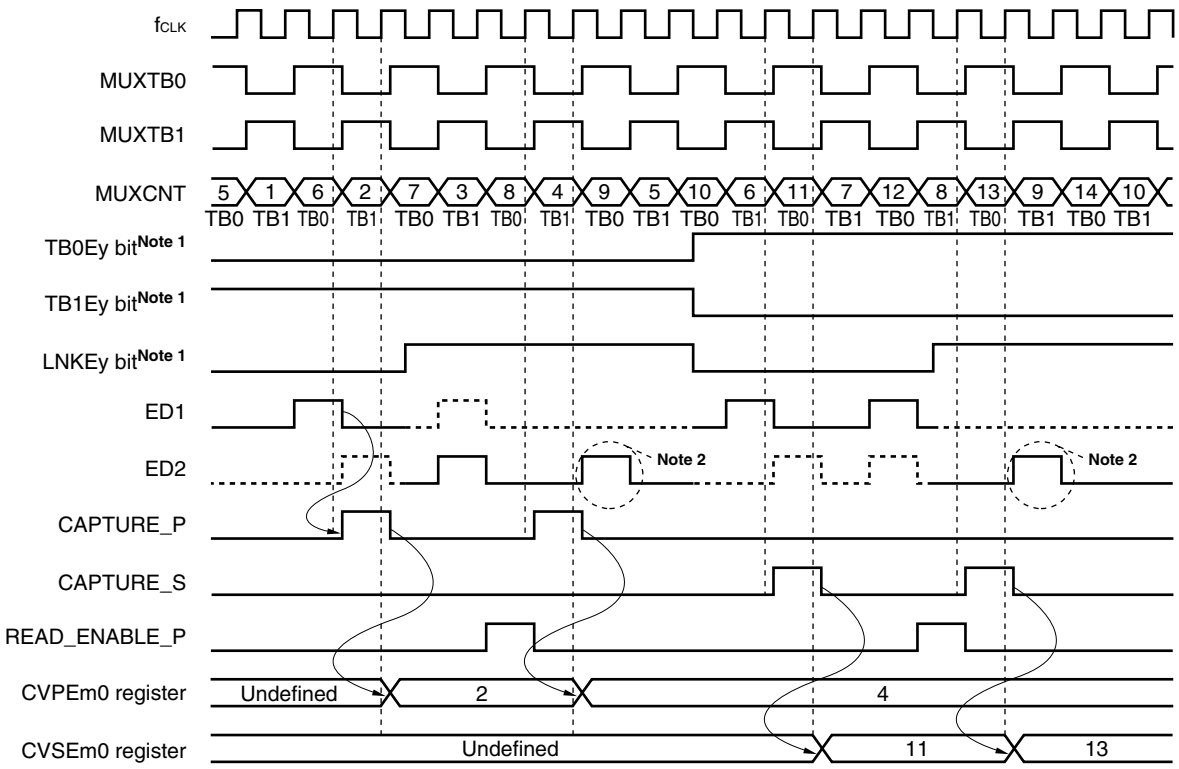


Remarks 1. f_{CLK}: Base clock

2. CNT: Count value of timer 2
MUXTB0, MUXTB1: Multiplex signal of TM20, TM21
MUXCNT: Count value to subchannel m (m = 1 to 4)
TB0: Count value of TM20
TB1: Count value of TM21

Figures 9-73 to 9-78 show the operation of the capture/compare register (subchannels 1 to 4).

Figure 9-73. Capture Operation: 16-Bit Buffer-Less Mode (When Operation Is Delayed Through Setting of LNKEy Bit of CMSEx0 Register, and CMSEx0 Register's CCSEy Bit = 0, BFEEy Bit = 0, EEVEy Bit = 1, and CSCE0 Register's SEVEy Bit = 0)



Notes 1. Bits TB0Ey, TB1Ey of CMSEx register

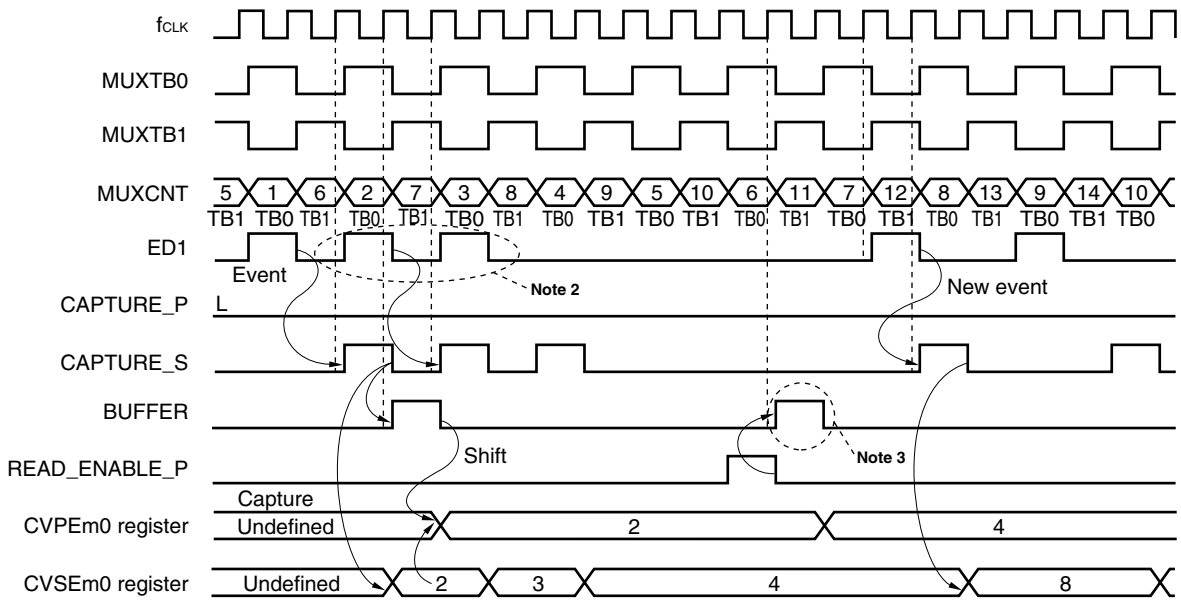
2. If an event occurs at this timing, it is ignored.

Remarks 1. f_{CLK}: Base clock

2. CAPTURE_P: Capture trigger signal of main capture register
CAPTURE_S: Capture trigger signal of sub capture register
ED1, ED2: Capture event signal input from edge selector
MUXCNT: Count value to subchannel m
MUXTB0, MUXTB1: Multiplex signal of TM20, TM21
READ_ENABLE_P: Read timing for CVPEm0 register
TB0: Count value of TM20
TB1: Count value of TM21

3. m = 1 to 4, x = 12, 34
y: When x = 12, y = 1, 2, and when x = 34, y = 3, 4

Figure 9-74. Capture Operation: Mode with 16-Bit Buffer[Note 1] (When CMSEx0 Register's TByE1 Bit = 0, TByE0 Bit = 1, CCSEy Bit = 0, LNKEy Bit = 0, BFEEy Bit = 1, EEVEy Bit = 1, and CSCE0 Register's SEVEy Bit = 0)



Notes 1. To operate TM2n in the mode with 16-bit buffer, perform a capture at least twice at the start of an operation and read the CVPEm0 register. Also, read the CVPEm0 register after performing a capture at least once.

2. A write operation to the CVPEn0 register is not performed at these signal inputs because the CVSEm0 register operates as a buffer.

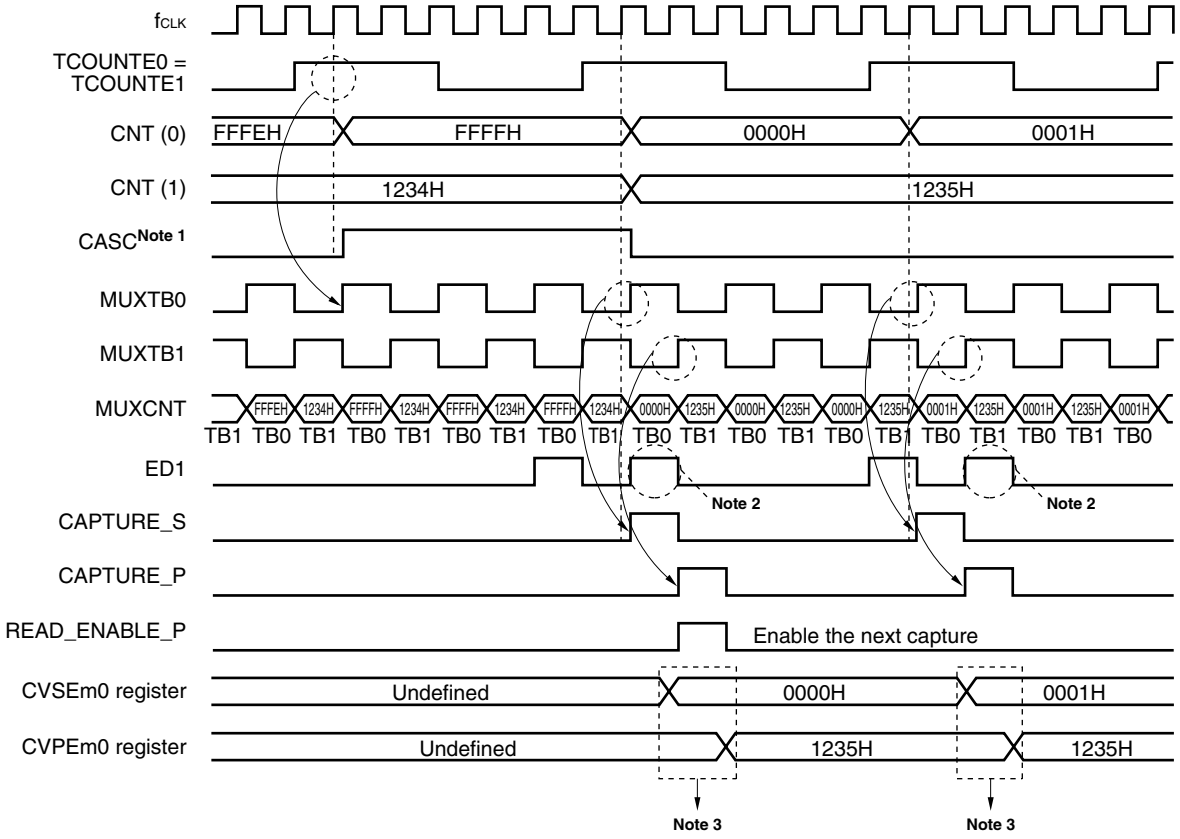
3. After this timing, a write operation from the CVSEm0 register to the CVPEm0 register is enabled.

Remarks 1. f_{CLK}: Base clock

2. BUFFER: Timing of write operation from CVSEm0 register to CVPEm0 register
CAPTURE_P: Capture trigger signal of main capture register
CAPTURE_S: Capture trigger signal of sub capture register
ED1: Capture event signal input from edge selector
MUXCNT: Count value to subchannel m
MUXTB0, MUXTB1: Multiplex signal of TM20, TM21
READ_ENABLE_P: Read timing of CVPEm0 register
TB0: Count value of TM20; TB1: Count value of TM21

3. m = 1 to 4, x = 12, 34
y: When x = 12, y = 1, 2, and when x = 34, y = 3, 4

Figure 9-75. Capture Operation: 32-Bit Cascade Operation Mode (When CMSEx Register's TByE1 Bit = 1, TByE0 Bit = 1, CCSEy Bit = 0, LNKEy Bit = 0, BFEEy Bit = Arbitrary, EEVEy Bit = 1, and CSCE0 Register's SEVEy Bit = 0)



Notes 1. TM21 performs a count operation when, in the 32-bit mode, CASC (CNT = MAX. for TM20) is input to TM21 and the rising edge of CTC is detected.

2. If an event occurs during this timing, it is ignored.

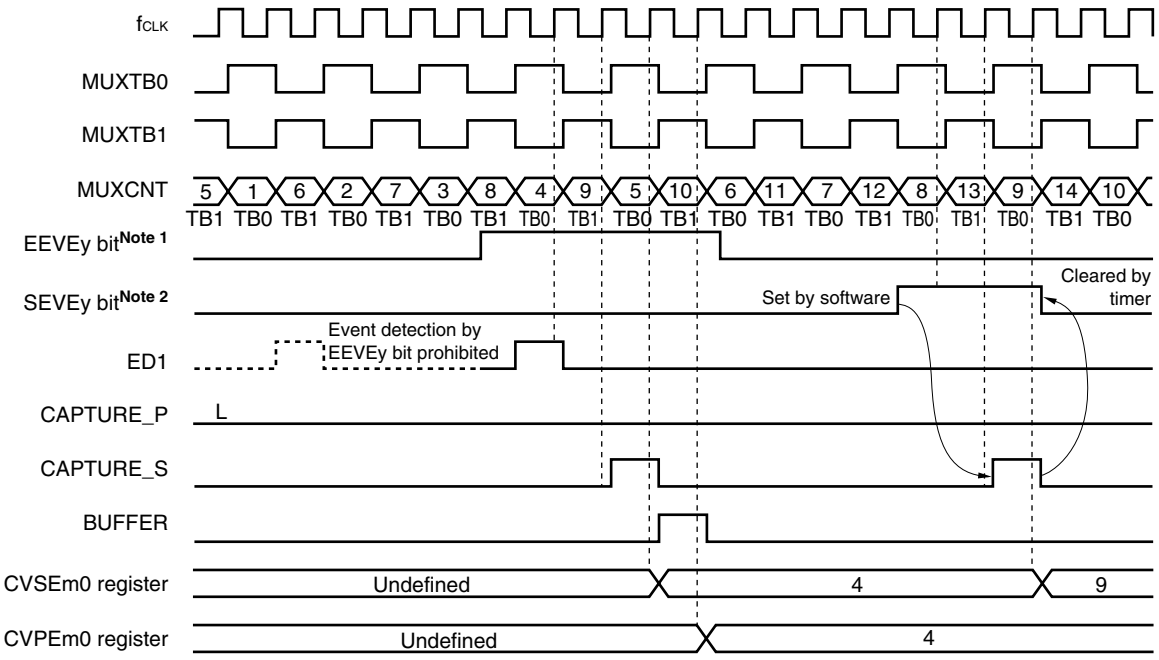
3. CPU read access is not performed at this timing (wait status).

Remarks 1. f_{CLK}: Base clock

2. CAPTURE_P: Capture trigger signal of main capture register
CAPTURE_S: Capture trigger signal of sub capture register
CASC: TM21 count signal in 32-bit mode
CNT: Count value of timer 2
ED1: Capture event signal input from edge selector
MUXCNT: Count value to subchannel m
MUXTB0, MUXTB1: Multiplex signal of TM20, TM21
READ_ENABLE_P: Read timing of CVPEm0 register
TB0: Count value of TM20
TB1: Count value of TM21
TCOUNTE0, TCOUNTE1: Count enable signal input of timer 2

3. m = 1 to 4, x = 12, 34
y: When x = 12, y = 1, 2, and when x = 34, y = 3, 4

Figure 9-76. Capture Operation: Capture Control by Software and Trigger Timing (When CMSEx0 Register's TByE1 Bit = 0, TByE0 Bit = 1, CCSEy Bit = 0, LNKEy Bit = 0, BFEEy Bit = 1)



Notes 1. EEVEy bit of CMSEx0 register

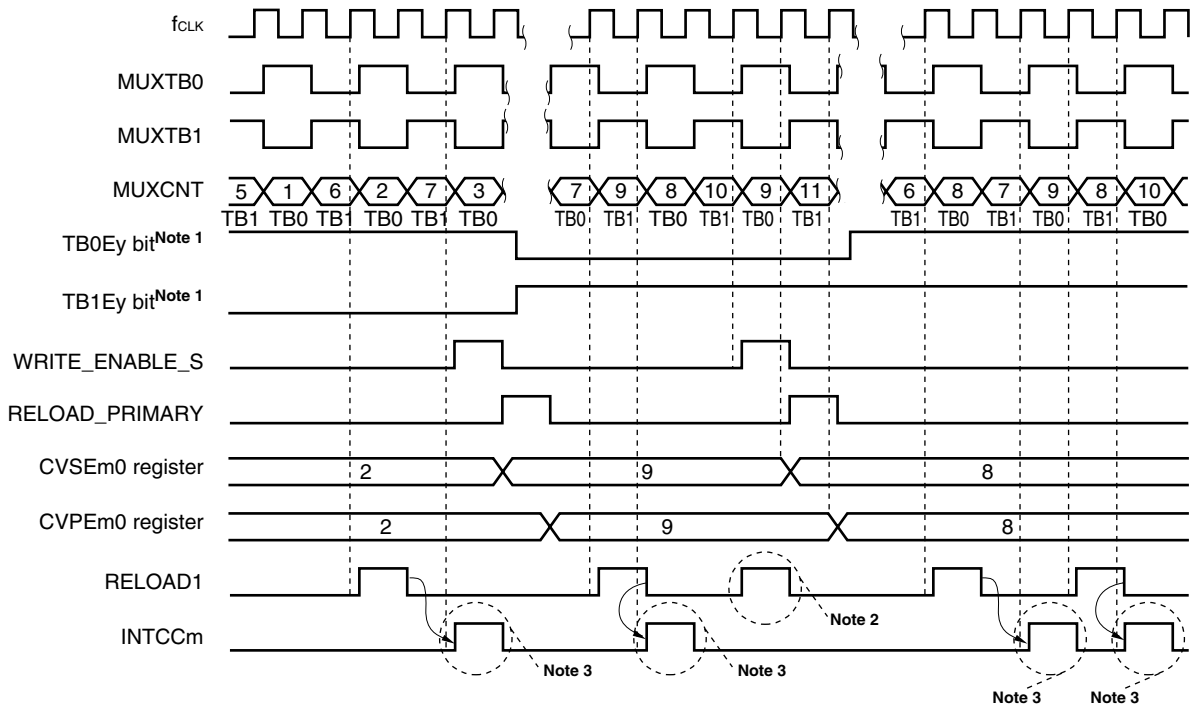
2. SEVEy bit of CSCE0 register

Remarks 1. f_{CLK}: Base clock

2. BUFFER: Timing of write operation from CVSEm0 register to CVPEm0 register
CAPTURE_P: Capture trigger signal of main capture register
CAPTURE_S: Capture trigger signal of sub capture register
ED1: Capture event signal input from edge selector
MUXCNT: Count value to subchannel m
MUXTB0, MUXTB1: Multiplex signal of TM20, TM21
TB0: Count value of TM20
TB1: Count value of TM21

3. m = 1 to 4, x = 12, 34
y: When x = 12, y = 1, 2, and when x = 34, y = 3, 4

Figure 9-77. Compare Operation: Buffer-Less Mode (When CMSEx0 Register's CCSEy Bit = 1, LNKEy Bit = Arbitrary, BFEEy Bit = 0)



Notes 1. TB1Ey, TB0Ey bits of CMSEx0 register

2. No interrupt is generated due to a compare match with counter differing from that set by the TB1Ey and TB0Ey bits.

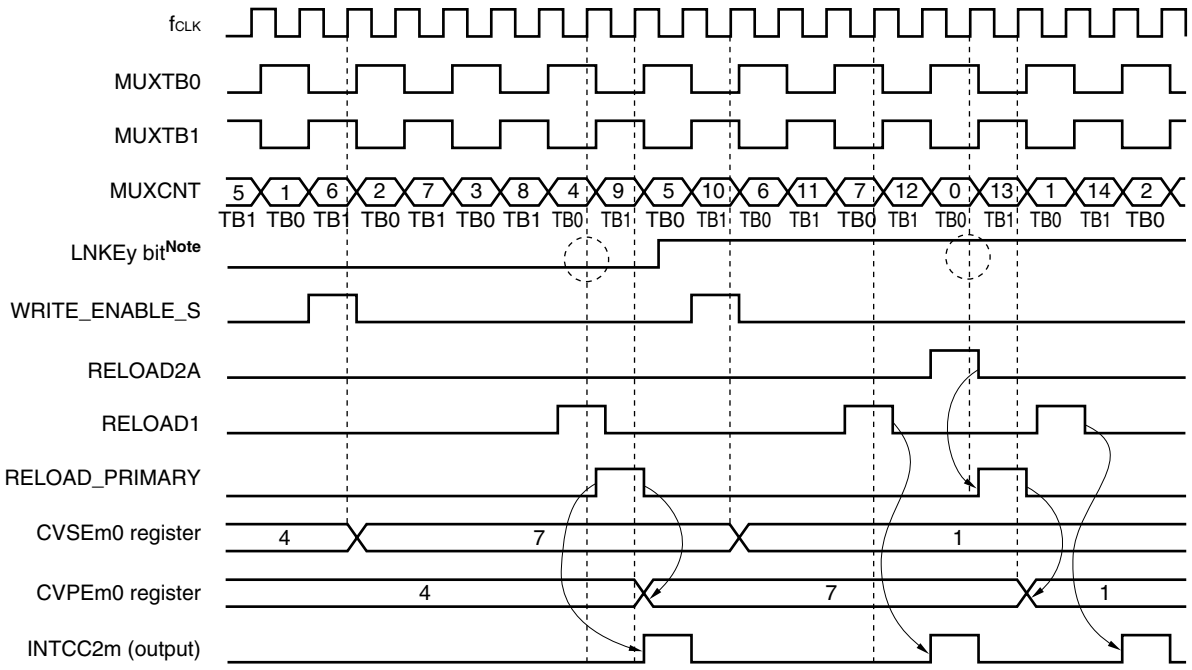
3. INTCC2m is generated to match the cycle from the rising edge to the falling edge of MUXTB0.

Remarks 1. fCLK: Base clock

2. MUXCNT: Count value to subchannel m
MUXTB0, MUXTB1: Multiplex signal of TM20, TM21
RELOAD1: Compare match signal
RELOAD_PRIMARY: Timing of write operation from CVSEm0 register to CVPEm0 register
WRITE_ENABLE_S: Timing of CVSEm0 register write operation
TB0: Count value of TM20
TB1: Count value of TM21

3. m = 1 to 4, x = 12, 34

Figure 9-78. Compare Operation: Mode with Buffer (When Operation Is Delayed Through Setting of LNKEy Bit of CMSEx0 Register, CMSEx0 Register's CCSEy Bit = 1, BFEEy Bit = 1)



Note LNKEy bit of CMSEx0 register

Remarks 1. f_{CLK}: Base clock

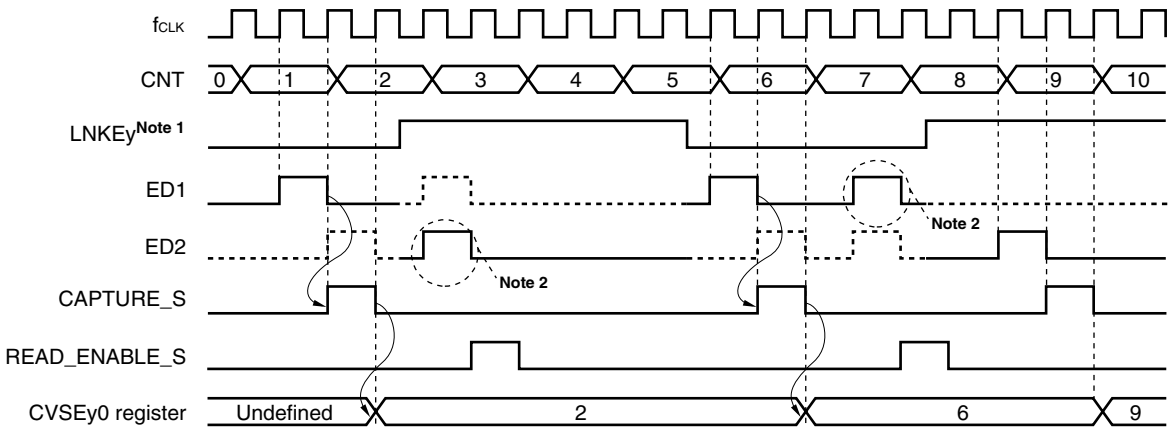
2. MUXCNT: Count value to subchannel m
MUXTB0, MUXTB1: Multiplex signal of TM20, TM21
RELOAD1: Compare match signal
RELOAD2A: Zero count signal input of TM20 (occurs when TM20 = 0000H)
RELOAD_PRIMARY: Timing of write operation from CVSEm0 register to CVPEm0 register
WRITE_ENABLE_S: Timing of CVSEm0 register write operation
TB0: Count value of TM20 (in this figure, the maximum count value is 7)
TB1: Count value of TM21

3. m = 1 to 4, x = 12, 34
y: When x = 12, y = 1, 2, and when x = 34, y = 3, 4

(4) Operation of capture/compare register (subchannels 0, 5)

Figures 9-79 and 9-80 show the operation of the capture/compare register (subchannels 0, 5).

Figure 9-79. Capture Operation: Timer 2 Count Value Read Timing (When CMSE050 Register's CCSEy Bit = 0, EEVEy Bit = 1, and CSCE0 Register's SEVEy Bit = 0)



Notes 1. LNKEy bit of CMSE050 register

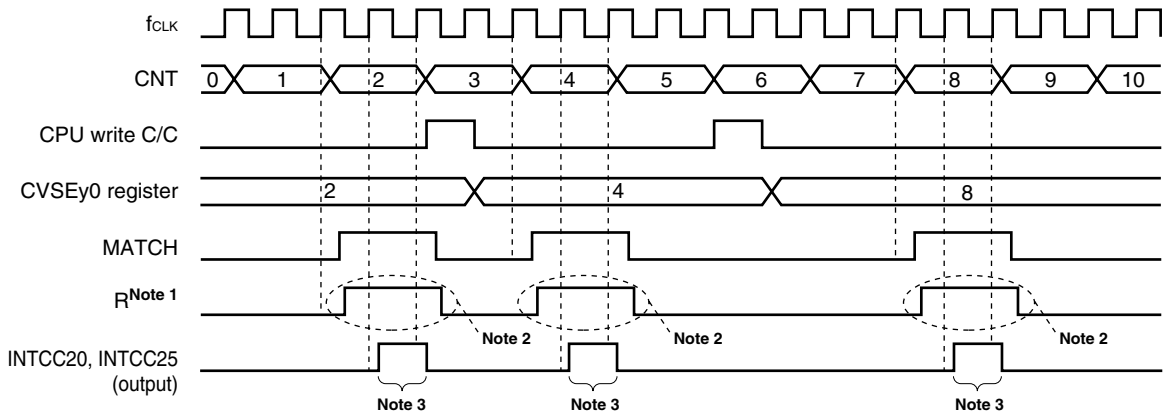
2. If an event occurs at this timing, it is ignored.

Remarks 1. f_{CLK}: Base clock

2. CNT: Count value of timer 2
CAPTURE_S: Capture trigger signal of sub capture register
ED1, ED2: Capture event signal inputs from edge selector
READ_ENABLE_S: Read timing for CVSEy0 register

3. y = 0, 5

Figure 9-80. Compare Operation: Timing of Compare Match and Write Operation to Register (When CMSE050 Register's CCSEy Bit = 1, EEVEy Bit = Arbitrary, and CSCE0 Register's SEVEy Bit = Arbitrary)



Notes 1. Can control TM20/TM21 clear by subchannel 0/5 compare match and count direction

2. When the MATCH signal occurs, the same waveform as the MATCH signal is generated.

3. The pulse width is always 1 clock.

Remarks 1. f_{CLK}: Base clock

2. CNT: Count value of timer 2

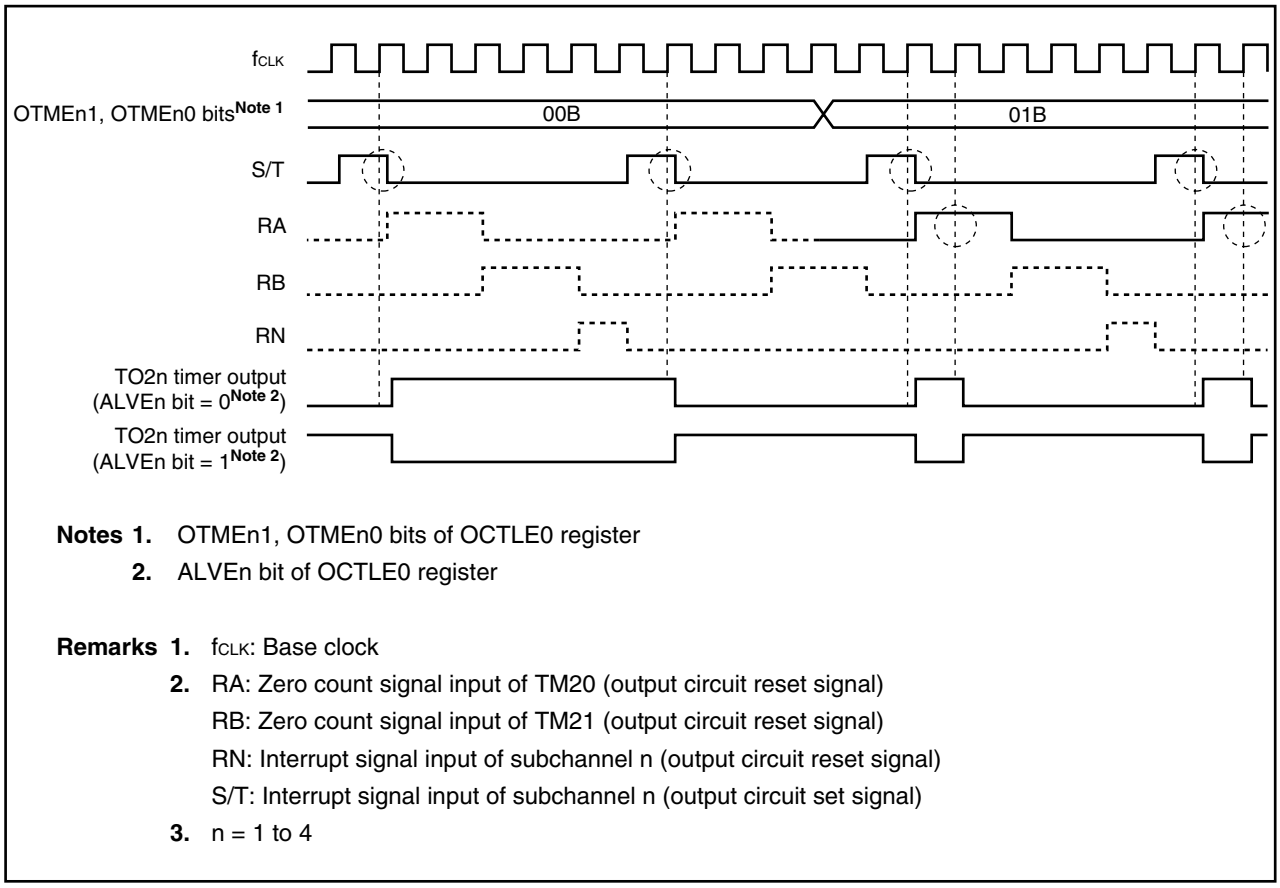
MATCH: CVSEy0 register compare match timing

R: Compare match input (subchannel 0/5)

(5) Operation of output circuit

Figures 9-81 to 9-84 show the output circuit operation.

Figure 9-81. Signal Output Operation: Toggle Mode 0 and Toggle Mode 1 (When OCTLE0 Register's SWFEn Bit = 0, and ODELE0 Register's ODLEn2 to ODLEn0 Bits = 0)



Notes 1. OTMEn1, OTMEn0 bits of OCTLE0 register

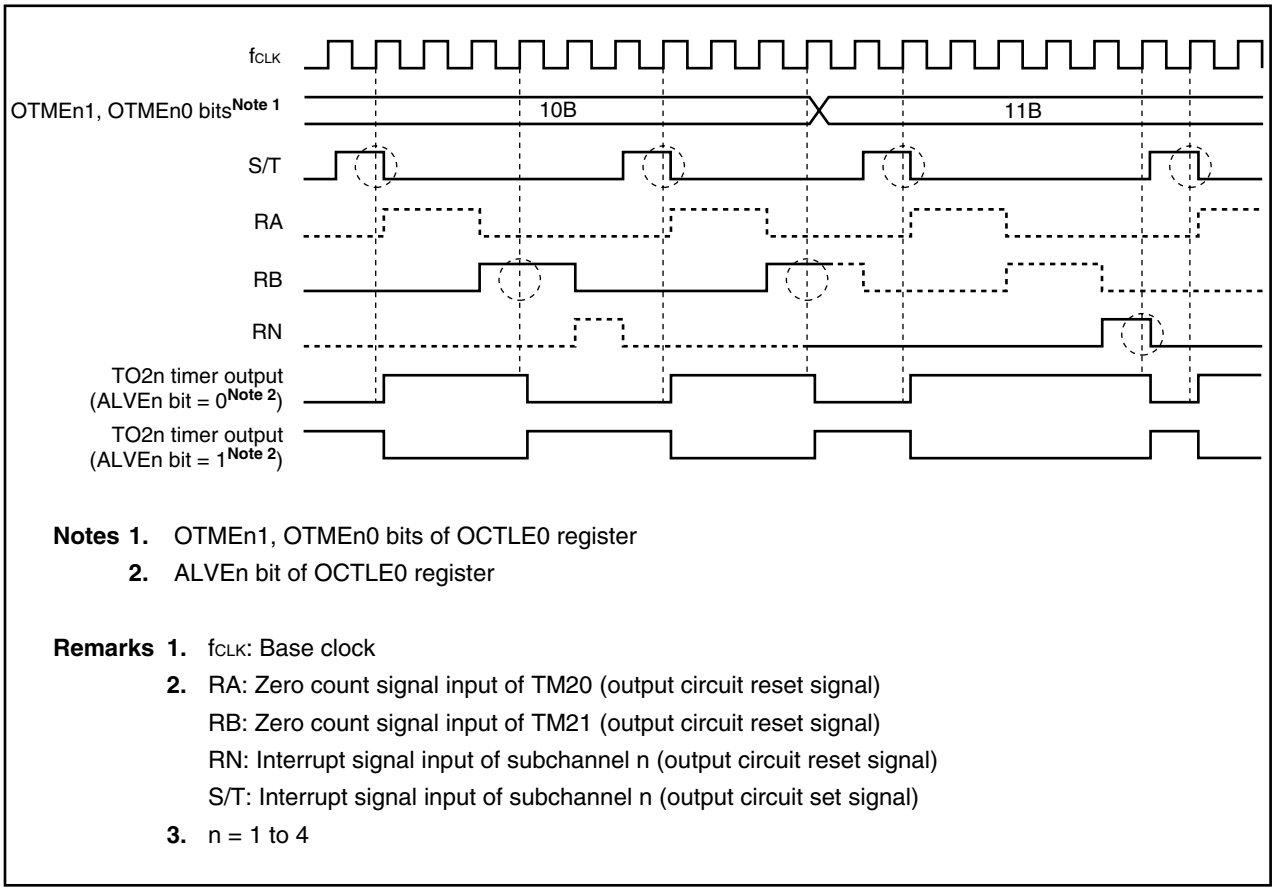
2. ALVEn bit of OCTLE0 register

Remarks 1. f_{CLK}: Base clock

2. RA: Zero count signal input of TM20 (output circuit reset signal)
RB: Zero count signal input of TM21 (output circuit reset signal)
RN: Interrupt signal input of subchannel n (output circuit reset signal)
S/T: Interrupt signal input of subchannel n (output circuit set signal)

3. n = 1 to 4

Figure 9-82. Signal Output Operation: Toggle Mode 2 and Toggle Mode 3 (When OCTLE0 Register's SWFEn Bit = 0, and ODELE0 Register's ODLEn2 to ODLEn0 Bits = 0)



Notes 1. OTMEn1, OTMEn0 bits of OCTLE0 register

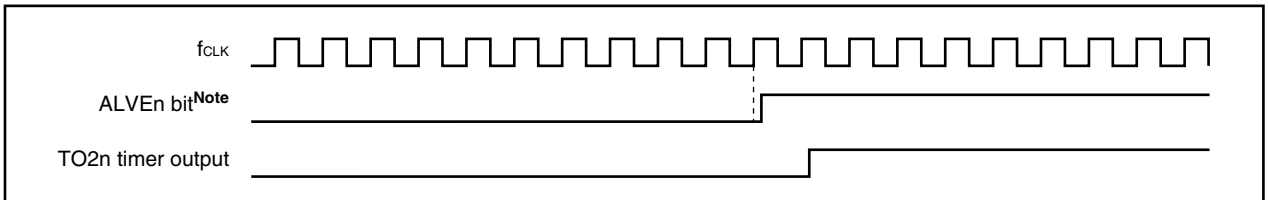
2. ALVEn bit of OCTLE0 register

Remarks 1. f_{CLK}: Base clock

2. RA: Zero count signal input of TM20 (output circuit reset signal)
RB: Zero count signal input of TM21 (output circuit reset signal)
RN: Interrupt signal input of subchannel n (output circuit reset signal)
S/T: Interrupt signal input of subchannel n (output circuit set signal)

3. n = 1 to 4

Figure 9-83. Signal Output Operation: During Software Control (When OCTLE0 Register's OTMEn1, OTMEn0 Bits = Arbitrary, SWFEn Bit = 1, and ODELE0 Register's ODLEn2 to ODLEn0 Bits = 0)

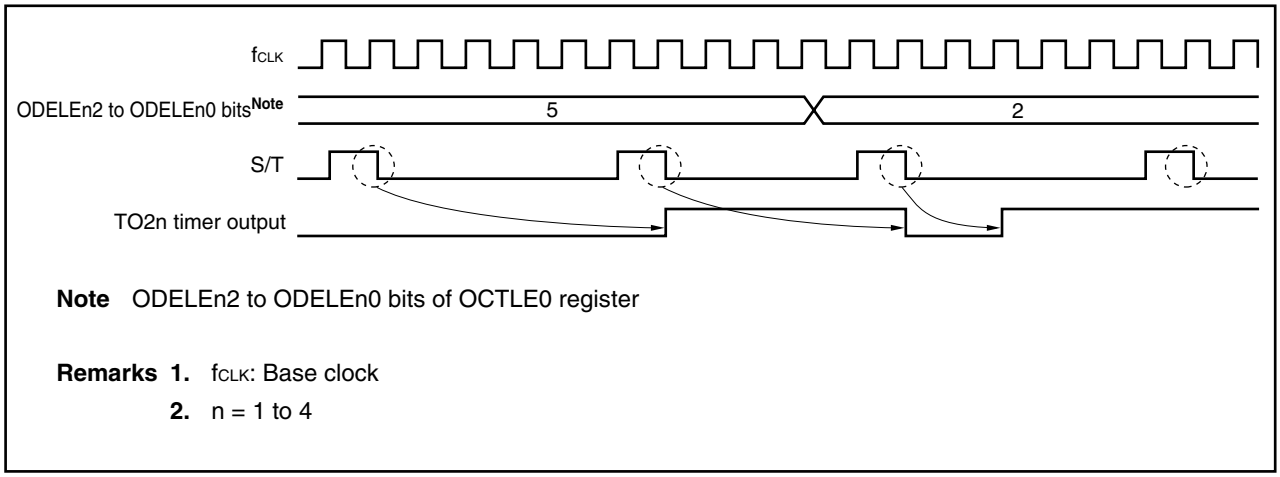


Note ALVEn bit of OCTLE0 register

Remarks 1. f_{CLK}: Base clock

2. n = 1 to 4

Figure 9-84. Signal Output Operation: During Delay Output Operation (When OCTLE0 Register's OTMEn1, OTMEn0 Bits = 0, ALVEn = 0, SWFEn Bit = 0)



Note ODELEn2 to ODELEn0 bits of OCTLE0 register

Remarks 1. f_{CLK}: Base clock

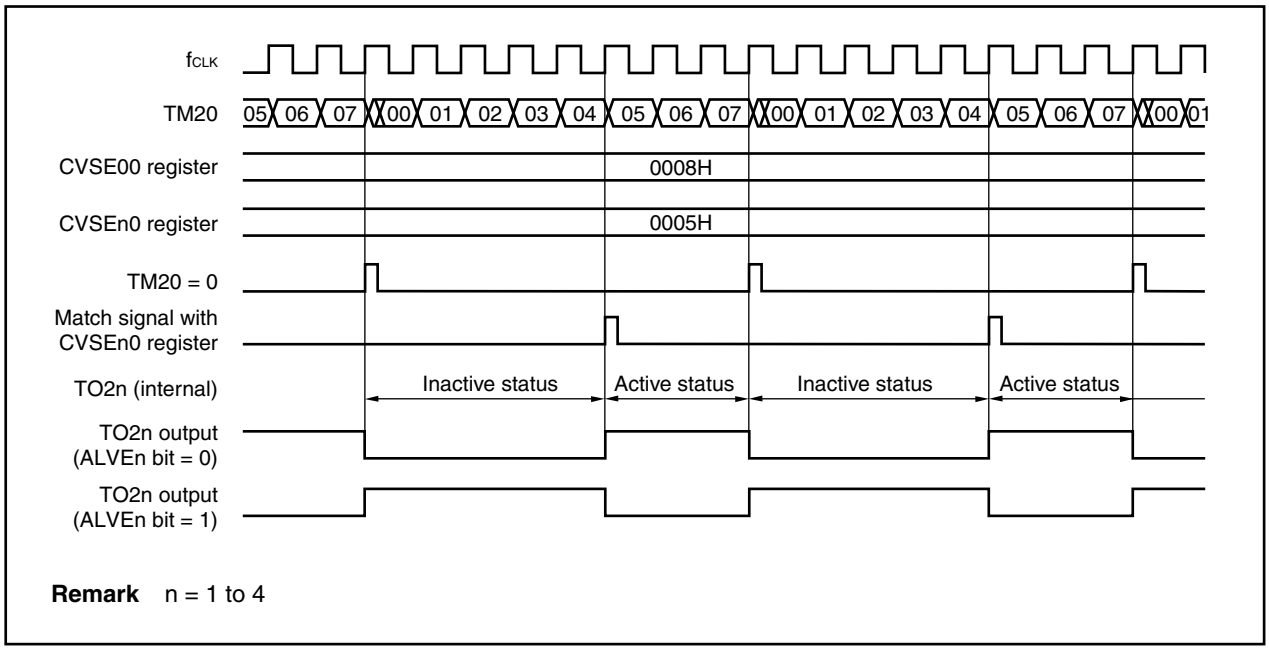
2. n = 1 to 4

9.3.6 PWM output operation in timer 2 compare mode

(1) Operation during PWM output operation of TO2n pin in toggle mode 1

In toggle mode 1, the output of TO2n (internal) is made inactive at the trigger signal when TM20 = 0, and the output of TO2n (internal) is made active triggered by a compare match signal with subchannel 1 (the CVSEn0 register). The TO2n pin outputs a high level or low level according to the TO2n (internal) status and the value of the OCTLE0.ALVEn bit.

Figure 9-85. During Normal Output Operation
(When OTMEn1, OTMEn0 Bits = 01 in OCTLE0 Register, ODLEn2 to ODLEn0 Bits = 000 in ODELE0 Register)



Remark n = 1 to 4

(2) Operations when output of the TO2n pin is controlled by manipulating the OCTLE0.SWFEn bit in toggle mode 1

(a) When compare match signal of subchannel n is output immediately after the SWFEn bit changes from 1 to 0

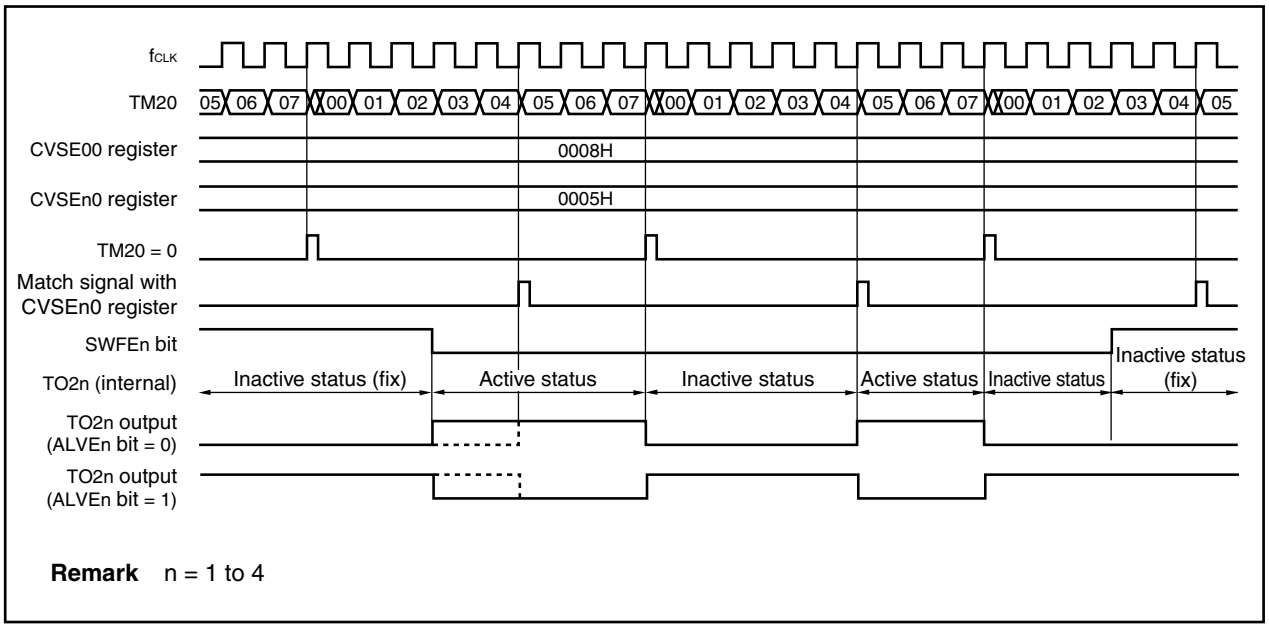
Figures 9-86 and 9-87 show the waveform of each block at output start/end when the output of the TO2n output pin is controlled by manipulating the SWFEn bit in toggle mode 1.

Timer 2 of the V850E/IA2 outputs levels according to the value of the ALVEn bit (low level when the ALVEn bit is 0, high level when the ALVEn bit is 1) by fixing the TO2n output to the inactive status. When the SWFEn bit is 0, timer 2 outputs an active level or inactive level by making TO2n (internal) operate according to the trigger signal.

However, if the SWFEn bit is changed from 1 to 0, forcibly activate the TO2n output once. If the SWFEn bit is changed from 0 to 1, forcibly fix the TO2n output to the inactive status.

If the compare match signal of subchannel n is output immediately after the SWFEn bit has been changed from 1 to 0, the period from when the SWFEn bit changes from 1 to 0 until the compare match signal is output is added to the active period of the normal TO2n output, lengthening the first active period (refer to **Figure 9-86**).

Figure 9-86. When Normal Output Operation Starts/Ends
(When OTMEn1, OTMEn0 Bits = 01 in OCTLE0 Register, ODLEn2 to ODLEn0 Bits = 000 in ODELD0 Register)



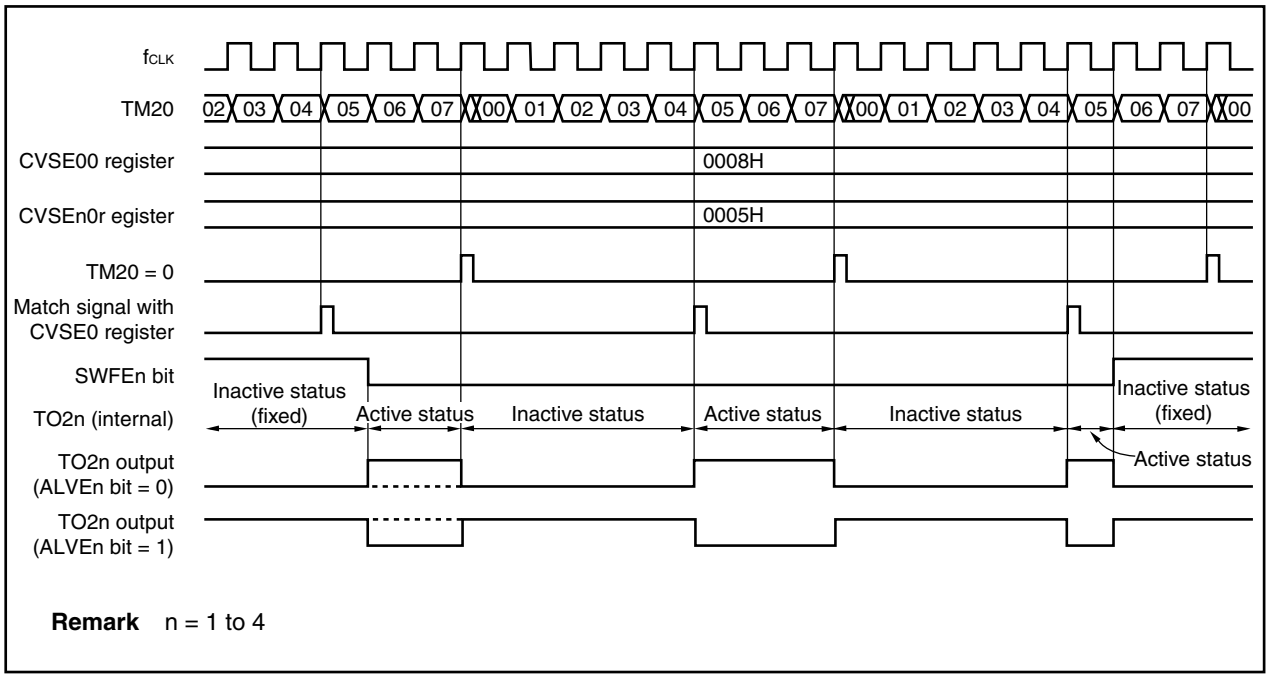
Remark n = 1 to 4

(b) When the trigger signal of TM20 = 0 is output immediately after the SWFEn bit is changed from 1 to 0

When the trigger signal of TM20 = 0 is output immediately after the SWFEn bit is changed from 1 to 0, the initial active period is from when the SWFEn bit is changed from 1 to 0 until the trigger signal of TM20 is output. Therefore, a shorter pulse than the active period of the normal TO2n output is output.

When the SWFEn bit is changed from 0 to 1, the TO2n output is forcibly fixed to inactive. If this operation is generated while active level is output, the active level output period is shorter (refer to **Figure 9-87**).

Figure 9-87. When Normal Output Operation Starts/Ends
(When OTMEn1, OTMEn0 Bits = 01 in OCTLE0 Register, ODLEn2 to ODLEn0 Bits = 000 in ODELD0 Register)



Remark n = 1 to 4

9.4 Timer 3

9.4.1 Features (timer 3)

Timer 3 (TM3) is a 16-bit timer/counter that can perform the following operations.

- Interval timer function
- PWM output
- External signal cycle measurement
- TO3 output buffer set to off by INTP4 input

9.4.2 Function overview (timer 3)

- 16-bit timer/counter (TM3): 1 channel
- Capture/compare registers: 2
- Count clock division selectable by prescaler (set the frequency of the count clock to 16 MHz or less)
- Base clock (f_{CLK}): 2 types (set f_{CLK} to 32 MHz or less)
  f_{XX} and $f_{XX}/2$ can be selected
- Prescaler division ratio
  The following division ratios can be selected according to the base clock (f_{CLK}).

Division Ratio	Base Clock (f_{CLK})	
	f_{XX} Selected	$f_{XX}/2$ Selected
1/2	$f_{XX}/2$	$f_{XX}/4$
1/4	$f_{XX}/4$	$f_{XX}/8$
1/8	$f_{XX}/8$	$f_{XX}/16$
1/16	$f_{XX}/16$	$f_{XX}/32$
1/32	$f_{XX}/32$	$f_{XX}/64$
1/64	$f_{XX}/64$	$f_{XX}/128$
1/128	$f_{XX}/128$	$f_{XX}/256$
1/256	$f_{XX}/256$	$f_{XX}/512$

- Interrupt request sources
  - Capture/compare match interrupt requests: 2 sources
    In case of capture register: INTCC3n generated by INTP3n input
    In case of compare register: INTCC3n generated by CC3n match signal
  - Overflow interrupt request: 1 source
    INTTM3 generated upon overflow of TM3 register
- Timer/counter count clock sources: 2 types
  (Selection of external pulse input, internal system clock cycle)
- One of two operation modes when the timer/counter overflows can be selected: free-running mode or overflow stop mode
- The timer/counter can be cleared by match of timer/counter and compare register
- External pulse output (TO3): 1
- TO3 output buffer set to off by INTP4 input (high-impedance state)

Remarks 1. f_{XX}: Internal system clock

2. n = 0, 1

9.4.3 Function added to V850E/IA2

Timer 3 (TM3) of the V850E/IA2 has an added function to control TO3 output by using the INTP4 pin. This additional function can be used to forcibly stop TO3 output, if any abnormality is detected, by inputting a signal to the INTP4 pin. This TO3 output stop function can also be used even when the clock supply is stopped.

9.4.4 Basic configuration

Table 9-13. Timer 3 Configuration List

Timer	Count Clock		Register	Read/Write	Generated Interrupt Signal	Capture Trigger	Timer Output S/R
	Note 1	**Note 2**					
Timer 3	$f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$, $f_{XX}/256$	$f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$, $f_{XX}/256$, $f_{XX}/512$	TM3	Read	INTTM3	–	–
			CC30	Read/write	INTCC30	INTP30	TO3 (S)
			CC31	Read/write	INTC31	INTP31	TO3 (R)

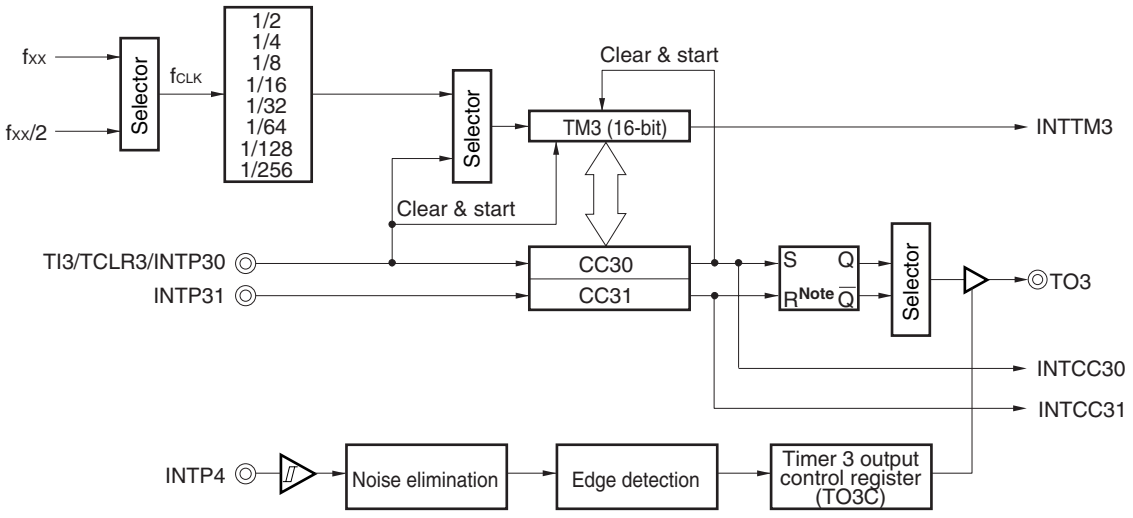
Notes 1. When f_{XX} is selected as the base clock (f_{CLK}) of TM3

2. When $f_{XX}/2$ is selected as the base clock (f_{CLK}) of TM3

Remark f_{XX}: Internal system clock
S/R: Set/Reset

Figure 9-88 shows the block diagram of timer 3.

Figure 9-88. Block Diagram of Timer 3



Note Reset priority

Remarks 1. TI3 input and TCLR3 input connected to port immediately before edge detection

2. f_{CLK}: Base clock (32 MHz (MAX.))
f_{XX}: Internal system clock

(1) Timer 3 (TM3)

TM3 functions as a 16-bit free-running timer or as an event counter for an external signal. Besides being mainly used for cycle measurement, TM3 can be used as pulse output.

TM3 is read-only, in 16-bit units.

Cautions 1. The TM3 register can only be read. If writing is performed to the TM3 register, the subsequent operation is undefined.

2. If the TM3CAE bit of the TMC30 register is cleared (0), a reset is performed asynchronously.

3. Continuous reading of TM3 is prohibited. If TM3 is continuously read, the second read value may differ from the actual value.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
TM3																	FFFFF680H	0000H

TM3 performs the count-up operations of an internal count clock or external count clock. Timer starting and stopping are controlled by the TM3CE bit of timer control register 30 (TMC30).

The internal or external count clock is selected by the ETI bit of timer control register 31 (TMC31).

(a) Selection of the external count clock

TM3 operates as an event counter.

When the ETI bit of timer control register 31 (TMC31) is set (1), TM3 counts the valid edges of the external clock input (TI3), synchronized with the internal count clock. The valid edge is specified by valid edge selection register (SESC).

Caution When using the INTP30, TI3, and TCLR3 pins as TI3 andTCLR3, either mask the interrupt signal to INTP30 or set CC3n in compare mode (n = 0 or 1).

(b) Selection of the internal count clock

TM3 operates as a free-running timer.

When an internal clock is specified as a count clock by timer control register 31 (TMC31), TM3 is counted up for each input clock cycle specified by the CS2 to CS0 bits of the TMC30 register.

Division by the prescaler can be selected for the count clock from among $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/16$, $f_{CLK}/32$, $f_{CLK}/64$, $f_{CLK}/128$ and $f_{CLK}/256$ by the TMC30 register (f_{CLK}: base clock).

An overflow interrupt can be generated if the timer overflows. Also, the timer can be stopped following an overflow by setting the OST bit of the TMC31 register to 1.

Caution The count clock cannot be changed while the timer is operating.

The conditions when the TM3 register becomes 0000H are shown below.

(i) Asynchronous reset

- TM3CAE bit of TMC30 register = 0
- Reset input

(ii) Synchronous reset

- TM3CE bit of TMC30 register = 0
- The CC30 register is used as a compare register, and the TM3 and CC30 registers match when clearing the TM3 register is enabled (CCLR bit of the TMC31 register = 1)

(2) Capture/compare registers 30 and 31 (CC30 and CC31)

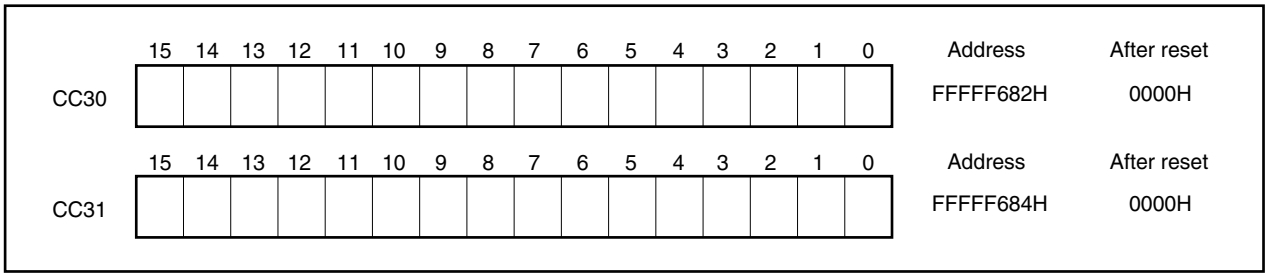
These capture/compare registers 30 and 31 are 16-bit registers.

They can be used as capture registers or compare registers according to the CMS1 and CMS0 bit specifications of timer control register 31 (TMC31).

These registers can be read/written in 16-bit units (however, write operations can only be performed in compare mode).

Caution Continuous reading of CC3n is prohibited. If CC3n is continuously read, the second read value may differ from the actual value. If CC3n must be read twice, be sure to read another register between the first and the second read operation.

Correct usage example	Incorrect usage example
CC30 read	CC30 read
CC31 read	CC30 read
CC30 read	CC31 read
CC31 read	CC31 read



(a) Setting these registers to capture registers (CMS1 and CMS0 of TMC31 = 0)

When these registers are set to capture registers, the valid edges of the corresponding external interrupt signals INTP30 and INTP31 are detected as capture triggers. The timer TM3 is synchronized with the capture trigger, and the value of TM3 is latched in the CC30 and CC31 registers (capture operation).

The valid edge of the INTP30 pin is specified (rising, falling, or both edges) according to the IES301 and IES300 bits of the SESC register, and the valid edge of the INTP31 pin is specified according to the IES311 and IES310 bits of the SESC register.

The capture operation is performed asynchronously to the count clock. The latched value is held in the capture register until the next capture operation is performed.

When the TM3CAE bit of timer control register 30 (TMC30) is 0, 0000H is read.

If these registers are specified as capture registers, an interrupt is generated by detecting the valid edge of the INTP30 and INTP31 signals.

Caution If the capture operation and the TM3 register count prohibit setting (TM3CE bit of TMC30 register = 0) timings conflict, the captured data becomes undefined, and no INTCC3n interrupt is generated (n = 0, 1).

(b) Setting these registers to compare registers (CMS1 and CMS0 of TMC31 = 1)

When these registers are set to compare registers, the TM3 and register values are compared for each count clock, and an interrupt is generated by a match. If the CCLR bit of timer control register 31 (TMC31) is set (1), the TM3 value is cleared (0) at the same time as a match with the CC30 register (it is not cleared (0) by a match with the CC31 register).

A compare register is equipped with a set/reset output function. The corresponding timer output (TO3) is set or reset, synchronized with the generation of a match signal.

The interrupt selection source differs according to the function of the selected register.

Cautions 1. To write to capture/compare registers 30 and 31 (CC30, CC31), always set the TM3CAE bit to 1 first. When the TM3CAE bit is 0, even if writing to registers CC30 and CC31, the data that is written will be invalid because the reset is asynchronous.

2. Perform a write operation to capture/compare registers 30 and 31 after setting them to compare registers according to the TMC30 or TMC31 register setting. If they are set to capture registers (CMS1 and CMS0 bits of TMC31 register = 0), no data is written even if a write operation is performed to CC30 and CC31.

3. When these registers are set to compare registers, INTP30 and INTP31 cannot be used as external interrupt input pins.

9.4.5 Control registers

(1) Timer 3 clock selection register (PRM03)

The PRM03 register is used to select the base clock (f_{CLK}) of timer 3 (TM3).

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

Cautions 1. Always set this register before using the timer.

2. Set f_{CLK} to 32 MHz or less.

	7	6	5	4	3	2	1	0	Address	After reset
PRM03	0	0	0	0	0	0	0	PRM3	FFFFF690H	00H

Bit position	Bit name	Function
0	PRM3	Specifies the base clock (f_{CLK}) of timer 3 (TM3). 0: $f_{XX}/2$ (when $f_{XX} > 32$ MHz) 1: f_{XX} (when $f_{XX} \leq 32$ MHz)

Remark f_{XX}: Internal system clock

(2) Timer control register 30 (TMC30)

The TMC30 register controls the operation of TM3.

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

Cautions 1. The TM3CAE bit and other bits cannot be set at the same time. Be sure to set the TM3CAE bit and then set the other bits and the other registers of TM3. When using an external pin related to the timer function when using timer 3, be sure to set (1) the CAE bit after setting the external pin to the control mode.

2. If occurrence of an overflow contends with writing to the TMC30 register, the value of the TM3OVF bit is the value written to the TMC30 register.

(1/2)

	<7>	6	5	4	3	2	<1>	<0>	Address	After reset
TMC30	TM3OVF	CS2	CS1	CS0	0	0	TM3CE	TM3CAE	FFFFF686H	00H

Bit position	Bit name	Function
7	TM3OVF	Flag that indicates TM3 overflow. 0: No overflow 1: Overflow The TM3OVF bit becomes 1 when TM3 changes from FFFFH to 0000H. An overflow interrupt request (INTTM3) is generated at the same time. However, if CC30 is set to the compare mode (CMS0 bit of the TMC31 register = 1) and match clear during comparison of TM3 and CC30 is enabled (CCLR bit of TMC31 register = 1), and TM3 is cleared to 0000H following match at FFFFH, TM3 is considered to have been cleared and the TM3OVF bit does not become 1, nor is the INTTM3 interrupt generated. The TM3OVF bit holds a “1” until 0 is written to it or an asynchronous reset is applied while the TM3CAE bit = 0. Interrupts by overflow and the TM3OVF bit are independent, and even if the TM3OVF bit is manipulated, this does not affect the interrupt request flag for INTTM3 (TM3IF0). If an overflow occurs while the TM3OVF bit is being read, the value of the flag changes and the value is returned at the next read.

(2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>6 to 4</td><td>CS2 to CS0</td><td>Selects the internal count clock for TM3.

<table>
<tr><th>CS2</th><th>CS1</th><th>CS0</th><th>Count clock</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>$f_{CLK}/2$</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>$f_{CLK}/4$</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>$f_{CLK}/8$</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>$f_{CLK}/16$</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>$f_{CLK}/32$</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>$f_{CLK}/64$</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>$f_{CLK}/128$</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>$f_{CLK}/256$</td></tr>
</table>

Caution Do not change the CS2 to CS0 bits during timer operation. If they are to be changed, they must be changed after setting the TM3CE bit to 0. If the CS2 to CS0 bits are overwritten during timer operation, the operation is not guaranteed.

Remark f_{CLK}: Base clock</td></tr>
<tr><td>1</td><td>TM3CE</td><td>Controls the operation of TM3.
0: Count disabled (timer stopped at 0000H and does not operate)
1: Count operation performed.

Caution If TM3CE = 0, the external pulse output (TO3) becomes inactive level (The active level of TO3 output is set with the ALV bit of the TMC31 register).</td></tr>
<tr><td>0</td><td>TM3CAE</td><td>Controls the internal count clock.
0: Entire TM3 unit asynchronously reset. Stop base clock supply to TM3 unit.
1: Base clock (f_{CLK}) supplied to TM3 unit.

Cautions 1. When TM3CAE = 0 is set, the TM3 unit can be reset asynchronously.
2. When TM3CAE = 0, the TM3 unit is in a reset state. To operate TM3, first set TM3CAE = 1.
3. When the TM3CAE bit is changed from 1 to 0, all the registers of the TM3 unit are initialized. When again setting TM3CAE = 1, be sure to then again set all the registers of the TM3 unit.</td></tr>
</table>

(3) Timer control register 31 (TMC31)

The TMC31 register controls the operation of TM3.

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

Cautions 1. Do not change the bits of the TMC31 register during timer operation. If they are to be changed, they must be changed after setting the TM3CE bit of the TMC30 register to 0. If the TMC31 register is overwritten during timer operation, the operation is not guaranteed.

2. If the ENT1 bit and the ALV bit are changed simultaneously, a glitch (spike-shaped noise) may be generated in the TO3 pin output. Either design a circuit that will not malfunction even if a glitch is generated, or make sure that the ENT1 bit and the ALV bit do not change at the same time.

3. TO3 output remains unchanged by external interrupt signals (INTP30, INTP31). When using the TO3 signal, set the capture/compare register to the compare register (CMS1, CMS0 bits of TMC31 register = 1).

Remark A reset takes precedence for the flip-flop of the TO3 output.

	7	6	5	4	3	2	1	0	Address	After reset
TMC31	OST	ENT1	ALV	ETI	CCLR	ECLR	CMS1	CMS0	FFFFF688H	20H

Bit position	Bit name	Function
7	OST	Sets the operation when TM3 overflows. 0: Count operation continues after overflow (free-running mode) 1: After overflow, timer holds 0000H and stops count operation (overflow stop mode). At this time, the TM3CE bit of TMC30 remains 1. The count operation is resumed by again writing 1 to the TM3CE bit.
6	ENT1	Enables/disables output of external pulse output (TO3). 0: Disable external pulse output. Output of inactive level of ALV bit to TO3 pin is fixed. TO3 pin level remains unchanged even if match signal from corresponding compare register is generated. 1: Enable external pulse output. Compare register match causes TO3 output to change. However, in capture mode, TO3 output does not change. An ALV bit inactive level is output from when timer output is enabled until a match signal is generated. **Caution If either CC30 or CC31 is specified as a capture register, the ENT1 bit must be set to 0.**
5	ALV	Specifies active level of external pulse output (TO3). 0: Active level is low level. 1: Active level is high level. **Caution The initial value of the ALV bit is “1”.**
4	ETI	Switches count clock between external clock and internal clock. 0: Specifies input clock (internal). The count clock can be selected with bits CS2 to CS0 of TMC30. 1: Specifies external clock (TI3). Valid edge can be selected with bits TES31, TES30 of SESC.
3	CCLR	Enables/disables TM3 clearing during compare operation. 0: Clearing disabled. 1: Clearing enabled (TM3 is cleared when CC30 and TM3 match during compare operation).
2	ECLR	Enables TM3 clearing by external clear input (TCLR3). 0: Clearing by TCLR3 disabled. 1: Clearing by TCLR3 enabled (counting resumes after clearing).
1	CMS1	Selects operation mode of capture/compare register (CC31). 0: Register operates as capture register. 1: Register operates as compare register.
0	CMS0	Selects operation mode of capture/compare register (CC30). 0: Register operates as capture register. 1: Register operates as compare register.

(4) Valid edge selection register (SESC)

This register specifies the valid edge of external interrupt requests (TI3, TCLR3, INTP30, INTP31) 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.

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

Caution Do not change the bits of the SESC register during timer operation. If they are to be changed, they must be changed after setting the TM3CE bit of the TMC30 register to 0. If the SESC register is overwritten during timer operation, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
SESC	TES31	TES30	CES31	CES30	IES311	IES310	IES301	IES300	FFFFF689H	00H
	TI3		TCLR3		INTP31		INTP30			

Bit position	Bit name	Function
7, 6	TES31, TES30	Specifies the valid edge of INTP30, INTP31 pins, TCLR3, and TI3 pins.
5, 4	CES31, CES30	
3, 2	IES311, IES310	
1, 0	IES301, IES300	

xESn1	xESn0	Operation
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Remark n = 3, 30, 31

(5) Timer 3 output control register (TO3C)

TO3C is a register that controls output of the TO3 pin.

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

Caution The TO3 output stop status can be canceled by writing 0 to the TO3SP bit of this register.

	7	6	5	4	3	2	1	<0>	Address	After reset
TO3C	0	0	0	0	0	0	0	TO3SP	FFFFF6A0H	00H

Bit position	Bit name	Function
0	TO3SP	Validates or invalidates output stop control of the TO3 pin by INTP4 pin input. 0: Invalidates INTP4 pin input (TO3 output (the output buffer of the TO3 pin is on)). 1: Validates INTP4 pin input (TO3 output is stopped by the valid edge of the INTP4 pin (the output buffer of the TO3 pin is off and the TO3 pin goes into a high-impedance state)).

The following table indicates the relationship between the setting of each register and the status of the TO3, P27, and INTP31 pins.

Table 9-14. Relationship Between Setting of Each Register and Status of TO3, P27, and INTP31 Pins

PMC27 Bit	PFC27 Bit	PM27 Bit	TO3SP Bit	TO3/P27/INTP31		
				Operation Mode of Pin	Output Buffer Status	Pin Function
0	×	0	×	Output port mode	On	Output port
0	×	1	×	Input port mode	Off	Input port
1	0	×	×	INTP31 input mode	Off	INTP31
1	1	×	0	TO3 output mode	On	TO3
1	1	×	1		On/off[Note]	TO3/Hi-Z[Note]

Note If the TO3SP bit is set to 1 in TO3 output mode (PMC27 bit = 1 and PFC27 bit = 1), the output buffer of the TO3 pin is turned off and the TO3 pin goes into a high-impedance state if the specified valid interrupt edge is generated on the INTP4 pin.

To avoid turning off the output drive by valid edge input to the INTP4 pin, be sure to clear the TO3SP bit to 0.

The valid edge of the INTP4 pin is specified by bit 0 (ES40) and bit 1 (ES41) of the INTM2 register. Specifying the valid edge of the INTP4 pin (changing the ES40 and ES41 bits) is prohibited while timer 3 is operating.

Remark ×: Don't care (does not have to be set)

9.4.6 Operation

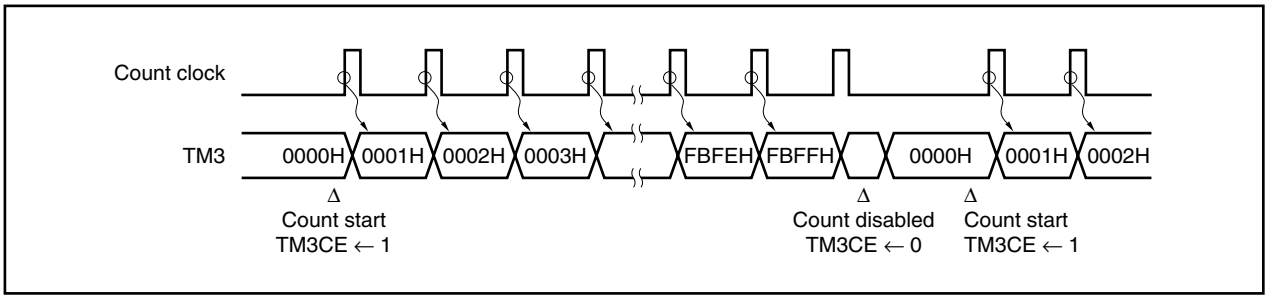
(1) Count operation

Timer 3 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 timer control register 3n (TMC3n) (n = 0, 1).

When it operates as a free-running timer, if the CC30 or CC31 register and the TM3 count value match, an interrupt signal is generated and the timer output signal (TO3) can be set or reset. Also, a capture operation that holds the TM3 count value in the CC30 or CC31 register is performed, synchronized 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 INTP30, TI3, and TCLR3 pins as TI3 and TCLR3, either mask the interrupt signal to INTP30 or set the CC3n register to compare mode (n = 0 or 1).

Figure 9-89. Basic Operation of Timer 3



(2) Overflow

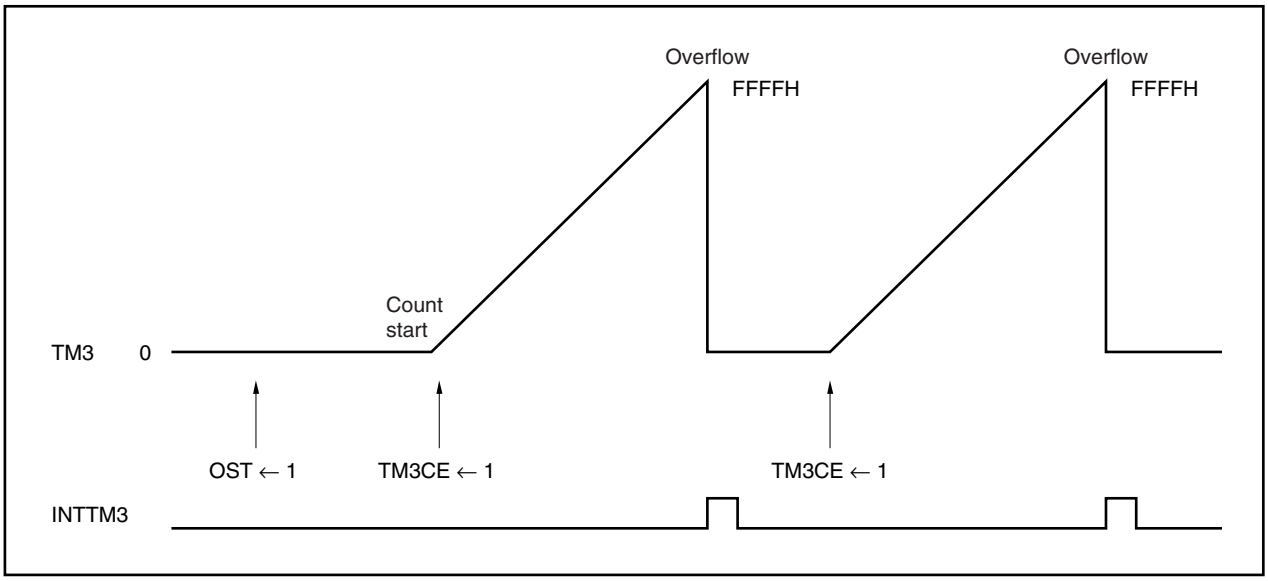
When the TM3 register has counted the count clock from FFFFH to 0000H, the TM3OVF bit of the TMC30 register is set (1), and an overflow interrupt (INTTM3) is generated at the same time. However, if the CC30 register is set to compare mode (CMS0 bit = 1) and to the value FFFFH when match clearing is enabled (CCLR bit = 1), then the TM3 register is considered to be cleared and the TM3OVF bit is not set (1) when the TM3 register changes from FFFFH to 0000H. Also, the overflow interrupt (INTTM3) is not generated .

When the TM3 register is changed from FFFFH to 0000H because the TM3CE bit changes from 1 to 0, the TM3 register is considered to be cleared, but the TM3OVF bit is not set (1) and no INTTM3 interrupt is generated.

Also, timer operation can be stopped after an overflow by setting the OST bit of the TMC31 register to 1. When the timer is stopped due to an overflow, the count operation is not restarted until the TM3CE bit of the TMC30 register is set (1).

Operation is not affected even if the TM3CE bit is set (1) during a count operation.

Figure 9-90. Operation After Overflow (When OST = 1)



(3) Capture operation

The TM3 register has two capture/compare registers. These are the CC30 register and the CC31 register. A capture operation or a compare operation is performed according to the settings of both the CMS1 and CMS0 bits of the TMC31 register. If the CMS1 and CMS0 bits of the TMC31 register are set to 0, the register operates as a capture register.

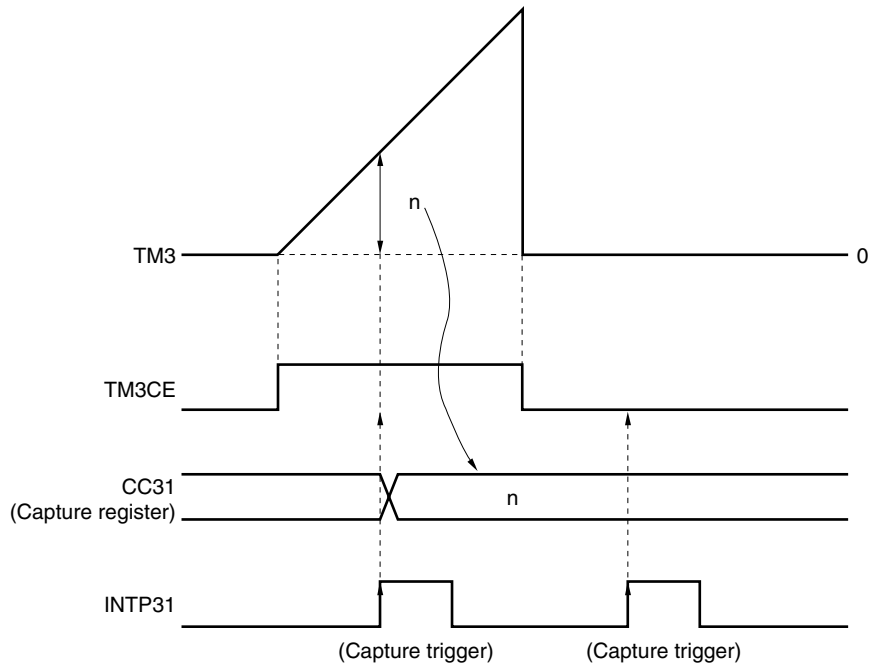
A capture operation that captures and holds the TM3 count value asynchronously relative to the count clock is performed synchronized with an external trigger. The valid edge that is detected from an external interrupt request input pin (INTP30 or INTP31) is used as an external trigger (capture trigger). The TM3 count value during counting is captured and held in the capture register, synchronized with that capture trigger signal. The capture register value is held until the next capture trigger is generated.

Also, an interrupt request (INTCC30 or INTCC31) is generated by INTP30 or INTP31 signal input.

The valid edge of the capture trigger is set by valid edge selection register (SESC).

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.

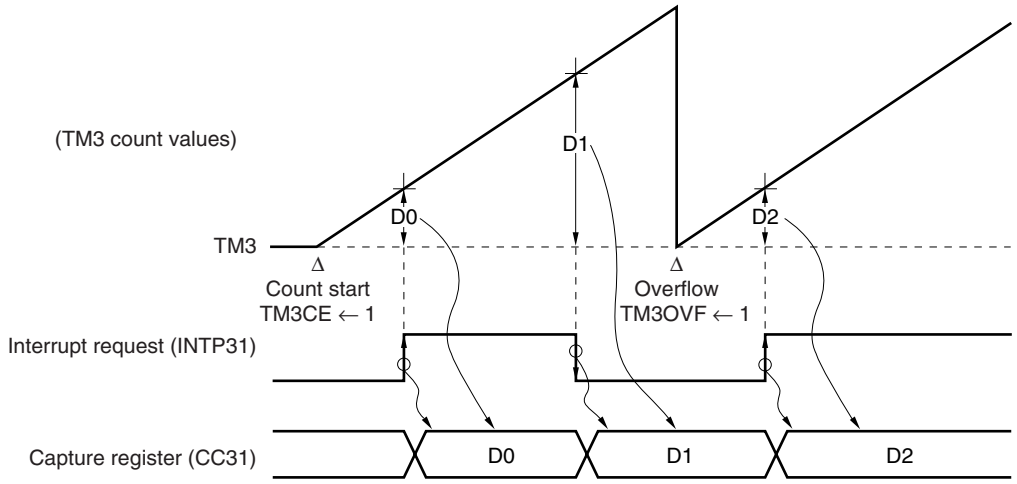
Figure 9-91. Capture Operation Example



Remarks 1. When the TM3CE bit is 0, no capture operation is performed even if INTP31 is input.

2. Valid edge of INTP31: Rising edge

Figure 9-92. TM3 Capture Operation Example (When Both Edges Are Specified)



Remark D0 to D2: TM3 count values

(4) Compare operation

The TM3 register has two capture/compare registers. These are the CC30 register and the CC31 register. A capture operation or a compare operation is performed according to the settings of both the CMS1 and CMS0 bits of the TMC31 register. If 1 is set in the CMS1 and CMS0 bits of the TMC31 register, the register operates as a compare register.

A compare operation that compares the value that was set in the compare register and the TM3 count value is performed.

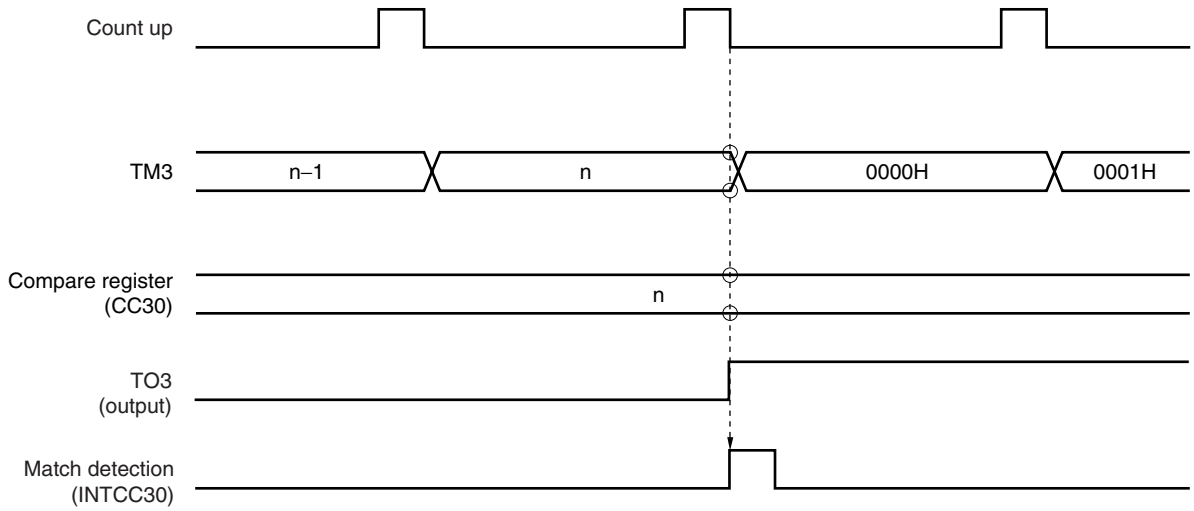
If the TM3 count value matches the value of the compare register, which had been set in advance, a match signal is sent to the output controller. The match signal causes the timer output pin (TO3) to change and an interrupt request signal (INTCC30, INTCC31) to be generated at the same time.

If the CC30 or CC31 register is set to 0000H, "0000H" after the TM3 register counts up from FFFFH to 0000H is judged as a match. In this case, the value of the TM3 register is cleared to 0 at the next count timing, but 0000H is not judged as a match at that time. 0000H when the TM3 register begins counting is not judged as a match either.

If match clearing is enabled (CCLR bit = 1) for the CC30 register, the TM3 register is cleared when a match with the TM3 register occurs during a compare operation.

Figure 9-93. Compare Operation Example (1/2)

(a) If CCLR bit = 1 and CC30 register is value other than 0000H

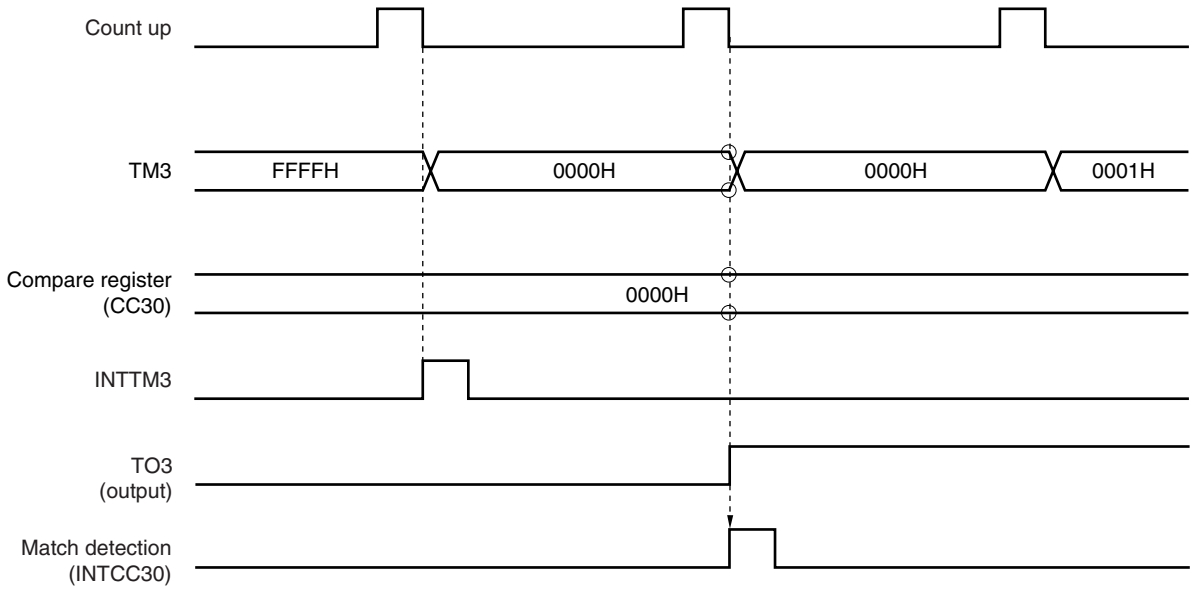


Remarks 1. The match is detected immediately after the count up, and the match detection signal is generated.

2. n ≠ 0000H

Figure 9-93. Compare Operation Example (2/2)

(b) If CCLR bit = 1 and CC30 register is 0000H



Remark The match is detected immediately after the count up, and the match detection signal is generated.

(5) External pulse output

Timer 3 has one timer output pin (TO3).

An external pulse output (TO3) is generated when a match of the two compare registers (CC30 and CC31) and the TM3 register is detected.

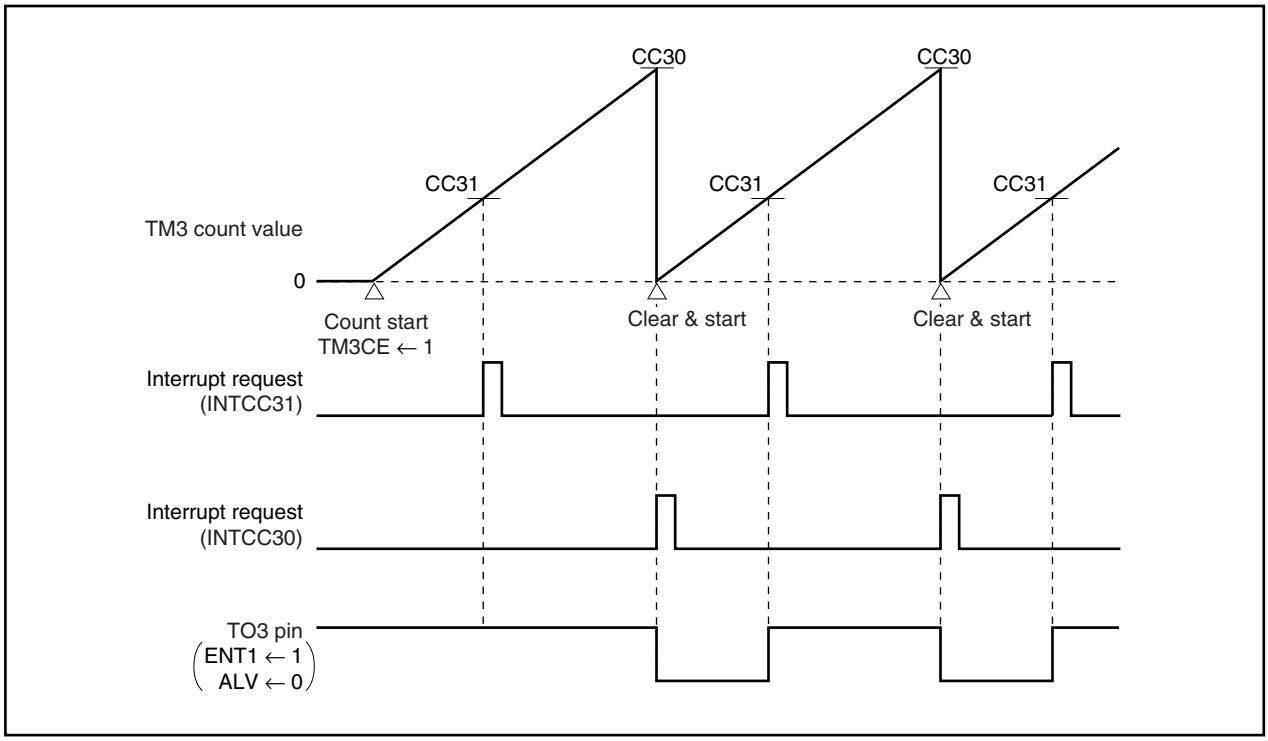
If a match is detected when the TM3 count value and the CC30 value are compared, the output level of the TO3 pin is set. Also, if a match is detected when the TM3 count value and the CC31 value are compared, the output level of the TO3 pin is reset.

The output level of the TO3 pin can be specified by the TMC31 register.

Table 9-15. TO3 Output Control

ENT1	ALV	TO3 Output	
		External Pulse Output	Output Level
0	0	Disable	High level
0	1	Disable	Low level
1	0	Enable	When the CC30 register is matched: Low level When the CC31 register is matched: High level
1	1	Enable	When the CC30 register is matched: High level When the CC31 register is matched: Low level

Figure 9-94. TM3 Compare Operation Example (Set/Reset Output Mode)



(6) TO3 output control function by INTP4 pin

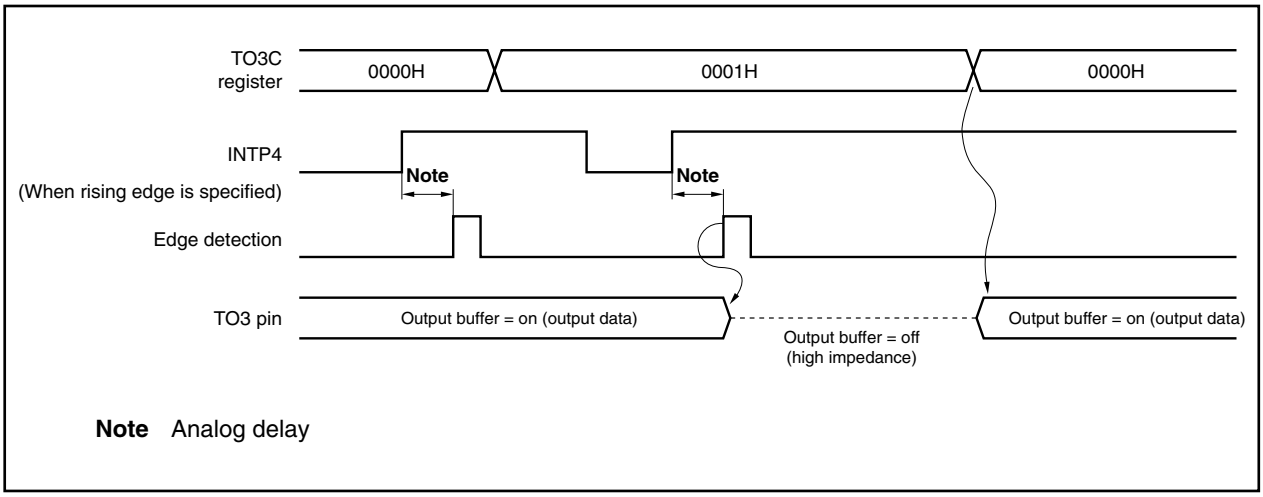
Output of the TO3 pin can be forcibly stopped by inputting a signal to the INTP4 pin if an abnormality is detected in the power system of a motor.

If the TO3 output mode is set (PMC27 = 1 and PFC27 = 1) and if the specified valid edge is generated on the INTP4 pin after the TO3SP bit of the timer 3 output control register (TO3C) has been set to 1, the output buffer of the TO3 pin can be turned off (the TO3 pin goes into a high-impedance state).

To resume output of the TO3 pin (output buffer = on) after output of the TO3 pin has been stopped (output buffer = off) by the valid edge of the INTP4 pin, rewrite the TO3SP bit from “1” to “0”.

The valid edge of the INTP4 pin can be specified by the ES40 and ES41 bits of the external interrupt mode register 2 (INTM2).

Figure 9-95. Example of Operation of TO3 Output Control Function by INTP4 Pin (in TO3 Output Mode (PMC27 Bit = 1 and PFC27 Bit = 1))



Note Analog delay

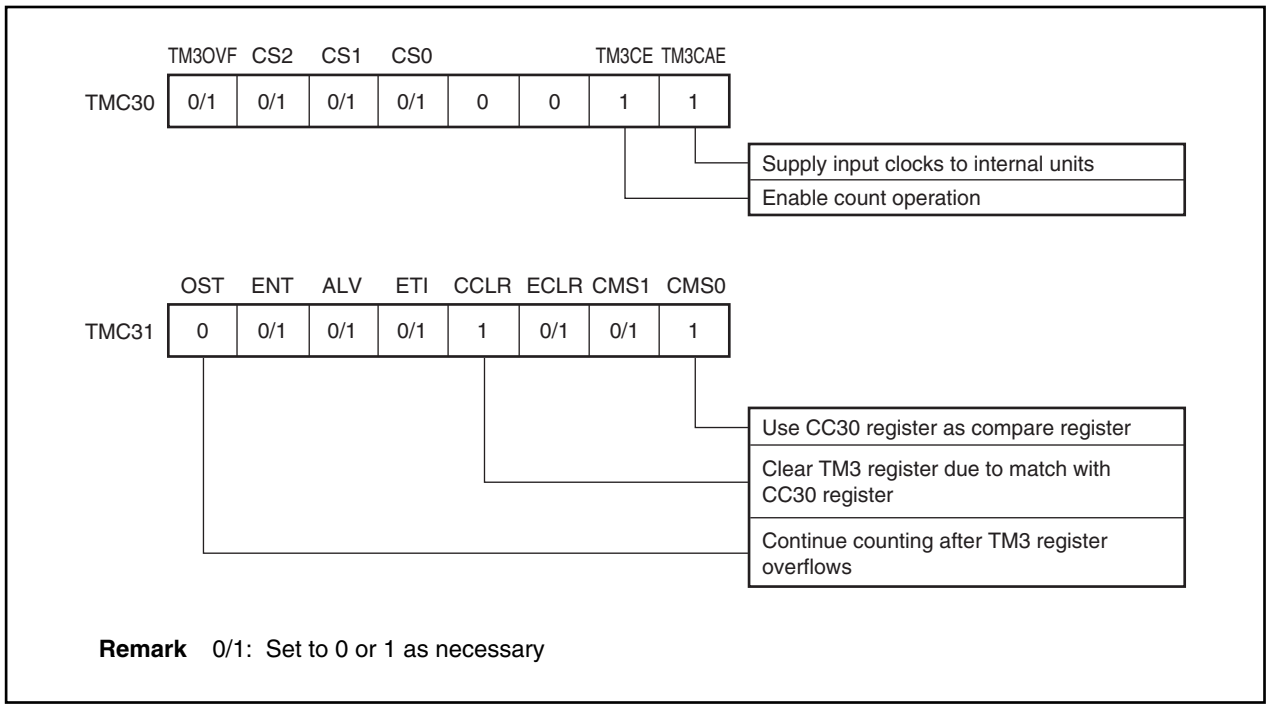
9.4.7 Application examples

(1) Interval timer

By setting the TMC30 and TMC31 registers as shown in Figure 9-96, timer 3 operates as an interval timer that repeatedly generates interrupt requests with the value that was set in advance in the CC30 register as the interval.

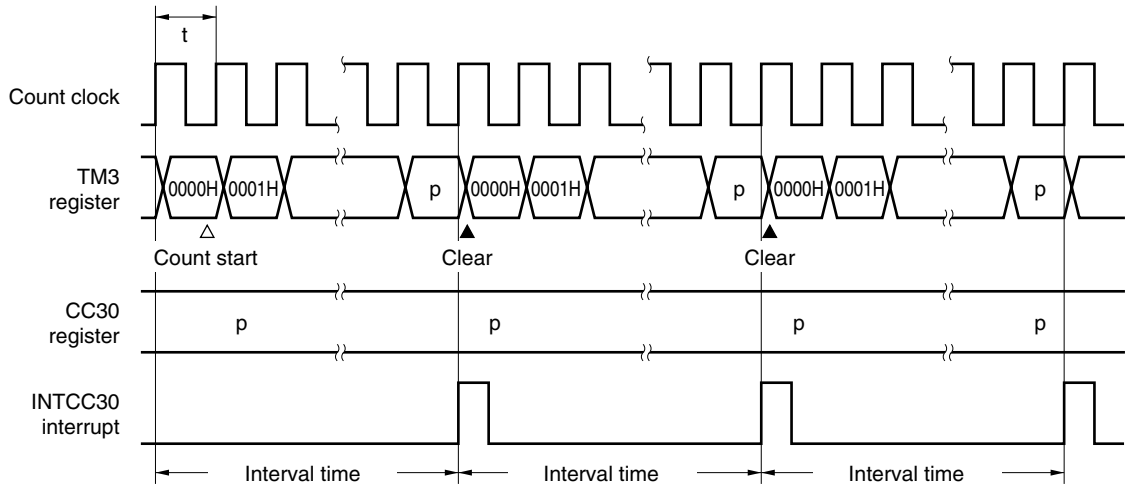
When the counter value of the TM3 register matches the setting value of the CC30 register, the TM3 register is cleared (0000H) and an interrupt request signal (INTCC30) is generated at the same time that the count operation resumes.

Figure 9-96. Contents of Register Settings When Timer 3 Is Used as Interval Timer



Remark 0/1: Set to 0 or 1 as necessary

Figure 9-97. Interval Timer Operation Timing Example



Remark p: Setting value of CC30 register (0000H to FFFFH)

t: Count clock cycle

Interval time = $(p + 1) \times t$

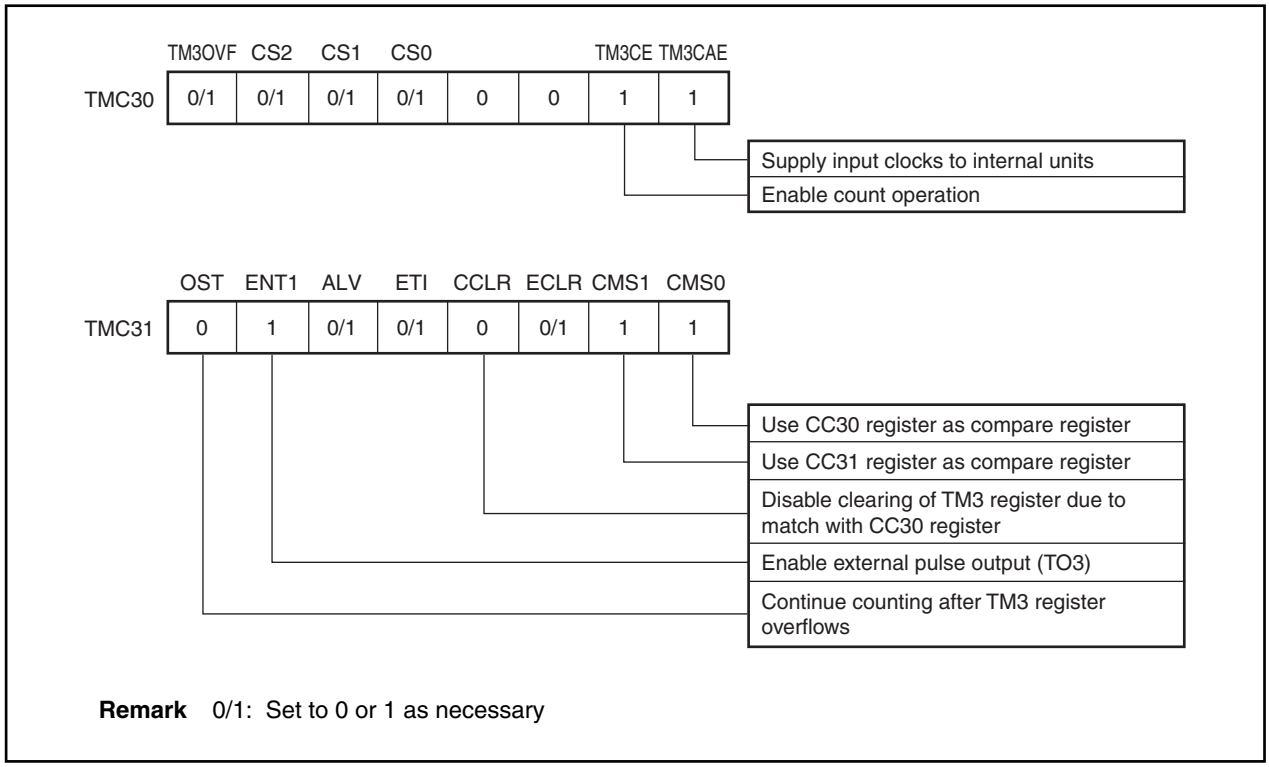
(2) PWM output

By setting the TMC30 and TMC31 registers as shown in Figure 9-98, timer 3 can output a PWM of the frequency determined by the setting of the CS2 to CS0 bits of the TMC30 register with the values that were set in advance in the CC30 and CC31 registers as the intervals.

When the counter value of the TM3 register matches the setting value of the CC30 register, the TO3 output becomes active. Then, when the counter value of the TM3 register matches the setting value of the CC31 register, the TO3 output becomes inactive. The TM3 register continues counting, and when an overflow occurs, clears the count value to 0000H and continues counting. This enables a PWM of the frequency determined by the setting of the CS2 to CS0 bits of the TMC30 register to be output. When the setting value of the CC30 register and the setting value of the CC31 register are the same, the TO3 output remains inactive and does not change.

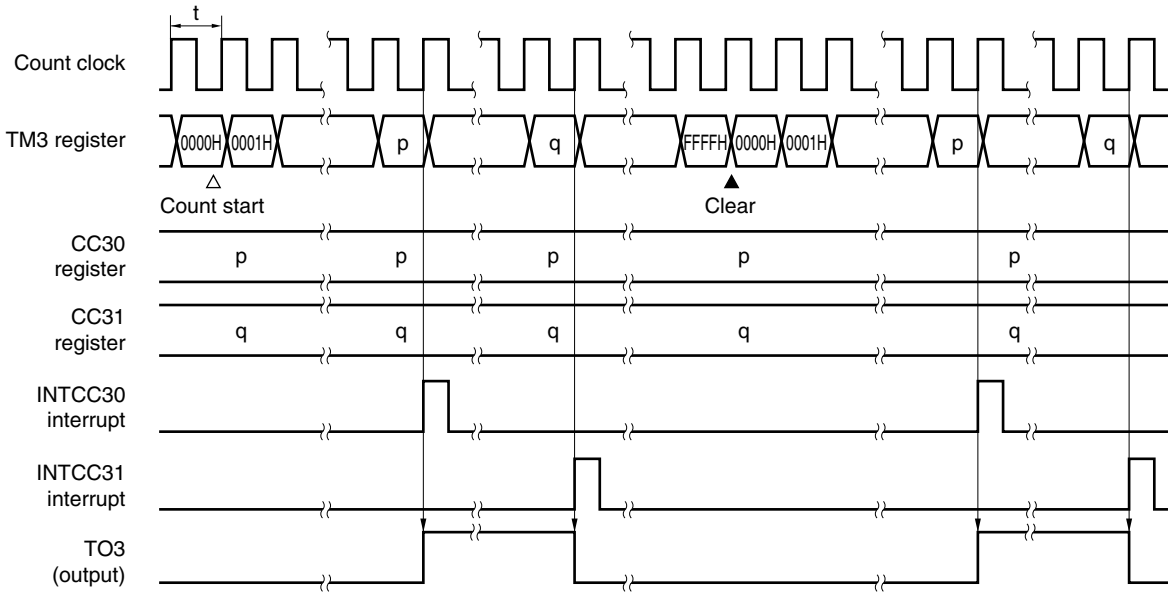
The active level of TO3 output can be set by the ALV bit of the TMC31 register.

Figure 9-98. Contents of Register Settings When Timer 3 Is Used for PWM Output



Remark 0/1: Set to 0 or 1 as necessary

Figure 9-99. PWM Output Operation Timing Example



Remarks 1. p: Setting value of CC30 register (0000H to FFFFH)

q: Setting value of CC31 register (0000H to FFFFH)

$p \neq q$

t: Count clock cycle

PWM cycle = $65536 \times t$

$$\text{Duty} = \frac{q - p}{65536}$$

2. In this example, the active level of TO3 output is set to high level.

(3) Cycle measurement

By setting the TMC30 and TMC31 registers as shown in Figure 9-100, timer 3 can measure the cycle of signals input to the INTP30 pin or INTP31 pin.

The valid edge of the INTP30 pin is selected according to the IES301 and IES300 bits of the SESC register, and the valid edge of the INTP31 pin is selected according to the IES311 and IES310 bits of the SESC register. Either the rising edge, the falling edge, or both edges can be selected as the valid edges of both pins.

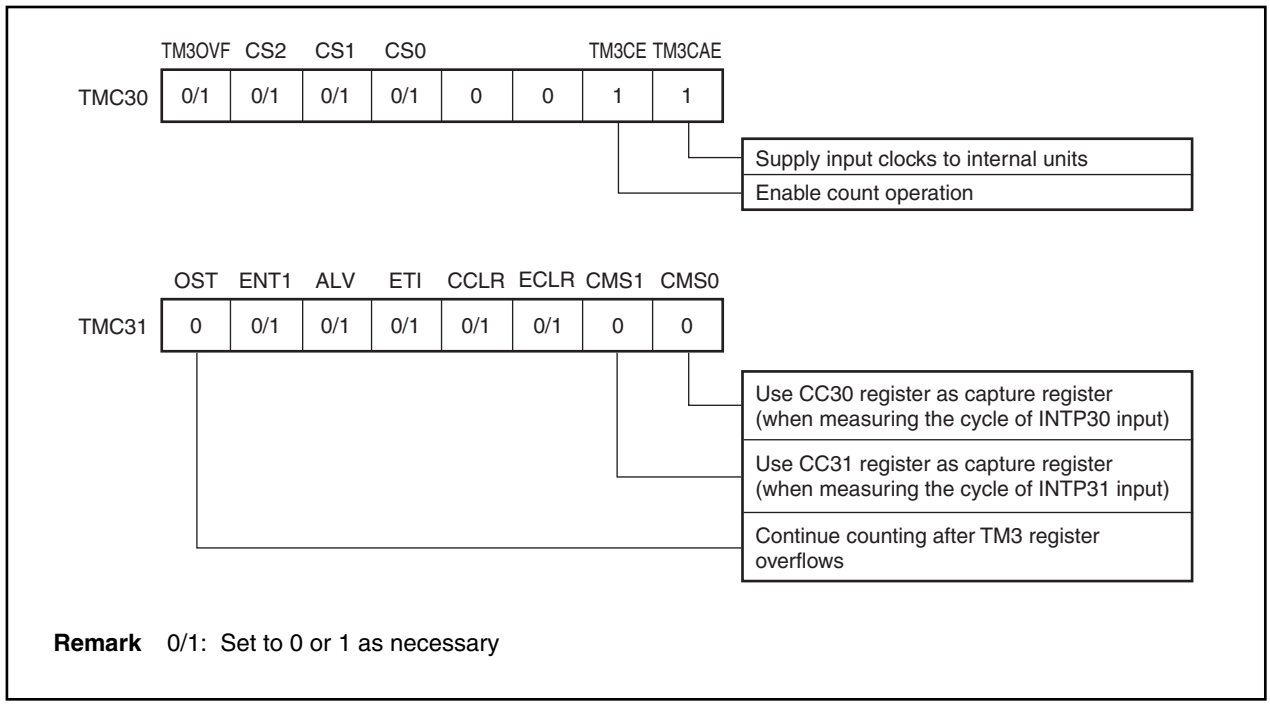
If the CC30 register is set to a capture register and TM3 is started, the valid edge input of the INTP30 pin is set as the trigger for capturing the TM3 register value in the CC30 register. When this value is captured, an INTCC30 interrupt is generated.

Similarly, if the CC31 register is set to a capture register and TM3 is started, the valid edge input of the INTP31 pin is set as the trigger for capturing the TM3 register value in the CC31 register. When this value is captured, an INTCC31 interrupt is generated.

The cycle of signals input to the INTP30 pin is calculated by obtaining the difference between the TM3 register's count value (Dx) that was captured in the CC30 register according to the x-th valid edge input of the INTP30 pin and the TM3 register's count value (D(x+1)) that was captured in the CC30 register according to the (x+1)-th valid edge input of the INTP30 pin and multiplying the value of this difference by the cycle of the clock control signal.

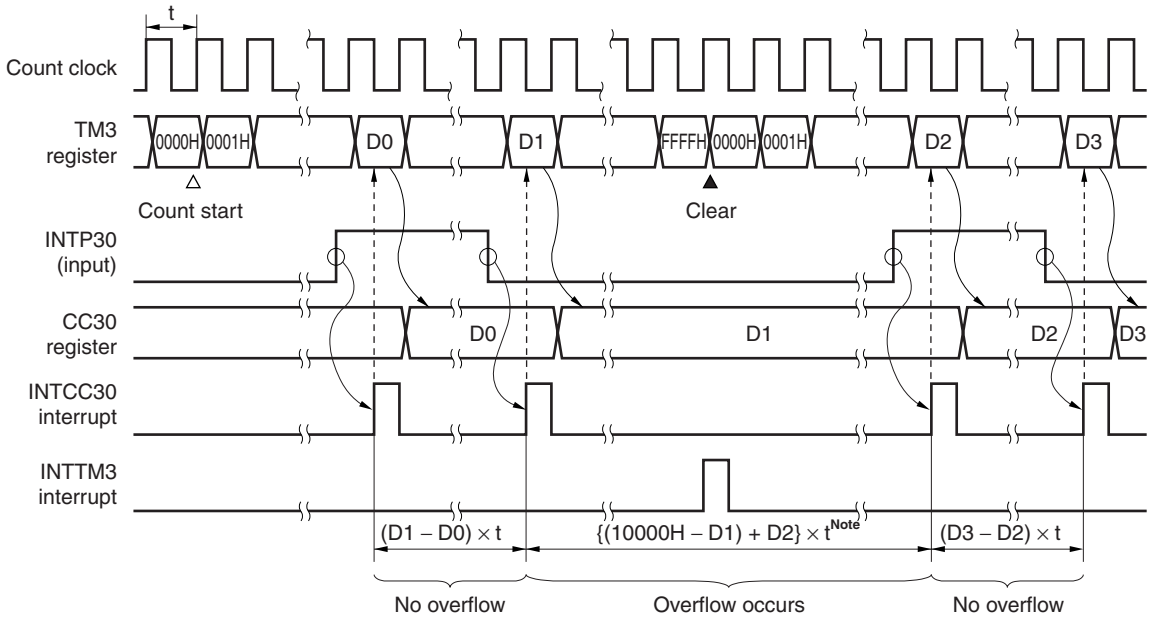
The cycle of signals input to the INTP31 pin is calculated by obtaining the difference between the TM3 register's count value (Dx) that was captured in the CC31 register according to the x-th valid edge input of the INTP31 pin and the TM3 register's count value (D(x+1)) that was captured in the CC31 register according to the (x+1)-th valid edge input of the INTP31 pin and multiplying the value of this difference by the cycle of the clock control signal.

Figure 9-100. Contents of Register Settings When Timer 3 Is Used for Cycle Measurement



Remark 0/1: Set to 0 or 1 as necessary

Figure 9-101. Cycle Measurement Operation Timing Example



Note When an overflow occurs once.

Remarks 1. D0 to D3: TM3 register count values
t: Count clock cycle

2. In this example, the valid edge of INTP30 input has been set to both edges (rising and falling).

9.4.8 Cautions

Various cautions concerning timer 3 are shown below.

(1) If a conflict occurs between the reading of the CC30 register and a capture operation when the CC30 register is used in capture mode, an external trigger (INTP30) valid edge is detected and an external interrupt request signal (INTCC30) is generated, but the timer value is not stored in the CC30 register.

(2) If a conflict occurs between the reading of the CC31 register and a capture operation when the CC31 register is used in capture mode, an external trigger (INTP31) valid edge is detected and an external interrupt request signal (INTCC31) is generated, but the timer value is not stored in the CC31 register.

(3) The following bits and registers must not be rewritten during operation (TMC30 register TM3CE = 1).
- CS2 to CS0 bits of TMC30 register
- TMC31 register
- SESC register

(4) The TM3CAE bit of the TMC30 register is a TM3 reset signal. To use TM3, first set (1) the TM3CAE bit.

(5) The analog noise elimination time + two count clock cycles are required to detect a valid edge of the external interrupt input (INTP30 or INTP31) and external clock input (TI3). Therefore, edge detection will not be performed normally for changes that are less than the analog noise elimination time + two count clock cycles. For the analog noise elimination, refer to **12.5 Noise Eliminator**.

(6) The operation of an external interrupt output (INTCC30 or INTCC31) is automatically determined according to the operating state of the capture/compare registers 30, 31 (CC30, CC31). When the capture/compare register is used for a capture mode, the external trigger (INTP30, INTP31) is used for valid edge detection. When the capture/compare register is used for a compare mode, the external interrupt output is used for a match interrupt indicating a match with the TM3 register.

(7) If the ENT1 and ALV bits of the TMC31 register are changed at the same time, a glitch (spike shaped noise) may be generated in the TO3 pin output. Either create a circuit configuration that will not malfunction even if a glitch is generated or make sure that the ENT1 and ALV bits do not change at the same time.

9.5 Timer 4

9.5.1 Features (timer 4)

Timer 4 (TM4) functions as a 16-bit interval timer.

9.5.2 Function overview (timer 4)

- 16-bit interval timer: 1 channel
- Compare register: 1
- Count clock selected from divisions of internal system clock (set the frequency of the count clock to 16 MHz or less)
- Base clock (f_{CLK}): 1 type (set f_{CLK} to 32 MHz or less)
  $f_{XX}/2$
- Prescaler division ratio
  The following division ratios can be selected according to the base clock (f_{CLK}).

Division Ratio	Base Clock (f_{CLK})
1/2	$f_{XX}/4$
1/4	$f_{XX}/8$
1/8	$f_{XX}/16$
1/16	$f_{XX}/32$
1/32	$f_{XX}/64$
1/64	$f_{XX}/128$
1/128	$f_{XX}/256$
1/256	$f_{XX}/512$

- Interrupt request source: 1
  - Compare match interrupt
    INTCM4 generated by CM4 match signal
- Timer clear
  The TM4 register can be cleared by a CM4 register match.

Remark f_{XX}: Internal system clock

9.5.3 Basic configuration

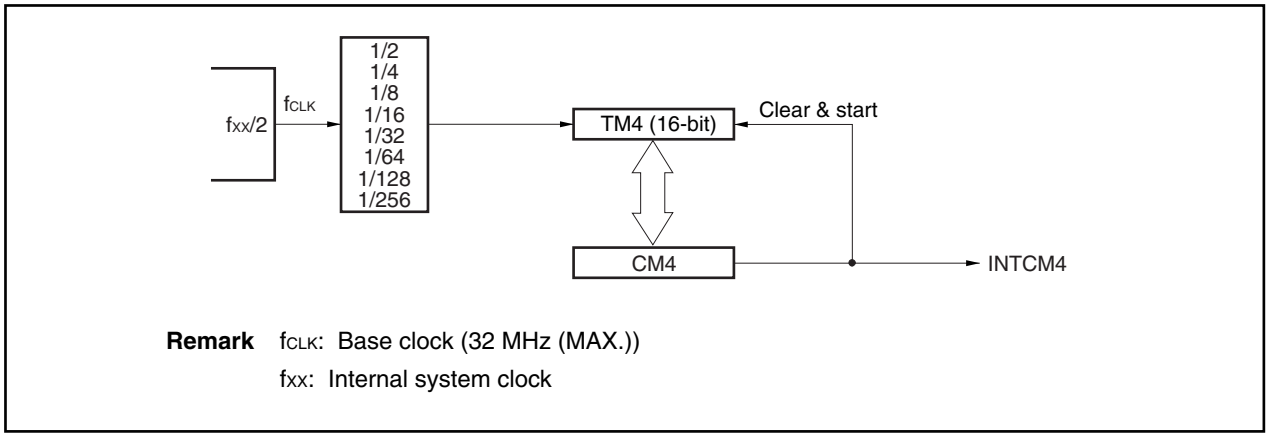
Table 9-16. Timer 4 Configuration List

Timer	Count Clock	Register	Read/Write	Generated Interrupt Signal	Capture Trigger	Timer Output S/R	Other Functions
Timer 4	$f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$, $f_{XX}/256$, $f_{XX}/512$	TM4	Read	–	–	–	–
		CM4	Read/write	INTCM4	–	–	–

Remark f_{XX}: Internal system clock
S/R: Set/Reset

Figure 9-102 shows the block diagram of timer 4.

Figure 9-102. Block Diagram of Timer 4



Remark f_{CLK}: Base clock (32 MHz (MAX.))
f_{XX}: Internal system clock

(1) Timer 4 (TM4)

TM4 is a 16-bit timer. It is mainly used as an interval timer for software.

Starting and stopping TM4 is controlled by the TM4CE0 bit of timer control register 4 (TMC4).

Division by the prescaler can be selected for the count clock from among fxx/4, fxx/8, fxx/16, fxx/32, fxx/64, fxx/128, fxx/256, and fxx/512 by the CS2 to CS0 bits of the TMC4 register (fxx: Internal system clock).

TM4 is read-only, in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
TM4																	FFFFF540H	0000H

The conditions under which the TM4 register becomes 0000H are shown below.

- Reset input
- TM4CAE0 bit = 0
- TM4CE0 bit = 0
- Match of TM4 register and CM4 register
- Overflow

Cautions 1. If the TM4CAE0 bit of the TMC4 register is cleared (0), a reset is performed asynchronously.

2. If the TM4CE0 bit of the TMC4 register is cleared (0), a reset is performed, synchronized with the internal clock. Similarly, a synchronized reset is performed after a match with the CM4 register and after an overflow.

3. The count clock must not be changed during a timer operation. If it is to be overwritten, it should be overwritten after the TM4CE0 bit is cleared (0).

4. Up to 4 internal system clocks are required after a value is set in the TM4CE0 bit until the set value is transferred to internal units. When a count operation begins, the count cycle from 0000H to 0001H differs from subsequent count cycles.

5. After a compare match is generated, the timer is cleared at the next count clock. Therefore, if the division ratio is large, the timer value may not be zero even if the timer value is read immediately after a match interrupt is generated.

(2) Compare register 4 (CM4)

CM4 and the TM4 register count value are compared, and an interrupt request signal (INTCM4) is generated when a match occurs. TM4 is cleared, synchronized with this match. If the TM4CAE0 bit of the TMC4 register is set to 0, a reset is performed asynchronously, and the registers are initialized.

The CM4 register has a master/slave configuration. When a write operation to a CM4 register is performed, data is first written to the master register and then the master register data is transferred to the slave register. In a compare operation, the slave register value is compared with the count value of the TM4 register. When a read operation to the CM4 register is performed, data on the master side is read out.

CM4 can be read/written in 16-bit units.

Cautions 1. A write operation to the CM4 register requires 4 internal system clocks until the value that was set in the CM4 register is transferred to internal units. When writing continuously to the CM4 register, be sure to reserve a time interval of at least 4 internal system clocks.

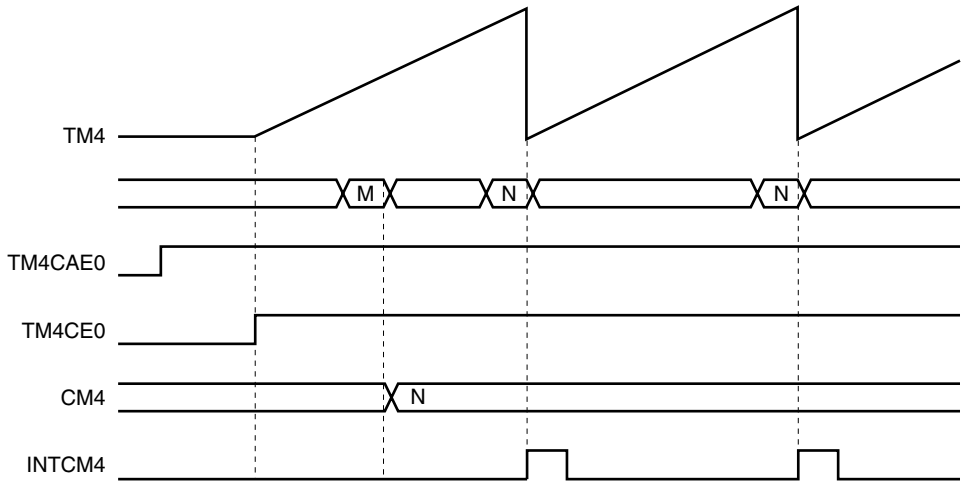
2. The CM4 register can be overwritten only once in a single TM4 register cycle (from 0000H until an INTCM4 interrupt is generated due to a match of the TM4 register and CM4 register). If this cannot be secured by the application, make sure that the CM4 register is not overwritten during timer operation.

3. Note that an INTCM4 interrupt will be generated after an overflow if a value less than the counter value is written in the CM4 register during TM4 register operation (Figure 9-103).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CM4																	FFFFF542H	0000H

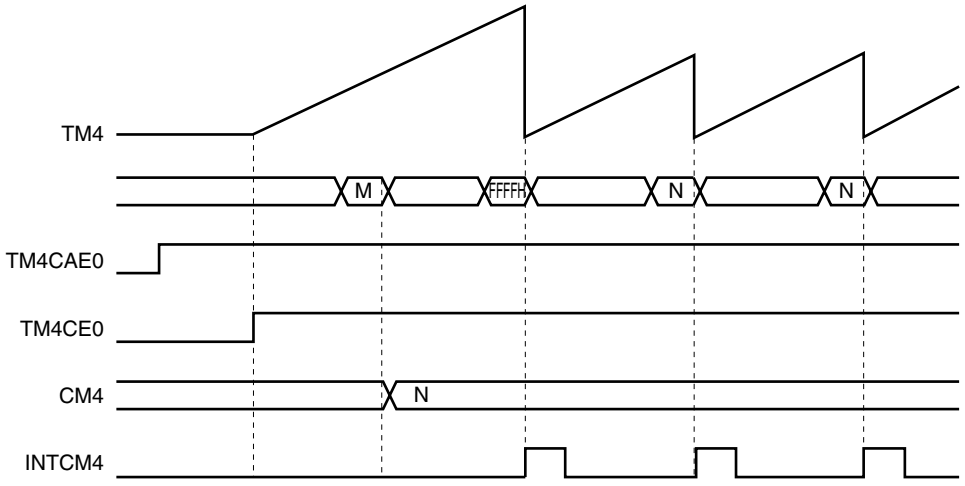
Figure 9-103. Example of Timing During TM4 Operation

(a) When TM4 < CM4



Remark M = TM4 value when overwritten
N = CM4 value after overwrite
M < N

(b) When TM4 > CM4



Remark M = TM4 value when overwritten
N = CM4 value after overwrite
M > N

9.5.4 Control register

(1) Timer control register 4 (TMC4)

The TMC4 register controls the operation of timer 4.

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

Caution The TM4CAE0 bit and other bits cannot be set at the same time. Be sure to set the TM4CAE0 bit and then set the other bits and the other registers of TM4.

	7	6	5	4	3	2	<1>	<0>	Address	After reset
TMC4	0	CS2	CS1	CS0	0	0	TM4CE0	TM4CAE0	FFFFF544H	00H

Bit position	Bit name	Function
6 to 4	CS2 to CS0	Selects the TM4 count clock. CS2 / CS1 / CS0 / Count clock 0 / 0 / 0 / f_{XX}/4 0 / 0 / 1 / f_{XX}/8 0 / 1 / 0 / f_{XX}/16 0 / 1 / 1 / f_{XX}/32 1 / 0 / 0 / f_{XX}/64 1 / 0 / 1 / f_{XX}/128 1 / 1 / 0 / f_{XX}/256 1 / 1 / 1 / f_{XX}/512 **Caution Do not change the CS2 to CS0 bits during timer operation. If they are to be changed, they must be changed after setting the TM4CE0 bit to 0. If the CS2 to CS0 bits are overwritten during timer operation, the operation is not guaranteed.**
1	TM4CE0	Controls the operation of TM4. 0: Count disabled (timer stopped at 0000H and does not operate) 1: Count operation performed **Caution The TM4CE0 bit is not cleared even if a match is detected by the compare operation. To stop the count operation, clear the TM4CE0 bit.**
0	TM4CAE0	Controls the internal count clock. 0: Entire TM4 unit asynchronously reset. Base clock (f_{CLK}) supply to TM4 unit stopped. 1: Base clock (f_{CLK}) supplied to TM4 unit. **Cautions 1. When TM4CAE0 = 0 is set, the TM4 unit can be reset asynchronously.** **2. When TM4CAE0 = 0, the TM4 unit is in a reset state. To operate TM4, first set TM4CAE0 = 1.** **3. When the TM4CAE0 bit is changed from 1 to 0, all the registers of the TM4 unit are initialized. When again setting TM4CAE0 = 1, be sure to then set all the registers of the TM4 unit again.**

9.5.5 Operation

(1) Compare operation

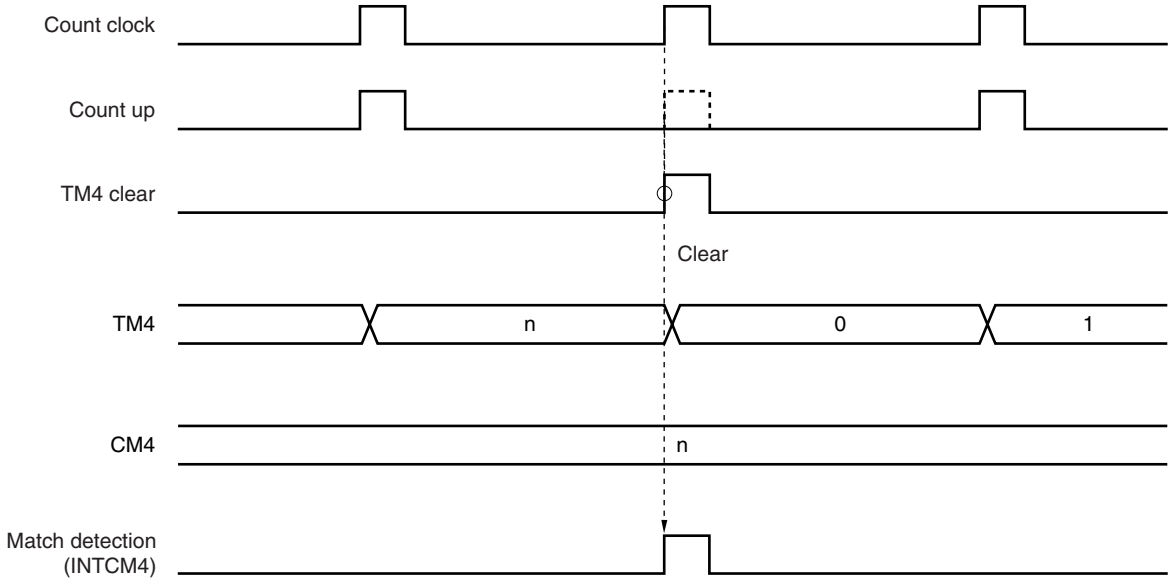
TM4 can be used for a compare operation in which the value that was set in the compare register (CM4) is compared with the TM4 count value.

If a match is detected by the compare operation, an interrupt (INTCM4) is generated. The generation of the interrupt causes TM4 to be cleared (0) at the next count timing. This function enables timer 4 to be used as an interval timer.

CM4 can also be set to 0. In this case, when an overflow occurs and TM4 becomes 0, a match is detected and INTCM4 is generated. Although the TM4 value is cleared (0) at the next count timing, INTCM4 is not generated by this match.

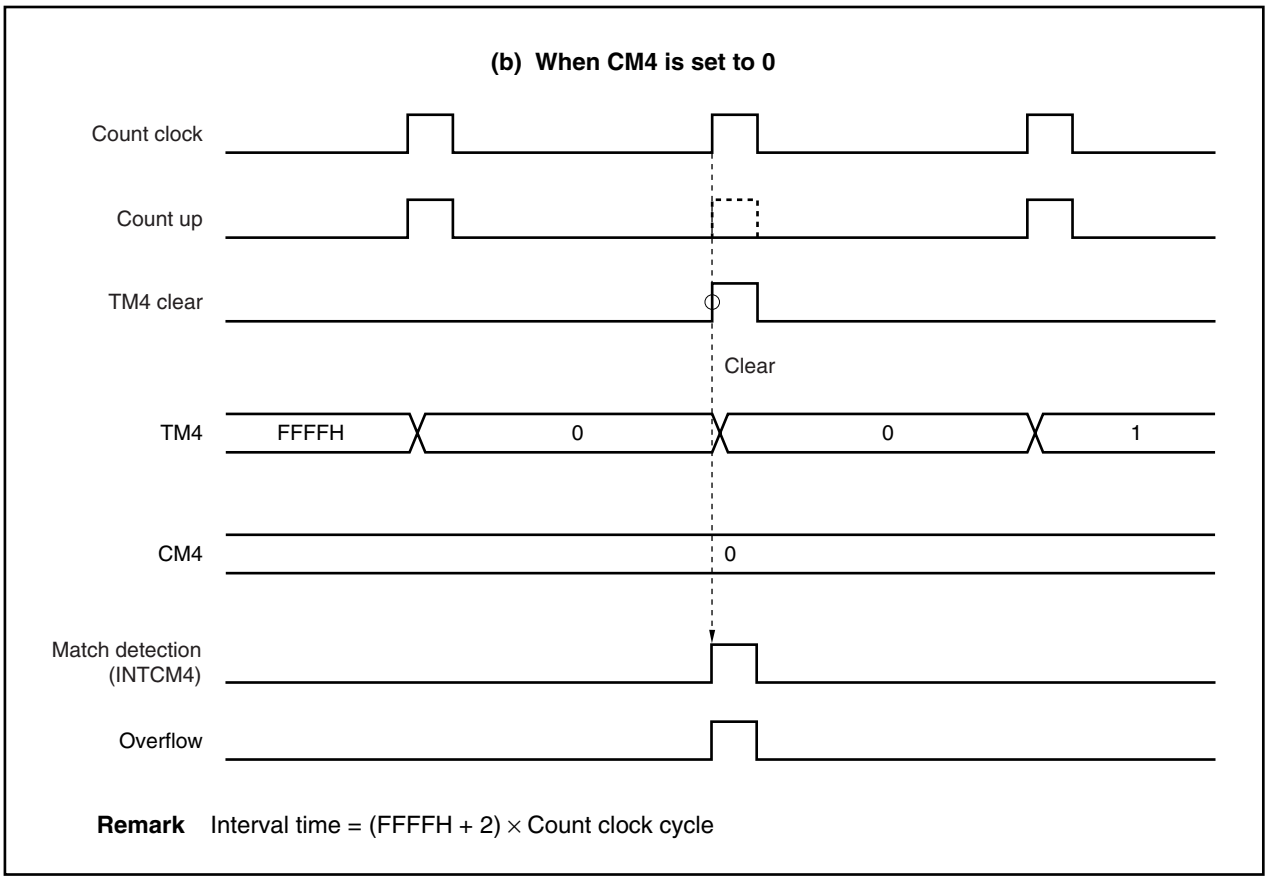
Figure 9-104. TM4 Compare Operation Example (1/2)

(a) When CM4 is set to n (non-zero)



Remark Interval time = (n + 1) × Count clock cycle
n = 1 to 65536 (FFFFH)

Figure 9-104. TM4 Compare Operation Example (2/2)



Remark Interval time = (FFFFH + 2) × Count clock cycle

9.5.6 Application example

(1) Interval timer

This section explains an example in which timer 4 is used as an interval timer with 16-bit precision.

Interrupt requests (INTCM4) are output at equal intervals (refer to **Figure 9-104 TM4 Compare Operation Example**). The setting procedure is shown below.

<1> Set (1) the TM4CAE0 bit.

<2> Set each register.

- Select the count clock using the CS2 to CS0 bits of the TMC4 register.
- Set the compare value in the CM4 register.

<3> Start counting by setting (1) the TM4CE0 bit.

<4> If the TM4 register and CM4 register values match, the INTCM4 interrupt is generated.

<5> INTCM4 interrupts are generated thereafter at equal intervals.

9.5.7 Cautions

Various cautions concerning timer 4 are shown below.

(1) To operate TM4, first set (1) the TM4CAE0 bit of the TMC4 register.

(2) Up to 4 internal system clocks are required after a value is set in the TM4CE0 bit of the TMC4 register until the set value is transferred to internal units. When a count operation begins, the count cycle from 0000H to 0001H differs from subsequent count cycles.

(3) To initialize the TM4 register status and start counting again, clear (0) the TM4CE0 bit and then set (1) the TM4CE0 bit after an interval of 4 internal system clocks has elapsed.

(4) Up to 4 internal system clocks are required until the value that was set in the CM4 register is transferred to internal units. When writing continuously to the CM4 register, be sure to secure a time interval of at least 4 internal system clocks.

(5) The CM4 register can be overwritten only once during a timer/counter operation (from 0000H until the INTCM4 interrupt is generated due to a match of the TM4 register and CM4 register). If this cannot be secured, make sure that the CM4 register is not overwritten during a timer/counter operation.

(6) The count clock must not be changed during a timer operation. If it is to be overwritten, it should be overwritten after the TM4CE0 bit is cleared (0). If the count clock is overwritten during a timer operation, operation cannot be guaranteed.

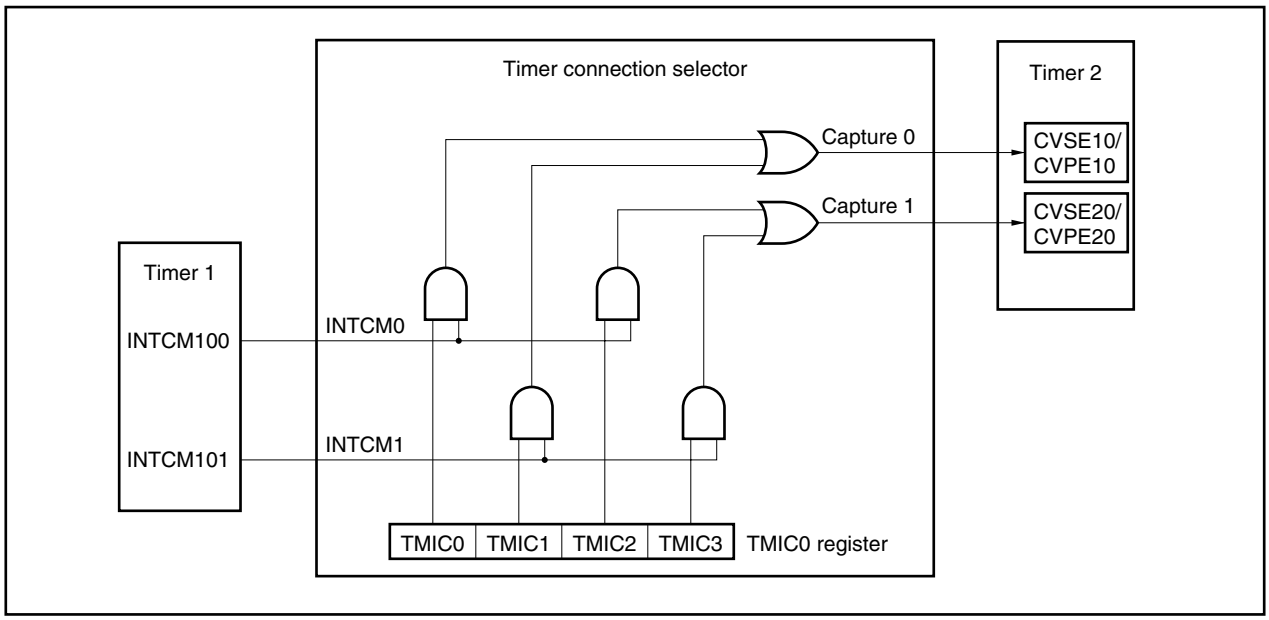
(7) An INTCM4 interrupt will be generated after an overflow if a value less than the counter value is written in the CM4 register during TM4 register operation.

9.6 Timer Connection Function

9.6.1 Overview

The V850E/IA2 provides a function to connect timer 1 and timer 2.

Figure 9-105. Block Diagram of Timer Connection Function



9.6.2 Control register

(1) Timer connection selection register 0 (TMIC0)

The TMIC0 register enables/disables input of the INTCM100 and INTCM101 signals to the CVSEn0/CVPEn0 registers (n = 1, 2).

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

	7	6	5	4	3	2	1	0	Address	After reset
TMIC0	0	0	0	0	TMIC3	TMIC2	TMIC1	TMIC0	FFFFF620H	00H

Bit position	Bit name	Function
3	TMIC3	Enables/disables input of INTCM101 signal to CVSE20/CVPE20 registers. 0: INTCM101 signal not input to CVSE20/CVPE20 registers. 1: INTCM101 signal input to CVSE20/CVPE20 registers.
2	TMIC2	Enables/disables input of INTCM100 signal to CVSE20/CVPE20 registers. 0: INTCM100 signal not input to CVSE20/CVPE20 registers. 1: INTCM100 signal input to CVSE20/CVPE20 registers.
1	TMIC1	Enables/disables input of INTCM101 signal to CVSE10/CVPE10 registers. 0: INTCM101 signal not input to CVSE10/CVPE10 registers. 1: INTCM101 signal input to CVSE10/CVPE10 registers.
0	TMIC0	Enables/disables input of INTCM100 signal to CVSE10/CVPE10 registers. 0: INTCM100 signal not input to CVSE10/CVPE10 registers. 1: INTCM100 signal input to CVSE10/CVPE10 registers.

CHAPTER 10 SERIAL INTERFACE FUNCTION

10.1 Features

The serial interface function provides two types of serial interfaces combining a total of four transmit/receive channels. Three of these channels can be used simultaneously.

The two interface formats are as follows.

(1) Asynchronous serial interfaces (UART0, UART1): 2 channels
(2) Clocked serial interfaces (CSI0, CSI1): 2 channels

UART0, UART1, in which one byte of serial data is transmitted/received following a start bit, support full-duplex communication. In the UART1 interface, one higher bit is added to 8 bits of transmit/receive data, enabling communication using 9-bit data.

CSI0 and CSI1 perform data transfer according to three types of signals: serial clocks ($\overline{SCK0}$, $\overline{SCK1}$), serial inputs (SI0, SI1), and serial outputs (SO0, SO1) (3-wire serial I/O).

10.1.1 Selecting UART1 or CSI1 mode

UART1 and CSI1 of the V850E/IA2 share pins, and therefore these interfaces cannot be used at the same time. Select UART1 or CSI1 in advance by using the port 3 mode control register (PMC3) and port 3 function control register (PFC3) (refer to **12.3.4 Port 3**).

Caution UART1 or CSI1 transmission/reception operations are not guaranteed if the mode is switched between UART1 and CSI1 during transmission or reception.

Figure 10-1. Selecting Mode of UART1 or CSI1

	7	6	5	4	3	2	1	0	Address	After reset
PMC3	0	0	0	PMC34	PMC33	PMC32	PMC31	PMC30	FFFFF446H	00H

	7	6	5	4	3	2	1	0	Address	After reset
PFC3	0	0	0	PFC34	PFC33	PFC32	0	0	FFFFF466H	00H

PFC3n	PMC3n	Operation mode
0	0	Port I/O mode
0	1	UART1 mode
1	0	Port I/O mode
1	1	CSI1 mode

Remark n = 2 to 4

10.2 Asynchronous Serial Interface 0 (UART0)

10.2.1 Features

- Transfer rate: 300 bps to 1,250 kbps (using a dedicated baud rate generator and an internal system clock of 40 MHz)
- Full-duplex communications
  On-chip receive buffer register 0 (RXB0)
  On-chip transmit buffer register 0 (TXB0)
- Two-pin configuration[Note]
  TXD0: Transmit data output pin
  RXD0: Receive data input pin
- Reception error detection functions
  - Parity error
  - Framing error
  - Overrun error
- Interrupt sources: 3 types
  - Reception error interrupt (INTSER0): Interrupt is generated according to the logical OR of the three types of reception errors
  - Reception completion interrupt (INTSR0): Interrupt is generated when receive data is transferred from the receive shift register to receive buffer register 0 after serial transfer is completed during a reception enabled state
  - Transmission completion interrupt (INTST0): Interrupt is generated when the serial transmission of transmit data (8 or 7 bits) from the transmit shift register is completed
- The character length of transmit/receive data is specified by to the ASIM0 register
- 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

Note The SCK and CTS pins are not available for UART0.

10.2.2 Configuration

UART0 is controlled by asynchronous serial interface mode register 0 (ASIM0), asynchronous serial interface status register 0 (ASIS0), and asynchronous serial interface transmission status register 0 (ASIF0). Receive data is maintained in receive buffer register 0 (RXB0), and transmit data is written to transmit buffer register 0 (TXB0).

Figure 10-2 shows the configuration of asynchronous serial interface 0 (UART0).

(1) Asynchronous serial interface mode register 0 (ASIM0)

The ASIM0 register is an 8-bit register for specifying the operation of the asynchronous serial interface.

(2) Asynchronous serial interface status register 0 (ASIS0)

The ASIS0 register 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 ASIS0 register is read.

(3) Asynchronous serial interface transmission status register 0 (ASIF0)

The ASIF0 register 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 TXB0 data, 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 ASIM0 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 ASIS0 register.

(5) Receive shift register

This is a shift register that converts the serial data that was input to the RXD0 pin to parallel data. One byte of data is received, and if a stop bit is detected, the receive data is transferred to receive buffer register 0 (RXB0).

This register cannot be directly manipulated.

(6) Receive buffer register 0 (RXB0)

This 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 RXB0 register, synchronized with the end of the shift-in processing of one frame.

Also, the reception completion interrupt request (INTSR0) is generated by the transfer of data to the RXB0 register.

(7) Transmit shift register

This is a shift register that converts the parallel data that was transferred from transmit buffer register 0 (TXB0) to serial data.

When one byte of data is transferred from the TXB0 register, the shift register data is output from the TXD0 pin.

The transmission completion interrupt request (INTST0) is generated synchronized with the completion of transmission of one frame.

This register cannot be directly manipulated.

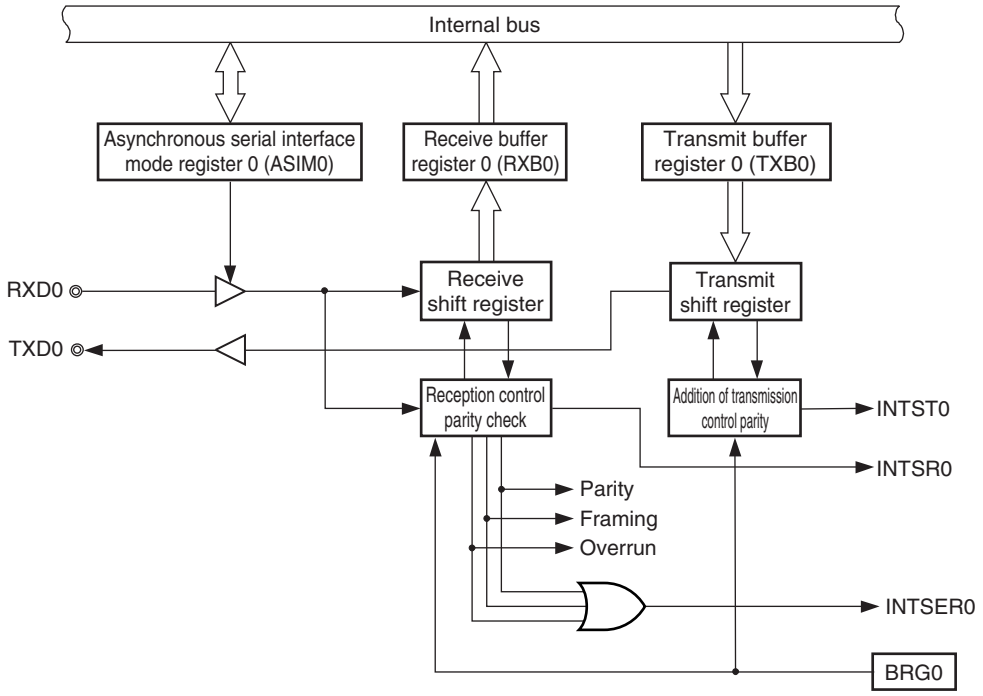
(8) Transmit buffer register 0 (TXB0)

TXB0 is an 8-bit buffer for transmit data. A transmit operation is started by writing transmit data to TXB0.

(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 TXB0 register, according to the contents that were set in the ASIM0 register.

Figure 10-2. Asynchronous Serial Interface 0 Block Diagram



★ **Remark** For the configuration of baud rate generator 0, see **Figure 10-13**.

10.2.3 Control registers

(1) Asynchronous serial interface mode register 0 (ASIM0)

The ASIM0 register is an 8-bit register that controls the UART0 transfer operation.

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

Cautions 1. When using UART0, be sure to set the external pins related to UART0 functions to the control made before setting clock select register 0 (CKSR0) and the baud rate generator control register (BRGC0), and then set the UARTCAE0 bit to 1. Then set the other bits.

2. Set the UARTCAE0 and RXE0 bits to 1 while a high level is input to the RXD0 pin. If these bits are set to 1 while the pin is at low level, reception is started.

(1/3)

	<7>	<6>	<5>	4	3	2	1	0	Address	After reset
ASIM0	UARTCAE0	TXE0	RXE0	PS1	PS0	CL	SL	ISRM	FFFFFA00H	01H

Bit position	Bit name	Function
7	UARTCAE0	Controls the operating clock. 0: Stops clock supply to UART0. 1: Supplies clock to UART0. **Cautions 1. If UARTCAE0 = 0, UART0 is asynchronously reset[Note].** **2. If UARTCAE0 = 0, UART0 is reset. To operate UART0, first set UARTCAE0 to 1.** **3. If the UARTCAE0 bit is cleared from 1 to 0, all the registers of UART0 are initialized. To set UARTCAE0 to 1 again, be sure to re-set the registers of UART0.** The output of the TXD0 pin goes high when transmission is disabled, regardless of the setting of the UARTCAE0 bit.
6	TXE0	Enables/disables transmission. 0: Disables transmission 1: Enables transmission **Cautions 1. Set the TXE0 bit to 1 after setting the UARTCAE0 bit to 1 at startup. Set the UARTCAE0 bit to 0 after setting the TXE0 bit to 0 to stop.** **2. To initialize the transmission unit, clear (0) the TXE0 bit, and after letting 2 Clock cycles (base clock) elapse, set (1) the TXE0 bit again. If the TXE0 bit is not set again, initialization may not be successful. (For details about the base clock, refer to 10.2.6 (1) (a) Base clock (Clock).)**

Note The ASIS0, ASIF0, and RXB0 registers are reset.

(2/3)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>5</td><td>RXE0</td><td>Enables/disables reception.
0: Disables reception[Note]
1: Enables reception

Cautions 1. Set the RXE0 bit to 1 after setting the UARTCAE0 bit to 1 at startup. Set the UARTCAE0 bit to 0 after setting the RXE0 bit to 0 to stop.
2. To initialize the reception unit status, clear (0) the RXE0 bit, and after letting 2 Clock cycles (base clock) elapse, set (1) the RXE0 bit again. If the RXE0 bit is not set again, initialization may not be successful. (For details about the base clock, refer to 10.2.6 (1) (a) Base clock (Clock).)</td></tr>
<tr><td>4, 3</td><td>PS1, PS0</td><td>Controls parity bit.

<table>
<tr><th>PS1</th><th>PS0</th><th>Transmit operation</th><th>Receive operation</th></tr>
<tr><td>0</td><td>0</td><td>Don’t output parity bit</td><td>Receive with no parity</td></tr>
<tr><td>0</td><td>1</td><td>Output 0 parity</td><td>Receive as 0 parity</td></tr>
<tr><td>1</td><td>0</td><td>Output odd parity</td><td>Judge as odd parity</td></tr>
<tr><td>1</td><td>1</td><td>Output even parity</td><td>Judge as even parity</td></tr>
</table>

Cautions 1. To overwrite the PS1 and PS0 bits, first clear (0) the TXE0 and RXE0 bits.
2. If “0 parity” is selected for reception, no parity judgment is performed. Therefore, no error interrupt is generated because the PE bit of the ASIS0 register is not set.
• Even parity
If the transmit data contains an odd number of bits with the value “1”, the parity bit is set (1). If it contains an even number of bits with the value “1”, the parity bit is cleared (0). This controls the number of bits with the value “1” contained in the transmit data and the parity bit so that it is an even number.
During reception, the number of bits with the value “1” contained in the receive data and the parity bit is counted, and if the number is odd, a parity error is generated.

• Odd parity
In contrast to even parity, odd parity controls the number of bits with the value “1” contained in the transmit data and the parity bit so that it is an odd number.
During reception, the number of bits with the value “1” contained in the receive data and the parity bit is counted, and if the number is even, a parity error is generated.</td></tr>
</table>

Note When reception is disabled, the receive shift register does not detect a start bit. No shift-in processing or transfer processing to receive buffer register 0 (RXB0) is performed, and the contents of the RXB0 register are retained.

When reception is enabled, the reception 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 RXB0 register. A reception completion interrupt (INTSR0) is also generated in synchronization with the transfer to the RXB0 register.

(3/3)

Bit position	Bit name	Function
4, 3	PS1, PS0	• 0 parity During transmission, the parity bit is cleared (0) regardless of the transmit data. During reception, no parity error is generated because no parity bit is checked. • No parity No parity bit is added to transmit data. During reception, the receive data is considered to have no parity bit. No parity error is generated because there is no parity bit.
2	CL	Specifies character length of 1 frame of transmit/receive data. 0: 7 bits 1: 8 bits **Caution To overwrite the CL bit, first clear (0) the TXE0 and RXE0 bits.**
1	SL	Specifies stop bit length of transmit data. 0: 1 bit 1: 2 bits **Cautions 1. To overwrite the SL bit, first clear (0) the TXE0 bit.** **2. Since reception is always done with a stop bit length of 1, the SL bit setting does not affect receive operations.**
0	ISRM	Enables/disables generation of reception completion interrupt requests when an error occurs. 0: Generate a reception error interrupt request (INTSER0) as an interrupt when an error occurs. In this case, no reception completion interrupt request (INTSR0) is generated. 1: Generate a reception completion interrupt request (INTSR0) as an interrupt when an error occurs. In this case, no reception error interrupt request (INTSER0) is generated. **Caution To overwrite the ISRM bit, first clear (0) the RXE0 bit.**

★ **(2) Asynchronous serial interface status register 0 (ASIS0)**

The ASIS0 register, which consists of 3-bit error flags (PE, FE and OVE), indicates the error status when UART0 reception is complete.

The ASIS0 register is cleared to 00H by a read operation. When a reception error occurs, receive buffer register 0 (RXB0) should be read and the error flag should be cleared after the ASIS0 register is read.

This register is read-only, in 8-bit units.

Cautions 1. When the UARTCAE0 bit or RXE0 bit of the ASIM0 register is set to 0, or when the ASIS0 register is read, the PE, FE, and OVE bits of the ASIS0 register are cleared (0).

★ **2. Manipulation using a bit manipulation instruction is prohibited.**

	7	6	5	4	3	2	1	0	Address	After reset
ASIS0	0	0	0	0	0	PE	FE	OVE	FFFFFA03H	00H

Bit position	Bit name	Function
2	PE	This is a status flag that indicates a parity error. 0: When the ASIM0 register's UARTCAE0 and RXE0 bits are both set to 0, or after the ASIS0 register is read 1: When the receive data parity does not match the parity bit after receive completion **Caution The operation of the PE bit differs according to the settings of the PS1 and PS0 bits of the ASIM0 register.**
1	FE	This is a status flag that indicates a framing error. 0: When the ASIM0 register's UARTCAE0 and RXE0 bits are both set to 0, or after the ASIS0 register is read 1: When no stop bit was detected after receive completion **Caution For receive data stop bits, only the first bit is checked regardless of the stop bit length.**
0	OVE	This is a status flag that indicates an overrun error. 0: When the ASIM0 register's UARTCAE0 and RXE0 bits are both 0, or after the ASIS0 register is read. 1: When UART0 completed the next receive operation before reading the receive data in the RXB0 register. **Caution When an overrun error occurs, the next receive data value is not written to the RXB0 register and the data is discarded.**

(3) Asynchronous serial interface transmission status register 0 (ASIF0)

The ASIF0 register, which consists of 2-bit status flags, indicates the status during transmission.

By writing the next data to the TXB0 register after data is transferred from the TXB0 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 TXBF0 bit of the ASIF0 register to prevent writing to the TXB0 register by mistake.

This register is read-only, in 8-bit or 1-bit units.

	7	6	5	4	3	2	<1>	<0>	Address	After reset
ASIF0	0	0	0	0	0	0	TXBF0	TXSF0	FFFFFA05H	00H

Bit position	Bit name	Function
1	TXBF0	This is a transmit buffer data flag. 0: Data to be transferred next to TXB0 register does not exist (When the ASIM0 register's UARTCAE0 or TXE0 bits is 0, or when data has been transferred to the transmit shift register) 1: Data to be transferred next exists in TXB0 register (Data exists in TXB0 register when the TXB0 register has been written to) **Caution When transmission is performed continuously, data should be written to the TXB0 register after confirming that this flag is 0. If writing to TXB0 register is performed when this flag is 1, transmit data cannot be guaranteed**.
0	TXSF0	This is a transmit shift register data flag. It indicates the transmission status of UART0. 0: Initial status or a waiting transmission (When the ASIM0 register's UARTCAE0 or TXE0 bits is set to 0, or when following transmission completion, the next data transfer from the TXB0 register is not performed) 1: Transmission in progress (When data has been transferred from the TXB0 register) **Caution 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 (INTST0). If initialization is performed when this flag is 1, transmit data cannot be guaranteed.**

(4) Receive buffer register (RXB0)

The RXB0 register is an 8-bit buffer register for storing parallel data that had been converted by the receive shift register.

When reception is enabled (RXE0 bit = 1 in the ASIM0 register), receive data is transferred from the receive shift register to the RXB0 register, synchronized with the completion of the shift-in processing of one frame. Also, a reception completion interrupt request (INTSR0) is generated by the transfer to the RXB0 register. For information about the timing for generating this interrupt request, refer to **10.2.5 (4) Receive operation**.

If reception is disabled (RXE0 bit = 0 in the ASIM0 register), the contents of the RXB0 register are retained, and no processing is performed for transferring data to the RXB0 register even when the shift-in processing of one frame is completed. Also, no INTSR0 signal is generated.

When 7 bits is specified for the data length, bits 6 to 0 of the RXB0 register are transferred for the receive data and the MSB (bit 7) is always 0. However, if an overrun error (OVE bit of ASIS0 register = 1) occurs, the receive data at that time is not transferred to the RXB0 register.

Except when a reset is input, the RXB0 register becomes FFH even when UARTCAE0 bit = 0 in the ASIM0 register.

This register is read-only, in 8-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
RXB0	RXB7	RXB6	RXB5	RXB4	RXB3	RXB2	RXB1	RXB0	FFFFFA02H	FFH

Bit position	Bit name	Function
7 to 0	RXB7 to RXB0	Stores receive data. 0 can be read for RXB7 when 7-bit or character data is received.

(5) Transmit buffer register 0 (TXB0)

The TXB0 register is an 8-bit buffer register for setting transmit data.

When transmission is enabled (TXE0 bit = 1 in the ASIM0 register), the transmit operation is started by writing data to TXB0 register.

When transmission is disabled (TXE0 bit = 0 in the ASIM0 register), even if data is written to TXB0 register, the value is ignored.

The TXB0 register data is transferred to the transmit shift register, and a transmission completion interrupt request (INTST0) 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 **10.2.5 (2) Transmit operation**.

When TXBF0 bit = 1 in the ASIF0 register, writing must not be performed to TXB0 register.

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

	7	6	5	4	3	2	1	0	Address	After reset
TXB0	TXB7	TXB6	TXB5	TXB4	TXB3	TXB2	TXB1	TXB0	FFFFFA04H	FFH

Bit position	Bit name	Function
7 to 0	TXB7 to TXB0	Writes transmit data.

10.2.4 Interrupt requests

The following three types of interrupt requests are generated from UART0.

- Reception completion interrupt (INTSR0)
- Transmission completion interrupt (INTST0)
- Reception error interrupt (INTSER0)

The default priorities among these three types of interrupt requests is, from high to low, reception completion interrupt, transmission completion interrupt, and reception error interrupt.

Table 10-1. Generated Interrupts and Default Priorities

Interrupt	Priority
Reception completion	1
Transmission completion	2
Reception error	3

(1) Reception completion interrupt (INTSR0)

When reception is enabled, an INTSR0 signal is generated when data is shifted in to the receive shift register and transferred to receive buffer register 0 (RXB0).

An INTSR0 signal can be generated in place of a reception error interrupt (INTSER0) according to the ISRM bit of the ASIM0 register even when a reception error has occurred.

When reception is disabled, no INTSR0 signal is generated.

(2) Transmission completion interrupt (INTST0)

An INTST0 signal is generated when one frame of transmit data containing 7-bit or 8-bit characters is shifted out from the transmit shift register.

(3) Reception error interrupt (INTSER0)

When reception is enabled, an INTSER0 signal is generated according to the logical OR of the three types of reception errors explained for the ASIS0 register. Whether an INTSER0 signal or an INTSR0 signal is generated when an error occurs can be specified using the ISRM bit of the ASIM0 register.

When reception is disabled, no INTSER0 signal is generated.

10.2.5 Operation

(1) Data format

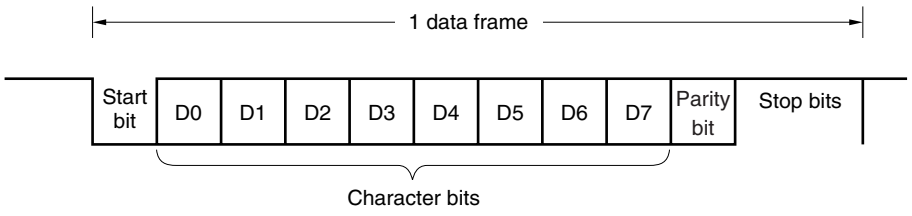
Full-duplex serial data transmission and reception can be performed.

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 10-3.

The character bit length within one data frame, the type of parity, and the stop bit length are specified according to asynchronous serial interface mode register 0 (ASIM0).

Also, data is transferred with LSB first.

Figure 10-3. 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 UARTCAE0 bit is set to 1 in the ASIM0 register, a high level is output from the TXD0 pin.

Then, when the TXE0 bit is set to 1 in the ASIM0 register, transmission is enabled, and the transmit operation is started by writing transmit data to transmit buffer register 0 (TXB0).

(a) Transmission enabled state

This state is set by the TXE0 bit in the ASIM0 register.

- TXE0 = 1: Transmission enabled state
- TXE0 = 0: Transmission disabled state

Since UART0 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 transmit buffer register 0 (TXB0). When a transmit operation is started, the data in the TXB0 register is transferred to transmit shift register. Then, the transmit shift register outputs data to the TXD0 pin (the transmit data is transferred sequentially starting with the start bit). The start bit, parity bit, and stop bits are added automatically.

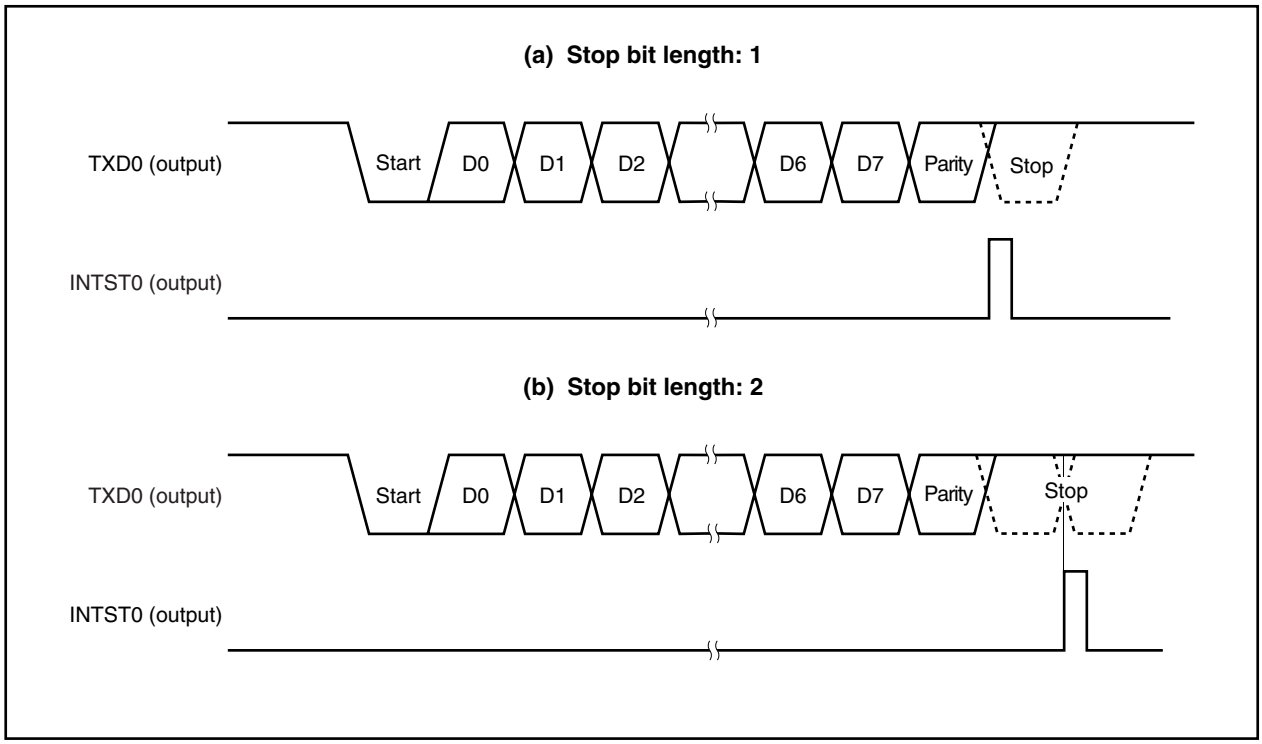
(c) Transmission interrupt request

When the transmit shift register becomes empty, a transmission completion interrupt request (INTST0) is generated. The timing for generating the INTST0 signal differs according to the specification of the stop bit length. The INTST0 signal 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 TXB0 register, the transmit operation is suspended.

Caution Normally, when the transmit shift register becomes empty, a transmission completion interrupt (INTST0) is generated. However, no INTST0 signal is generated if the transmit shift register becomes empty due to the input of $\overline{\text{RESET}}$.

Figure 10-4. Asynchronous Serial Interface Transmission Completion Interrupt Timing



(3) Continuous transmission operation

UART0 can write the next transmit data to the TXB0 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 servicing of the transmission completion interrupt (INTST0) after the transmission of one data frame. In addition, reading the TXSF0 bit of the ASIF0 register after the generation of an INTST0 signal enables the TXB0 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 ASIF0 register to confirm the transmission status and whether or not data can be written to the TXB0 register.

Caution The TXBF0 and TXSF0 bits of the ASIF0 register change "10" → "11" → "01" during continuous transmission. Therefore, do not confirm the status based on the combination of the TXBF0 and TXSF0 bits.
Judge the status based only on the TXBF0 bit when performing continuous transmission.

TXBF0	Whether or Not Writing to TXB0 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 TXB0 register and confirm that the TXBF0 bit is 0, and then write the next transmit data (second byte) to TXB0 register. If writing to the TXB0 register is performed when the TXBF0 bit is 1, transmit data cannot be guaranteed.

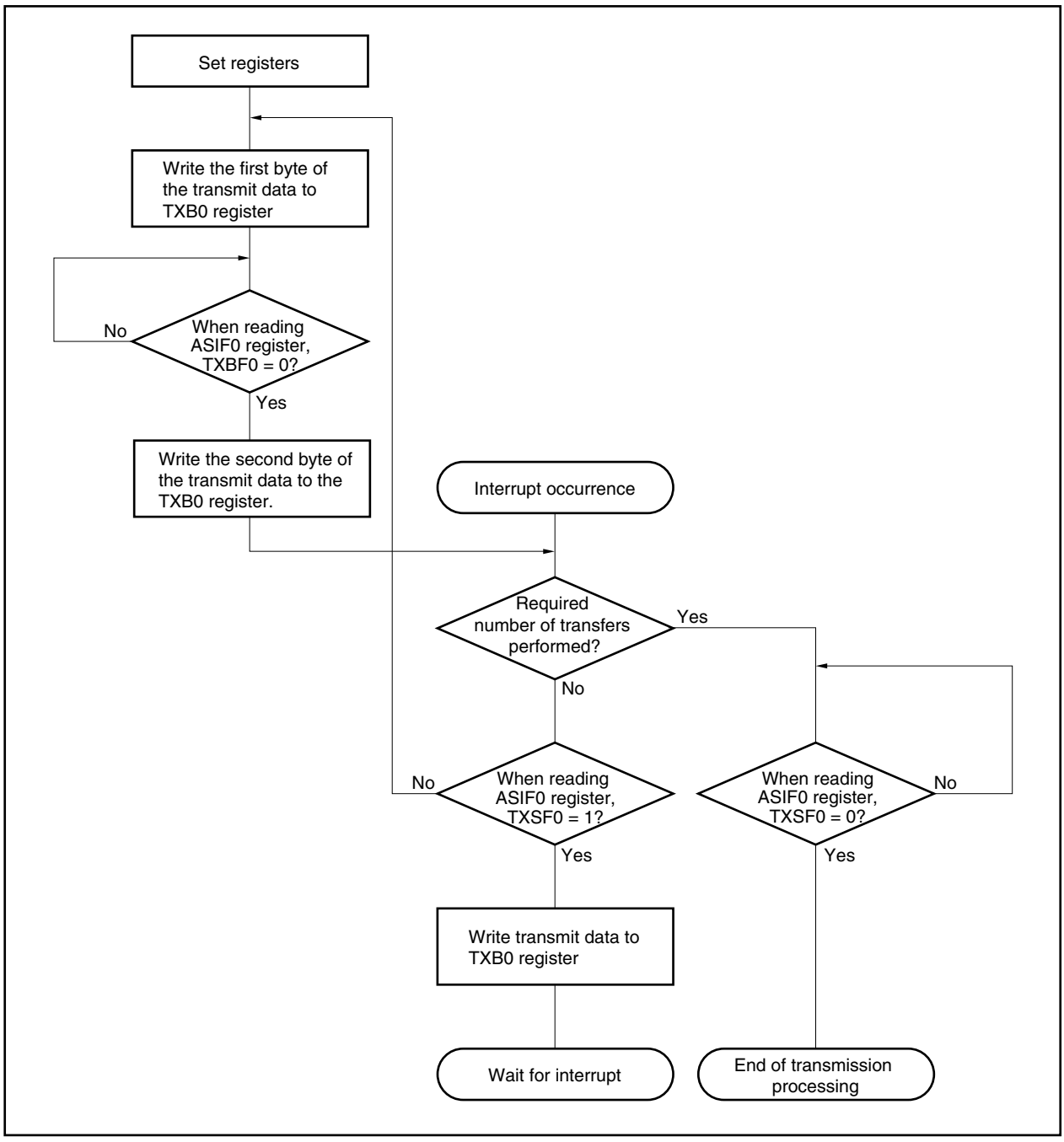
The communication status can be confirmed with the TXSF0 bit.

TXSF0	Transmission Status
0	Transmission is completed.
1	Under transmission.

Cautions 1. When initializing the transmission unit when continuous transmission is completed, confirm that the TXBF0 bit is 0 after the occurrence of the transmission completion interrupt, and then execute initialization. If initialization is performed when the TXBF0 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 INTST0 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 TXSF0 bit.

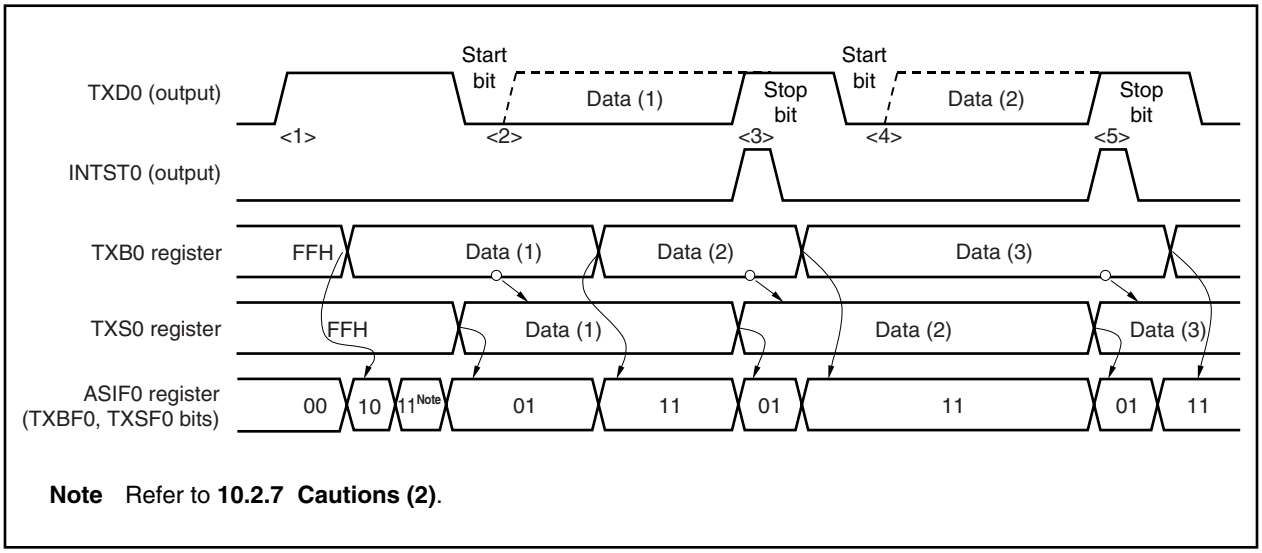
Figure 10-5. Continuous Transmission Processing Flow



(a) Starting procedure

The procedure to start continuous transmission is shown below.

Figure 10-6. Continuous Transmission Starting Procedure



Note Refer to **10.2.7 Cautions (2)**.

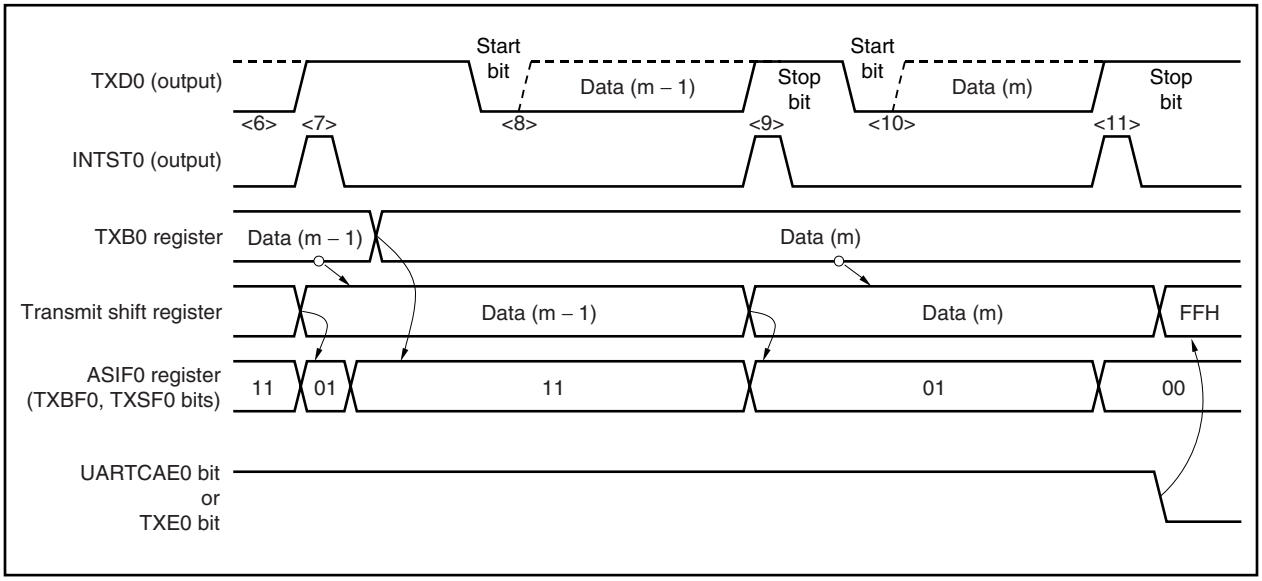
Transmission Starting Procedure	Internal Operation	ASIF0 Register	
		TXBF0	TXSF0
• Set transmission mode	<1> Start transmission unit	0	0
• Write data (1) →		1	0
	<2> Generate start bit	1	1 Note
		0	1
	Start data (1) transmission →	0	1
• Read ASIF0 register (confirm that TXBF0 bit = 0) ←		0	1
• Write data (2) →		1	1
	<<Transmission in progress>>		
	<3> INTST0 interrupt occurs →	0	1
• Read ASIF0 register (confirm that TXBF0 bit = 0) ←		0	1
• Write data (3) →		1	1
	<4> Generate start bit		
	Start data (2) transmission		
	<<Transmission in progress>>		
	<5> INTST0 interrupt occurs →	0	1
• Read ASIF0 register (confirm that TXBF0 bit = 0) ←		0	1
• Write data (4) →		1	1

Note Refer to **10.2.7 Cautions (2)**.

(b) Ending procedure

The procedure for ending continuous transmission is shown below.

Figure 10-7. Continuous Transmission End Procedure



Transmission End Procedure	Internal Operation	ASIF0 Register	
		TXBF0	TXSF0
	<6> Transmission of data (m – 2) is in progress	1	1
	<7> INTST0 interrupt occurs →	0	1
• Read ASIF0 register (confirm that TXBF0 bit = 0) ←		0	1
• Write data (m) →		1	1
	<8> Generate start bit Start data (m – 1) transmission <<Transmission in progress>>		
	<9> INTST0 interrupt occurs →	0	1
• Read ASIF0 register (confirm that TXSF0 bit = 1) ←		0	1
There is no write data			
	<10> Generate start bit Start data (m) transmission <<Transmission in progress>>		
	<11> Generate INTST0 interrupt →	0	0
• Read ASIF0 register (confirm that TXSF0 bit = 0) ←		0	0
• Clear (0) the UARTCAE0 bit or TXE0 bit	Initialize internal circuits		

(4) Receive operation

The awaiting reception state is set by setting the UARTCAE0 bit to 1 in the ASIM0 register and then setting the RXE0 bit to 1 in the ASIM0 register. To start reception, start sampling at the falling edge of the RXD0 pin upon detection of the falling edge. If the RXD0 pin is at low level at the sampling point of a start bit, 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 (INTSR0) is generated each time the reception of one frame of data is completed. Normally, the receive data is transferred from receive buffer register 0 (RXB0) to memory by this interrupt servicing.

(a) Reception enabled state

The receive operation is set to the reception enabled state by setting the RXE0 bit in the ASIM0 register to 1.

- RXE0 bit = 1: Reception enabled state
- RXE0 bit = 0: Reception disabled state

In reception disabled state, the reception hardware stands by in the initial state. At this time, the contents of receive buffer register 0 (RXB0) 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 RXD0 pin is sampled using the serial clock from baud rate generator 0 (BRG0).

(c) Reception completion interrupt

When RXE0 = 1 in the ASIM0 register and the reception of one frame of data is completed (the stop bit is detected), a reception completion interrupt (INTSR0) is generated and the receive data in the receive shift register is transferred to the RXB0 register at the same time.

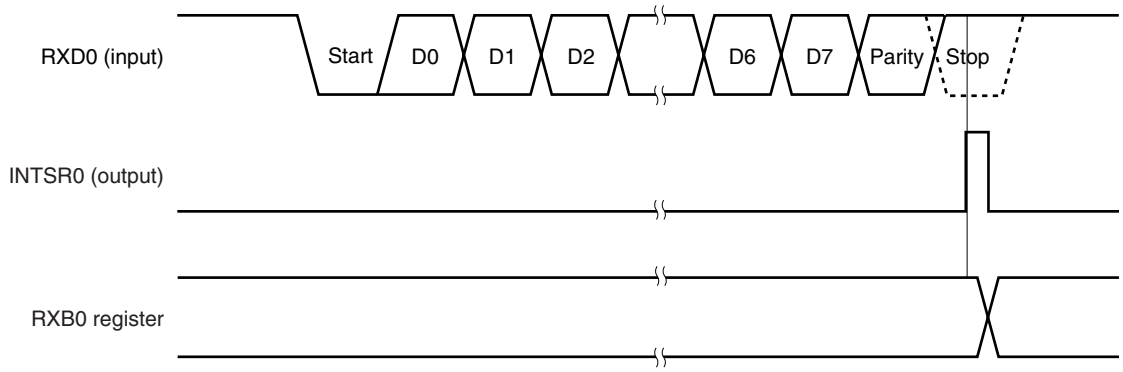
Also, if an overrun error (OVE bit of ASIS0 register = 1) occurs, the receive data at that time is not transferred to receive buffer register 0 (RXB0), and either an INTSR0 signal or a reception error interrupt (INTSER0) is generated according to the ISRM bit setting in the ASIM0 register.

Even if a parity error (PE bit of ASIS0 register = 1) or framing error (FE bit of ASIS0 register = 1) occurs during a reception operation, the receive operation continues until stop bit is received, and after reception is completed, either an INTSR0 signal or an INTSER0 signal is generated according to the ISRM bit setting in the ASIM0 register (the receive data in the receive shift register is transferred to the RXB0 register).

If the RXE0 bit is cleared (0) during a receive operation, the receive operation is immediately stopped. The contents of receive buffer register 0 (RXB0) and of the asynchronous serial interface status register (ASIS0) at this time do not change, and no INTSR0 or INTSER0 signal is generated.

No INTSR0 or INTSER0 signal is generated when RXE0 = 0 (reception is disabled).

Figure 10-8. Asynchronous Serial Interface Reception Completion Interrupt Timing



Cautions 1. Be sure to read receive buffer register 0 (RXB0) even when a reception error occurs. If the RXB0 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 ASIS0 register are set (1), and a reception error interrupt (INTSER0) or a reception completion interrupt (INTSR0) is generated at the same time. The ISRM bit of the ASIM0 register specifies whether an INTSER0 or INTSR0 signal is generated.

The type of error that occurred during reception can be detected by reading the contents of the ASIS0 register during the INTSER0 or INTSR0 interrupt servicing.

The contents of the ASIS0 register are cleared (0) by reading the ASIS0 register.

Table 10-2. Reception Error Causes

Error Flag	Reception Error	Cause
PE	Parity error	The parity specification during transmission did not match the parity of the reception data
FE	Framing error	No stop bit was detected
OVE	Overrun error	The reception of the next data was completed before data was read from receive buffer register 0 (RXB0)

(a) Separation of reception error interrupt

A reception error interrupt can be separated from the INTSR0 signal and generated as the INTSER0 signal by clearing the ISRM bit of the ASIM0 register to 0.

Figure 10-9. When Reception Error Interrupt Is Separated from INTSR0 Signal (ISRM Bit = 0)

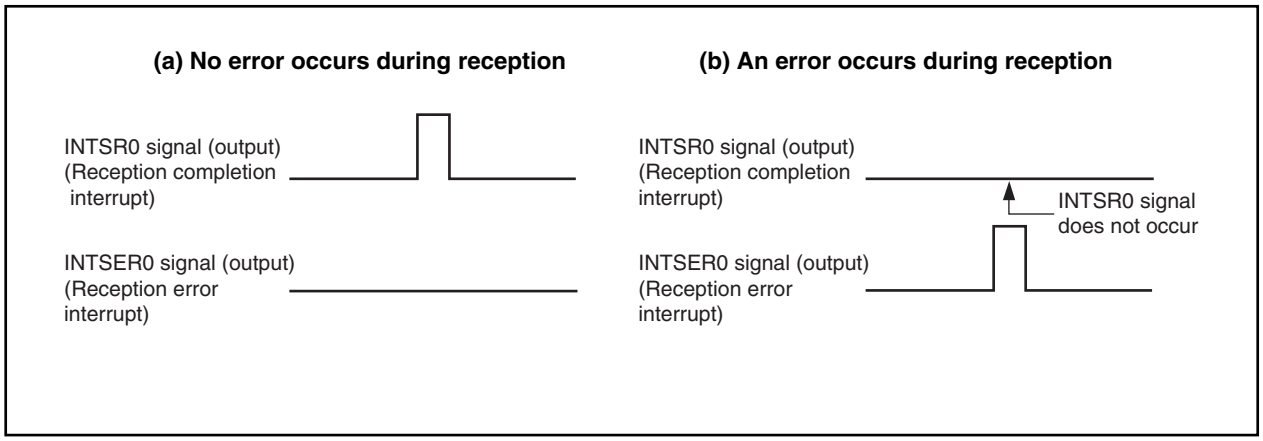
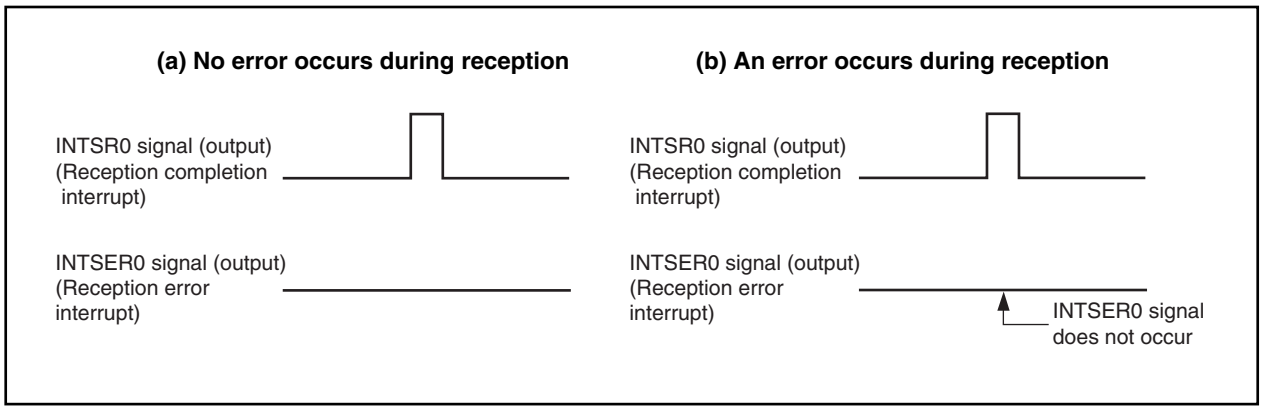


Figure 10-10. When Reception Error Interrupt Is Included in INTSR0 Signal (ISRM Bit = 1)



(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 RXD0 signal is sampled at the rising edge of the prescaler output base clock (f_{CLK}). 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 10-12**). Refer to **10.2.6 (1) (a) Base clock (Clock)** regarding the base clock.

Also, since the circuit is configured as shown in Figure 10-11, internal processing during a receive operation is delayed by up to 2 clocks according to the external signal status.

Figure 10-11. Noise Filter Circuit

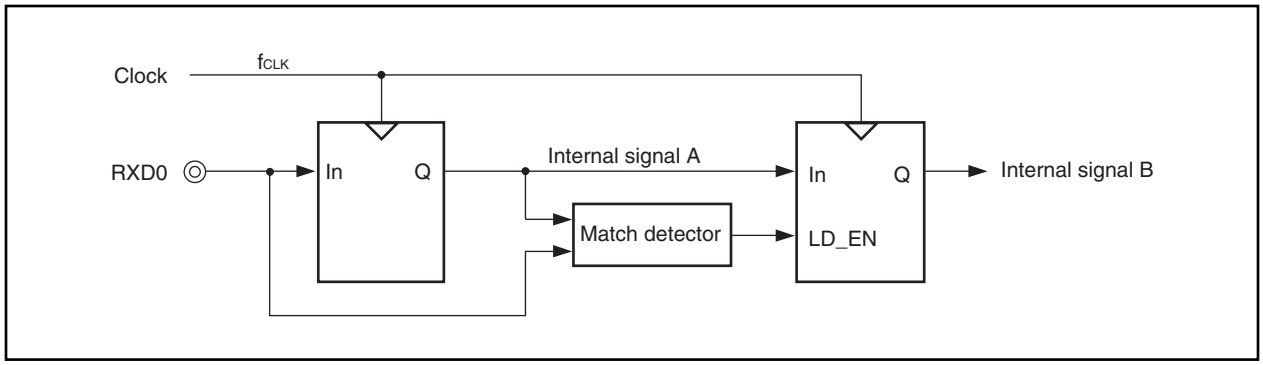
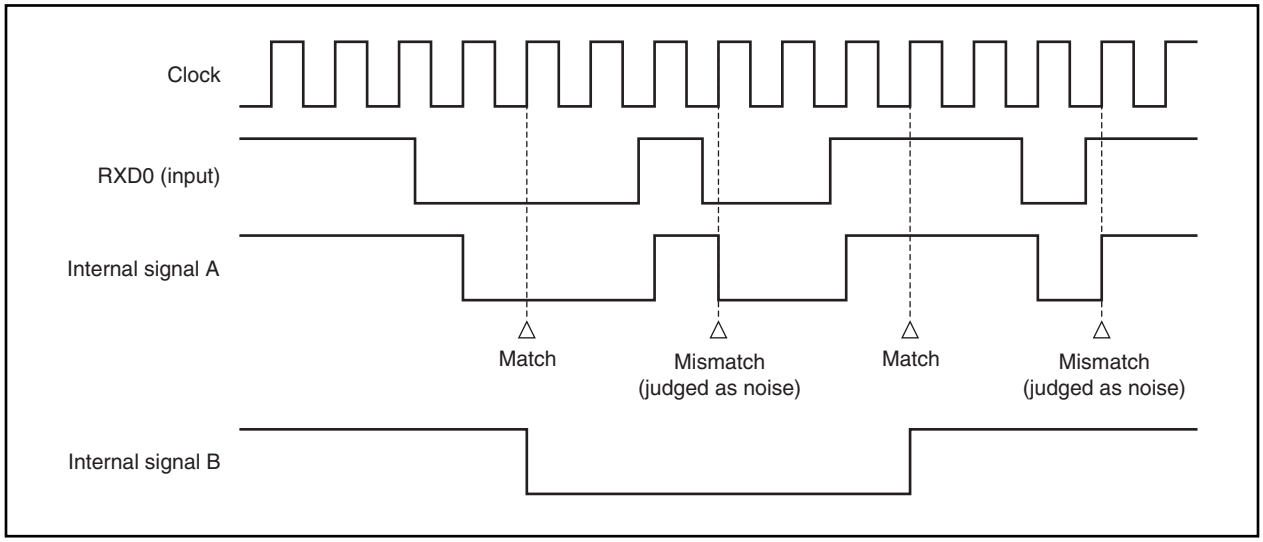


Figure 10-12. Timing of RXD0 Signal Judged as Noise



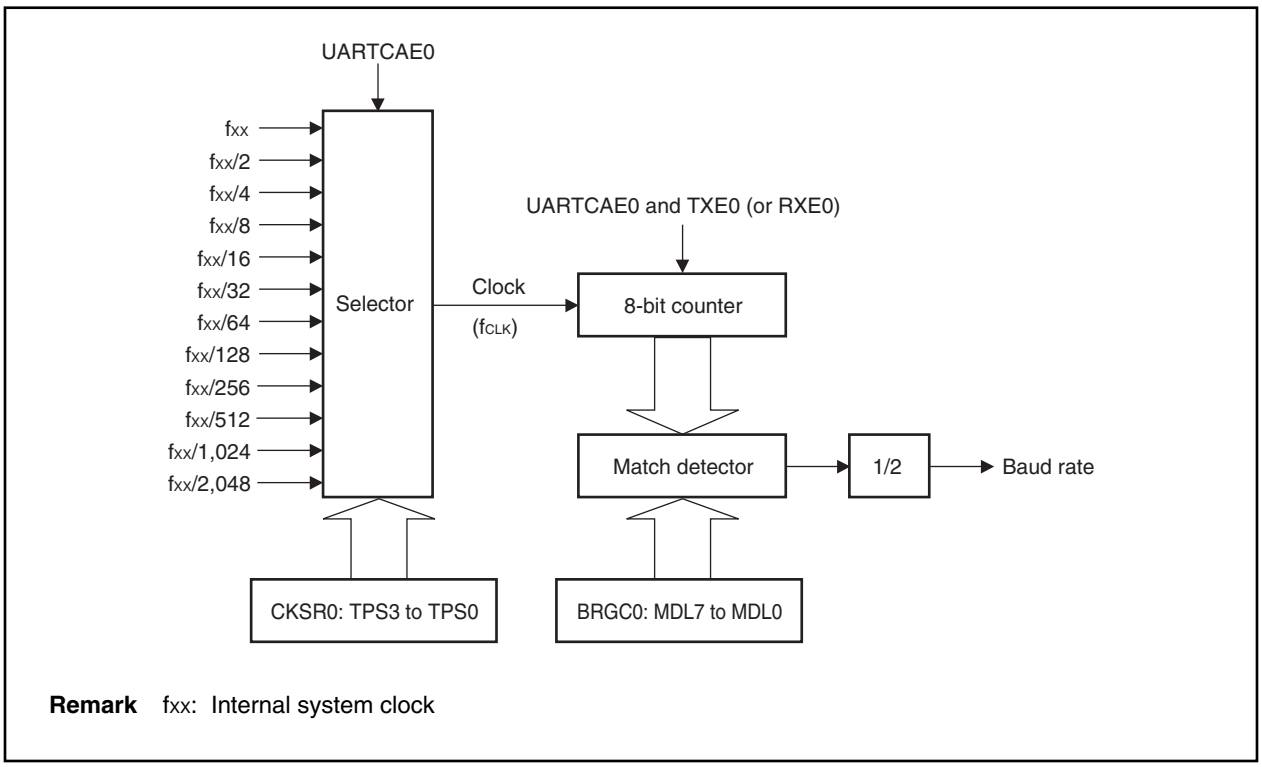
10.2.6 Dedicated baud rate generator 0 (BRG0)

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 UART0. 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 0 (BRG0) configuration

Figure 10-13. Configuration of Baud Rate Generator 0 (BRG0)



Remark fxx: Internal system clock

(a) Base clock (Clock)

When the UARTCAE0 bit = 1 in the ASIM0 register, the clock selected according to the TPS3 to TPS0 bits of the CKSR0 register is supplied to the transmission/reception unit. This clock is called the base clock (f_{CLK}). When UARTCAE0 = 0, f_{CLK} is fixed to low level.

(2) Serial clock generation

A serial clock can be generated according to the settings of the CKSR0 and BRGC0 registers.

The base clock to the 8-bit counter is selected by the TPS3 to TPS0 bits of the CKSR0 register.

The 8-bit counter divisor value can be set by the MDL7 to MDL0 bits of the BRGC0 register.

(a) Clock select register 0 (CKSR0)

The CKSR0 register is an 8-bit register for selecting the base clock (f_{CLK}) using the TPS3 to TPS0 bits. The clock selected by the TPS3 to TPS0 bits becomes the base clock (f_{CLK}) of the transmission/reception module.

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

Cautions 1. The maximum allowable frequency of the base clock (f_{CLK}) is 20 MHz. Therefore, when the system clock's frequency is 40 MHz, TPS3 to TPS0 bits cannot be set to 0000B.

At 40 MHz, set the TPS3 to TPS0 bits to a value other than 0000B, and set the UARTCAE0 bit of the ASIM0 register to 1.

2. Set the UARTCAE0 bit of the ASIM0 register to 0 before rewriting the TPS3 to TPS0 bits.

	7	6	5	4	3	2	1	0	Address	After reset
CKSR0	0	0	0	0	TPS3	TPS2	TPS1	TPS0	FFFFFA06H	00H

Bit position	Bit name	Function
3 to 0	TPS3 to TPS0	Specifies the base clock (f_{CLK})

TPS3	TPS2	TPS1	TPS0	Base clock (f_{CLK})
0	0	0	0	f_{XX}
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	$f_{XX}/2,048$
1	1	Arbitrary	Arbitrary	Setting prohibited

Remark f_{XX}: Internal system clock

(b) Baud rate generator control register 0 (BRGC0)

The BRGC0 register is an 8-bit register that controls the baud rate (serial transfer speed) of UART0.

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

Caution If the MDL7 to MDL0 bits are to be overwritten, the TXE0 and RXE0 bits should be set to 0 in the ASIM0 register first.

	7	6	5	4	3	2	1	0	Address	After reset
BRGC0	MDL7	MDL6	MDL5	MDL4	MDL3	MDL2	MDL1	MDL0	FFFFFA07H	FFH

Bit position	Bit name	Function
7 to 0	MDL7 to MDL0	Specifies the 8-bit counter's division value.

MDL7	MDL6	MDL5	MDL4	MDL3	MDL2	MDL1	MDL0	Division value (k)	Serial clock
0	0	0	0	0	×	×	×	–	Setting prohibited
0	0	0	0	1	0	0	0	8	$f_{CLK}/8$
0	0	0	0	1	0	0	1	9	$f_{CLK}/9$
0	0	0	0	1	0	1	0	10	$f_{CLK}/10$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	0	1	0	250	$f_{CLK}/250$
1	1	1	1	1	0	1	1	251	$f_{CLK}/251$
1	1	1	1	1	1	0	0	252	$f_{CLK}/252$
1	1	1	1	1	1	0	1	253	$f_{CLK}/253$
1	1	1	1	1	1	1	0	254	$f_{CLK}/254$
1	1	1	1	1	1	1	1	255	$f_{CLK}/255$

Remarks 1. f_{CLK}: Frequency [Hz] of base clock selected by TPS3 to TPS0 bits of CKSR0 register

2. k: Value set by MDL7 to MDL0 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.

$$\text{Baud rate} = \frac{f_{CLK}}{2 \times k} \text{ [bps]}$$

f_{CLK} = Frequency [Hz] of base clock selected by TPS3 to TPS0 bits of CKSR0 register.
k = Value set by MDL7 to MDL0 bits of BRGC0 register (k = 8, 9, 10, ..., 255)

(d) Baud rate error

The baud rate error is obtained by the following formula.

$$\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Target baud rate (normal baud rate)}} - 1 \right) \times 100 \text{ [\%]}$$

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 during reception, which is described in (4) Allowable baud rate during reception.

Example: Base clock frequency = 20 MHz = 20,000,000 Hz
Setting of MDL7 to MDL0 bits in BRGC0 register = 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 10-3. Baud Rate Generator Setting Data

Baud Rate (bps)	f_{XX} = 40 MHz			f_{XX} = 33 MHz			f_{XX} = 10 MHz		
	f_{CLK}	k	ERR	f_{CLK}	k	ERR	f_{CLK}	k	ERR
300	$f_{XX}/2^{10}$	65	0.16	$f_{XX}/2^{8}$	215	–0.07	$f_{XX}/2^{7}$	130	0.16
600	$f_{XX}/2^{9}$	65	0.16	$f_{XX}/2^{7}$	215	–0.07	$f_{XX}/2^{6}$	130	0.16
1200	$f_{XX}/2^{8}$	65	0.16	$f_{XX}/2^{6}$	215	–0.07	$f_{XX}/2^{5}$	130	0.16
2400	$f_{XX}/2^{7}$	65	0.16	$f_{XX}/2^{5}$	215	–0.07	$f_{XX}/2^{4}$	130	0.16
4800	$f_{XX}/2^{6}$	65	0.16	$f_{XX}/2^{4}$	215	–0.07	$f_{XX}/2^{3}$	130	0.16
9600	$f_{XX}/2^{5}$	65	0.16	$f_{XX}/2^{3}$	215	–0.07	$f_{XX}/2^{2}$	130	0.16
19200	$f_{XX}/2^{4}$	65	0.16	$f_{XX}/2^{2}$	215	–0.07	$f_{XX}/2^{1}$	130	0.16
31250	$f_{XX}/2^{3}$	80	0	$f_{XX}/2^{2}$	132	0	$f_{XX}/2^{1}$	80	0
38400	$f_{XX}/2^{3}$	65	0.16	$f_{XX}/2^{1}$	215	–0.07	$f_{XX}/2^{0}$	130	0.16
76800	$f_{XX}/2^{2}$	65	0.16	$f_{XX}/2^{1}$	107	0.39	$f_{XX}/2^{0}$	65	0.16
153600	$f_{XX}/2^{1}$	65	0.16	$f_{XX}/2^{1}$	54	–0.54	$f_{XX}/2^{0}$	33	–1.36
312500	$f_{XX}/2^{1}$	32	0	$f_{XX}/2^{1}$	26	1.54	$f_{XX}/2^{0}$	16	0
625000	$f_{XX}/2^{1}$	16	0	$f_{XX}/2^{1}$	13	1.54	$f_{XX}/2^{0}$	8	0
1250000	$f_{XX}/2^{1}$	8	0	$f_{XX}/2^{1}$	8	–17.5	–	–	–

Caution The maximum allowable frequency of the base clock (f_{CLK}) is 20 MHz.

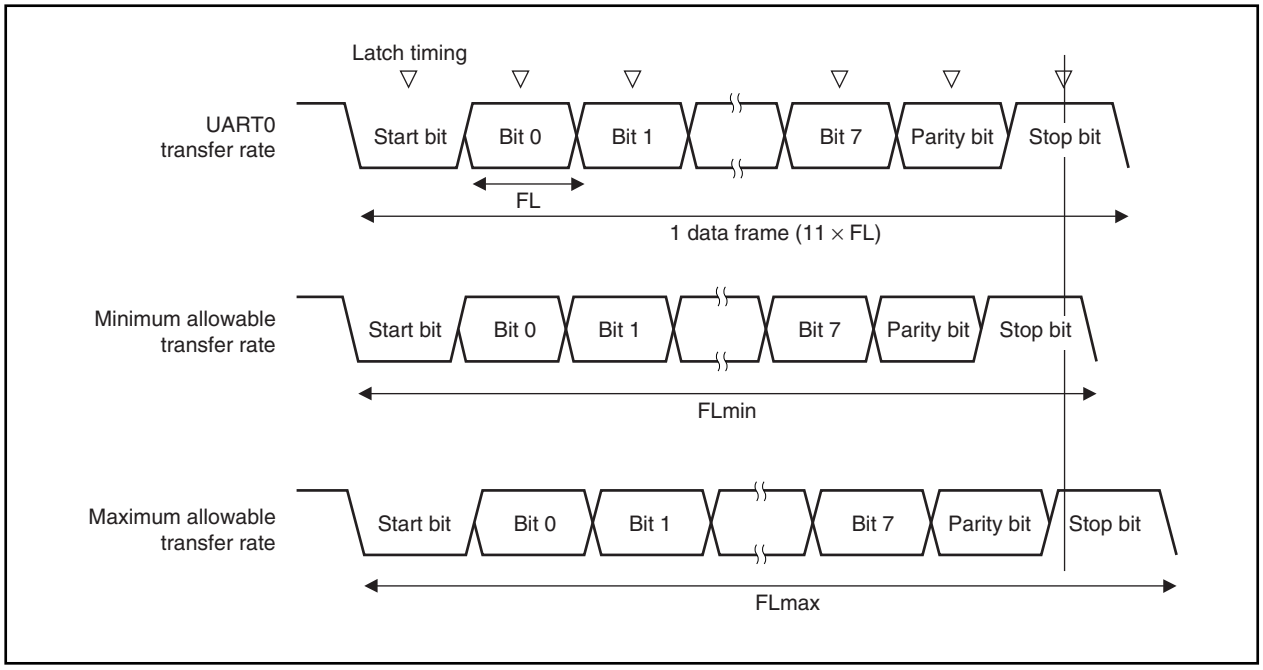
Remarks f_{XX}: Internal system clock frequency
f_{CLK}: Base clock frequency
k: Setting values of MDL7 to MDL0 bits in BRGC0 register
ERR: Baud rate error [%]

(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 10-14. Allowable Baud Rate Range During Reception



As shown in Figure 10-14, after the start bit is detected, the receive data latch timing is determined according to the counter that was set by the BRGC0 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: UART0 baud rate
k: BRGC0 register setting value
FL: 1-bit data length

When the latch timing margin is 2 base clocks, 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.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k + 2} \text{ Brate}$$

Similarly, the maximum allowable transfer rate (FLmax) can be obtained as follows.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k + 2}{2 \times k} \times FL = \frac{21k - 2}{2 \times k} FL$$

$$FLmax = \frac{21k - 2}{20k} FL \times 11$$

Therefore, the transfer destination's minimum receivable baud rate (BRmin) is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k - 2} \text{ Brate}$$

The allowable baud rate error of UART0 and the transfer destination can be obtained as follows from the expressions described above for computing the minimum and maximum baud rate values.

Table 10-4. 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%

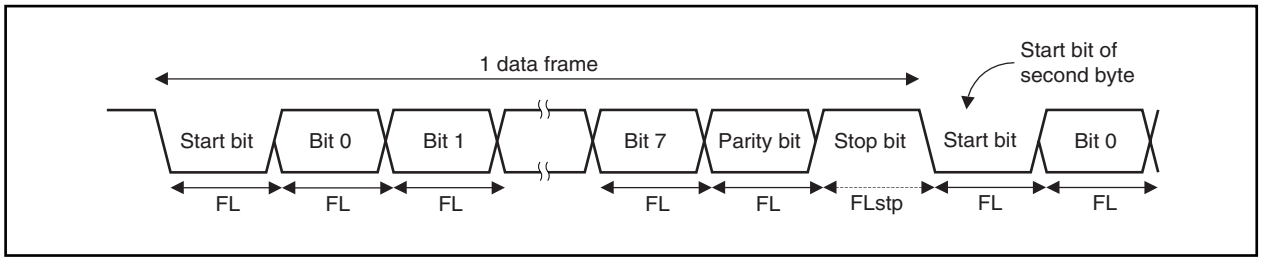
Remarks 1. The reception precision depends on the number of bits in one frame, the base clock frequency, and the division ratio (k). The higher the base clock frequency and the larger the division ratio (k), the higher the precision.

2. k: BRGC0 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 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 10-15. 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_{CLK} yields the following equation.

$$FLstp = FL + 2/f_{CLK}$$

★ Therefore, the transfer rate during continuous transmission is as follows. (when stop bit length = 1)

$$\text{Transfer rate} = 11 \times FL + (2/f_{CLK})$$

10.2.7 Cautions

Cautions to be observed when using UART0 are shown below.

(1) When the supply of clocks to UART0 is stopped (for example, in IDLE or software STOP mode), operation stops with each register retaining the value it had immediately before the supply of clocks was stopped. The TXD0 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 UARTCAE0 = 0, RXE0 = 0, and TXE0 = 0 in the ASIM0 register.

★ (2) UART0 has a 2-stage buffer configuration consisting of transmit buffer register 0 (TXB0) and the transmit shift register, and has status flags (the TXBF0 and TXSF0 bits of the ASIF0 register) that indicate the status of each buffer. When the TXBF0 and TXSF0 bits are read at the same time during continuous transmission, the read values change “10” → “11” → “01”. Judge the timing for writing the next data to the TXB0 register by reading only the TXBF0 bit when performing continuous transmission.

10.3 Asynchronous Serial Interface 1 (UART1)

10.3.1 Features

- Clocked (synchronous) mode/asynchronous mode can be selected
- Operation clock
  Synchronous mode: Baud rate generator/external clock selectable
  Asynchronous mode: Baud rate generator
- Transfer rate
  300 bps to 153,600 bps (in asynchronous mode, fxx = 40 MHz)
  4800 bps to 1000000 bps (in synchronous mode)
- Full-duplex communications (LSB first)
  On-chip receive buffer register 1 (RXB1)
- Three-pin configuration
  TXD1: Transmit data output pin
  RXD1: Receive data input pin
  $\overline{\text{ASCK1}}$: Synchronous serial clock I/O
- Reception error detection function
  - Parity error
  - Framing error
  - Overrun error
- Interrupt sources: 2 types
  - Reception completion interrupt (INTSR1): Interrupt is generated when receive data is transferred from the shift register to receive buffer register 1 (RXB1) after serial transfer is completed during a reception enabled state.
  - Transmission completion interrupt (INTST1): Interrupt is generated when the serial transmission of transmit data (8/7 bits) from the shift register is completed.
- The character length of transmit/receive data is specified by the ASIM10 register (extension bits are specified by the ASIM11 register)
- Character length: 7 or 8 bits
  9 bits (when extension bit is added)
- Parity functions: Odd, even, 0, or no parity
- Transmission stop bits: 1 or 2 bits
- Communication mode: 1-frame transfer or 2-frame continuous transfer enabled
- On-chip dedicated baud rate generator

Remark fxx: Internal system clock

10.3.2 Configuration

UART1 is controlled by asynchronous serial interface mode register 10 and 11 (ASIM10 and ASIM11) and asynchronous serial interface status register 1 (ASIS1). Receive data is held in the receive buffer registers (RXB1 and RXBL1), and transmit data is held in the transmit shift registers (TXS1 and TXSL1).

Figure 10-16 shows the configuration of asynchronous serial interface 1 (UART1).

(1) Asynchronous serial interface mode registers 10, 11 (ASIM10, ASIM11)

The ASIM10 and ASIM11 registers are 8-bit registers that specify the operation of the asynchronous serial interface.

(2) Asynchronous serial interface status register 1 (ASIS1)

The ASIS1 register consists of a transmission status flag (SOT1), reception status flag (SIR1), a bit (RB8) that indicates the 9th bit when extension bit addition is enabled, and 3-bit error flags (PE1, FE1, OVE1) that indicate the error status at reception end.

(3) Reception control parity check

The receive operation is controlled according to the contents set in the ASIM10 and ASIM11 registers. A check for parity errors is also performed during receive operation, and if an error is detected, a value corresponding to the error contents is set in the ASIS1 register.

(4) 2-frame continuous reception buffer register (RXB1)/receive buffer register (RXBL1)

RXB1 is a 16-bit (during 2-frame continuous reception, 9-bit extension data reception) buffer register that holds receive data. During 7 or 8 bit character reception, 0 is stored in the MSB.

For 16-bit access to this register, specify RXB1, and for access to the lower 8 bits, specify RXBL1.

In the reception enabled state, receive data is transferred from the receive shift register to the reception buffer in synchronization with the completion of shift-in processing of one frame.

A reception completion interrupt request (INTSR1) is generated upon transfer to the reception buffer (when 2-frame continuous reception is specified, reception buffer transmission of the second frame).

(5) 2-frame continuous transmission shift register (TXS1)/transmit shift registers (TXSL1)

TXS1 is a 9-bit/2-frame continuous transmission processing shift register. Transmission is started by writing data to this register.

A transmission completion interrupt request (INTST1) is generated in synchronization with the end of transmission of 1 frame or 2 frames including the TXS1 data.

For 16-bit access to this register, specify TXS1, and for access to the lower 8 bits, specify TXSL1.

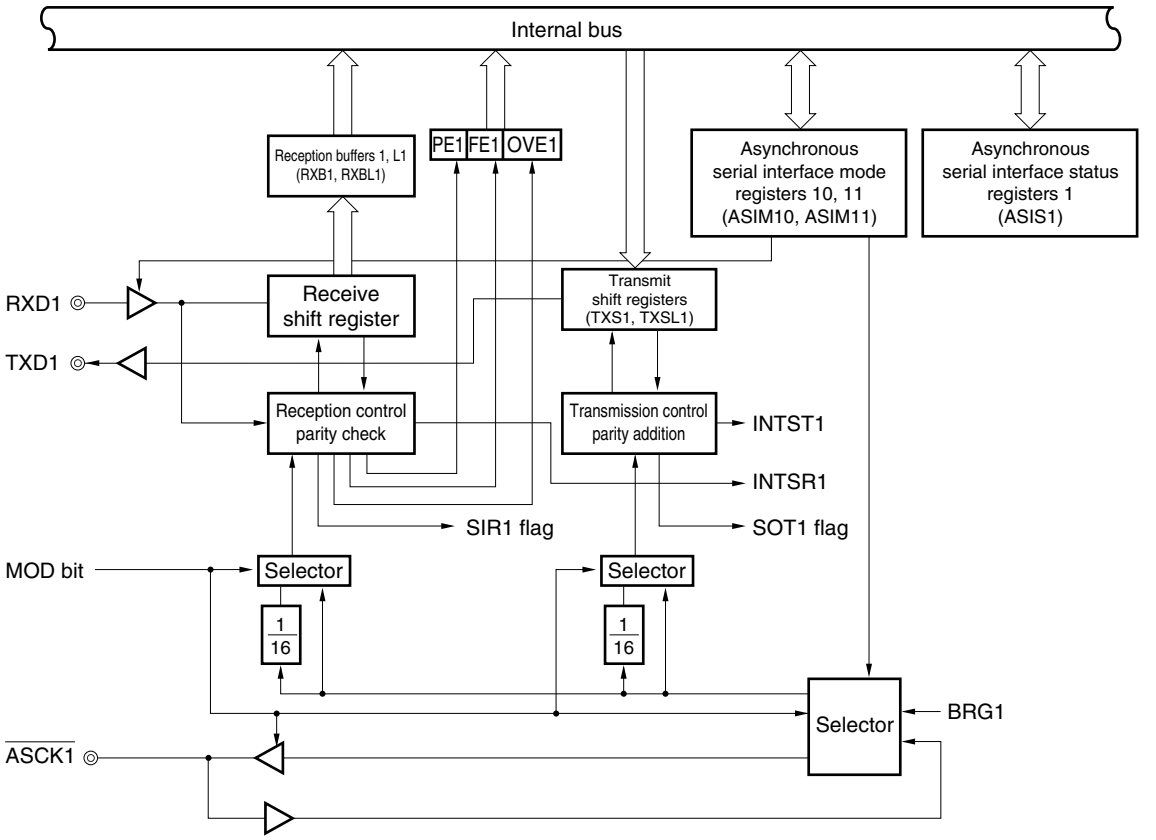
(6) Addition of transmission control parity

A transmission operation is controlled by adding a start bit, parity bit, or stop bit to the data that is written to the TXS1 or TXSL1 register, according to the contents set in the ASIM10, ASIM11 registers.

(7) Selector

The selector selects the serial clock source.

Figure 10-16. Block Diagram of Asynchronous Serial Interface 1



Remark The TXD1, RXD1, and $\overline{\text{ASCK1}}$ pins function alternately as the SO1, SI1, and $\overline{\text{SCK1}}$ pins.

10.3.3 Control registers

Because UART1 shares its pins with CSI1, the UART1 mode must be preset by using the PMC3 and RFC3 registers (refer to **10.1.1 Selecting UART1 or CSI1 mode**).

(1) Asynchronous serial interface mode register 10 (ASIM10)

The ASIM10 register is an 8-bit register that controls the UART1 transfer operation.

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

Cautions 1. If any bits other than the RXE1 bit of the ASIM10 register are changed during UART1 transmission or reception, the UART1 operation cannot be guaranteed.

2. Set bits other than the RXE1 bit of the ASIM10 register when the UART1 operation is stopped (when RXE1 = 0 and transmission is completed). Change the port 3 mode control register (PMC3) after setting the communication mode in the bits other than the RXE1 bit of the ASIM10 register.

3. In the case of serial clock output in the clocked (synchronous) mode, ensure that nodes do not output to one another causing conflict.

	7	<6>	5	4	3	2	1	0	Address	After reset
ASIM10	1	RXE1	PS1	PS0	CL	SL	0	SCLS	FFFFFA28H	81H

Bit position	Bit name	Function
6	RXE1	Enables/disables reception. 0: Disables reception 1: Enables reception
5, 4	PS1, PS0	Specify parity bit length

PS1	PS0	Operation
0	0	No parity, extension bit operation
0	1	0 parity Transmit side → Transmission with parity bit = 0 Receive side → No parity error generated during reception
1	0	Odd parity
1	1	Even parity

Bit position	Bit name	Function
3	CL	Specifies character length of transmit data (1 frame). 0: 7 bits 1: 8 bits
2	SL	Specifies stop bit length of transmit data. 0: 1 bit 1: 2 bits
0	SCLS	Specifies serial clock source.

SCLS	Operation	
	In asynchronous mode	In synchronous mode
0	Internal baud rate generator	External clock input
1		

(2) Asynchronous serial interface mode register 11 (ASIM11)

The ASIM11 register is an 8-bit register that controls the UART1 transfer mode.

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

	7	6	5	4	3	2	1	0	Address	After reset
ASIM11	0	0	0	0	MOD	UMST	UMSR	EBS	FFFFFA2AH	00H

Bit position	Bit name	Function
3	MOD	Specifies operation mode (asynchronous/synchronous mode) 0: Asynchronous mode 1: Synchronous mode
2	UMST	Specifies number of continuous frame transmissions. 0: 1-frame data transmission 1: 2-frame continuous data transmission
1	UMSR	Specifies number of continuous frame receptions. 0: 1-frame data reception 1: 2-frame continuous data reception
0	EBS	Specifies extension bit operation for transmit/receive data when no parity is specified (PS0 = PS1 = 0). 0: Disables extension bit addition 1: Enables extension bit addition When the extension bit is specified, 1 data bit is added on top of the 8 bits of transmit/receive data, enabling 9-bit data communication. Extension bit specification is valid only when no parity (ASIM10 register's PS0 bit = PS1 bit = 0) and 1-frame data transmission (UMST = 0) are specified. When 0 parity, odd parity, or even parity are specified, or when 2-frame continuous data transmission (UMST bit = 1) is specified, the EBS bit setting becomes invalid and extension bit addition is not performed. Extension bit addition (EBS bit = 1) and 2-frame continuous data reception (UMSR bit = 1) cannot be set simultaneously.

(3) Asynchronous serial interface status register 1 (ASIS1)

The ASIS1 register is a register that is configured of a UART1 transmission status flag (SOT1), reception status flag (SIR1), a bit (RB8) indicating the 9th bit when extension bit addition is enabled, and 3-bit error flags (PE1, FE1, OVE1) that indicate the error status at reception end.

The status flag that indicates reception errors always indicates the most recent error status. In other words, if the same error occurs several times before receive data is read, this flag holds only the status of the error that occurred last.

Each time the ASIS1 register is read after a reception completion interrupt (INTSR1), read the reception buffer (RXB1 or RXBL1). The error flag is cleared when the reception buffer (RXB1 or RXBL1) is read.

Also, clear the error flag by reading the reception buffer (RXB1 or RXBL1) when a reception error occurs.

This register is read-only, in 8-bit or 1-bit units.

	<7>	<6>	5	4	3	<2>	<1>	<0>	Address	After reset
ASIS1	SOT1	SIR1	0	RB8	0	PE1	FE1	OVE1	FFFFFA2CH	00H

Bit position	Bit name	Function
7	SOT1	Status flag indicating transmission status. 0: Transmission end timing (when INTST1 is generated) 1: Indicates transmission status[Note] **Note** The transmission status is the status until the specified number of stop bits has been transmitted following write operation to the transmit register. During 2-frame continuous transmission, this status is until the stop bit of the 2nd frame has been transmitted.
6	SIR1	Status flag indicating reception status. 0: Reception end timing (when INTSR1 is generated) 1: Indicates reception status[Note] **Note** The reception status is the status until stop bit detection from the start bit detection timing.
4	RB8	Indicates contents of receive data extension bit (1 bit) when 9-bit extended format is specified (EBS bit of ASIM11 register = 1)
2	PE1	Status flag indicating parity error 0: Processing to read data from reception buffer 1: When transmit parity and receive parity don't match **Caution No parity error is generated if no parity is specified or 0 parity is specified by the PS1, PS0 bits of the ASIM10 register.**
1	FE1	Status flag indicating framing error 0: Processing to read data from reception buffer 1: When stop bit is not detected
0	OVE1	Status flag indicating overrun error 0: Processing to read data from reception buffer 1: When UART1 has completed next reception processing prior to loading receive data from reception buffer Since the contents of the receive shift register are transferred to the reception buffer (RXB1, RXBL1) every time 1 frame is received, the next receive data is overwritten to the reception buffer (RXB1, RXBL1) and the previous receive data is discarded.

(4) 2-frame continuous reception buffer register 1 (RXB1)/receive buffer register L1 (RXBL1)

The RXB1 register is a 16-bit buffer register that holds receive data (during 2-frame continuous reception (UMSR bit of ASIM11 register = 1), during 9-bit extended data reception (EBS bit of ASIM11 register = 1)). During 7 or 8 bit character reception, 0 is stored in the MSB.

For 16-bit access to this register, specify RXB1, and for access to the lower 8 bits, specify RXBL1.

In the receive enabled status, receive data is transferred from the receive shift register to the reception buffer in synchronization with the end of shift-in processing for 1 frame of data.

The reception completion interrupt request (INTSR1) is generated upon transfer of data to the reception buffer (when 2-frame reception is specified, reception buffer transmission of the second frame).

In the reception disabled status, transfer processing to the reception buffer is not performed even if shift-in processing for 1 frame of data has been completed, and the contents of the reception buffer are held.

Neither is a reception completion interrupt request generated.

The RXB1 register can be read in 16-bit units, and the RXBL1 register can be read in 8-bit units.

[2-frame continuous reception buffer register 1]

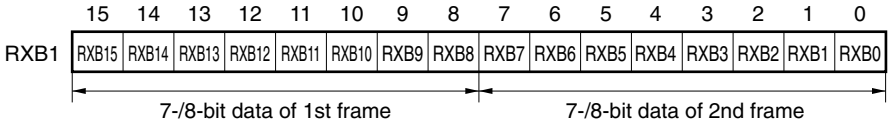
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
RXB1	RXB15	RXB14	RXB13	RXB12	RXB11	RXB10	RXB9	RXB8	RXB7	RXB6	RXB5	RXB4	RXB3	RXB2	RXB1	RXB0	FFFFFA20H	Undefined

[Receive buffer register L1]

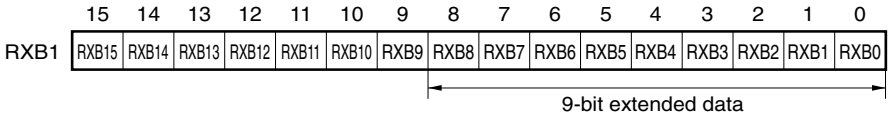
	7	6	5	4	3	2	1	0	Address	After reset
RXBL1	RXB7	RXB6	RXB5	RXB4	RXB3	RXB2	RXB1	RXB0	FFFFFA22H	Undefined

Bit position	Bit name	Function
15 to 0	RXB15 to RXB0	Stores receive data. 0 can be read for the RXB1 register when 7 or 8 bit character data is received. When an extension bit is set during 9 bit character data reception, the extension bit (RXB8) is stored in RB8 of the ASIS1 register simultaneously with saving to the reception buffer. 0 can be read for the RXB7 bit of the RXBL1 register during 7 bit character data reception.

(a) When 2-frame continuous reception is set



(b) When 9-bit extension reception is set



When 9-bit extension is set, the extension bit (RXB8) is stored in the RB8 bit of the ASIS1 register simultaneously with saving to the reception buffer.

(c) Cautions

<1> Operation upon occurrence of overrun error during 2-frame continuous reception

- **During normal operation**

  Reception completion interrupt (INTSR1) generated at end of reception of 2nd frame, no error

RXD1	Frame 1	Frame 2

- **Reception of 3rd frame started before performing reception processing**

  Reception completion interrupt (INTSR1) generated at end of reception of 2nd frame, no error

RXD1	Frame 1	Frame 2

  Reception interrupt not generated at end of reception of 3rd frame, occurrence of error

RXD1	Frame 3	Frame 3

  Value of OVE1 bit of ASIS1 register becomes 1.

- **Start of reception of 3rd frame and 4th frame before performing reception processing**

  Reception completion interrupt (INTSR1) generated at end of reception of 2nd frame, no error

RXD1	Frame 1	Frame 2

  No reception completion interrupt generated at end of reception of 3rd frame, occurrence of error

RXD1	Frame 3	Frame 3

  Value of OVE1 bit of ASIS1 register becomes 1.

  Reception completion interrupt (INTSR1) generated at end of reception of 4th frame, no error

RXD1	Frame 3	Frame 4

  Value of OVE1 frame of ASIS1 register remains 1.

- **Start of reception of 3rd frame before performing reception processing, start of reception of 4th frame after reception processing**

  Reception completion interrupt (INTSR1) generated at end of reception of 2nd frame, no error

RXD1	Frame 1	Frame 2

  Reception completion interrupt not generated at end of reception of 3rd frame, occurrence of error

RXD1	Frame 3	Frame 3

  Value of OVE1 bit of ASIS1 register becomes 1.

  Value of OVE1 flag becomes 0 during reception processing.

  Reception completion interrupt (INTSR1) generated at end of reception of 4th frame, no error

RXD1	Frame 3	Frame 4

  No occurrence of error

(5) 2-frame continuous transmission shift register 1 (TXS1)/transmit shift register L1 (TXSL1)

The TXS1 register is a 9-bit/2-frame continuous transmission processing shift register. Transmission is started by writing data to this register.

A transmission completion interrupt request (INTST1) is generated in synchronization with the end of transmission of 1 frame or 2 frames including the TXS1 data.

For 16-bit access to this register, specify TXS1, and for access to the lower 8 bits, specify TXSL1.

The TXS1 register is write-only in 16-bit units, and the TXSL1 register is write-only in 8-bit units.

Caution TXS1, TXSL1 can be read, but since shifting is done in synchronization with the shift clock, the data that is read cannot be guaranteed.

[2-frame continuous transmission shift register 1]

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
TXS1	TXS15	TXS14	TXS13	TXS12	TXS11	TXS10	TXS9	TXS8	TXS7	TXS6	TXS5	TXS4	TXS3	TXS2	TXS1	TXS0	FFFFFA24H	Undefined

[Transmit shift register L1]

	7	6	5	4	3	2	1	0	Address	After reset
TXSL1	TXS7	TXS6	TXS5	TXS4	TXS3	TXS2	TXS1	TXS0	FFFFFA26H	Undefined

Bit position	Bit name	Function
15 to 0	TXB15 to TXB0	Write transmit data.

10.3.4 Interrupt requests

The following two types of interrupt request are generated from UART1.

- Reception completion interrupt (INTSR1)
- Transmission completion interrupt (INTST1)

The reception completion interrupt has higher default priority than the transmission completion interrupt.

Table 10-5. Default Priority of Generated Interrupts

Interrupt	Priority
Reception completion	1
Transmission completion	2

(1) Reception completion interrupt (INTSR1)

In the reception enabled state, the reception completion interrupt (INTSR1) is generated when data in the receive shift register undergoes shift-in processing and is transferred to the reception buffer.

The reception completion interrupt request (INTSR1) is generated following stop-bit sampling and upon the occurrence of an error.

In the reception disabled state, no reception completion interrupt is generated.

Caution A reception completion interrupt (INTSR1) is generated when the last bit of receive data (stop bit) is sampled.

(2) Transmission completion interrupt (INTST1)

Since UART1 does not have a transmit buffer, a transmission completion interrupt request (INTST1) is generated when one frame of data containing 7-bit or 8-bit characters or two frames of data containing 9-bit characters are shifted out from the transmit shift register (TXS1, TXSL1).

10.3.5 Operation

(1) Data format

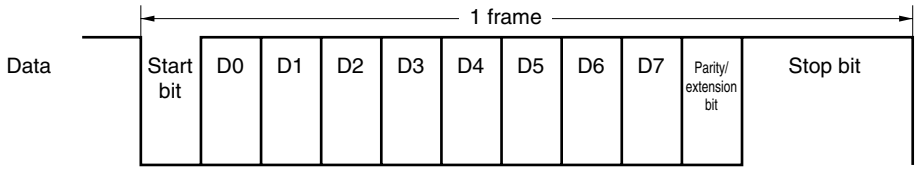
Full-duplex serial data is transmitted and received.

Figure 10-17 shows the format of transmit/receive data. One data frame consists of a start bit, character bits, a parity bit, and a stop bit(s). When 2 data frame transfer is set, both frames have the above-described format.

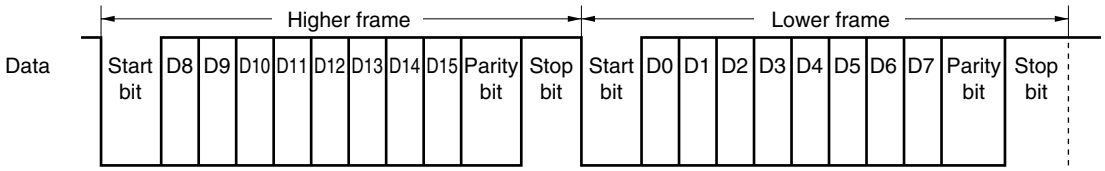
Specification of the character bit length in one data frame, parity selection, and specification of the stop bit length is done using asynchronous serial interface mode register 10 (ASIM10). Specification of the number of frames and specification of the extension bit is mode using asynchronous serial interface mode register 11 (ASIM11). Data is transmitted LSB first.

Figure 10-17. Asynchronous Serial Interface Transmit/Receive Data Format

(a) 1-frame format



(b) 2-frame format



- Start bit ... 1 bit
- Character bits ... 7 or 8 bits
- Parity bit ... Even parity, odd parity, 0 parity, or no parity
- Stop bit ... 1 or 2 bits

Caution The extension bit is invalid in the 2-frame continuous mode or when a parity bit is added.

Table 10-6. ASIM10, ASIM11 Register Settings and Data Format

ASIM10, ASIM11 Register Settings					Data Format				
CL Bit	PS1 Bit	PS0 Bit	SL Bit	EBS Bit	D0 to D6	D7	D8	D9	D10
0	0	0	0	0	DATA	Stop bit	—	—	—
0	Other than PS1 = PS0 = 0				DATA	Parity bit	Stop bit	—	—
1	0	0			DATA	DATA	Stop bit	—	—
1	Other than PS1 = PS0 = 0				DATA	DATA	Parity bit	Stop bit	—
0	0	0	1	0	DATA	Stop bit	Stop bit	—	—
0	Other than PS1 = PS0 = 0				DATA	Parity bit	Stop bit	Stop bit	—
1	0	0			DATA	DATA	Stop bit	Stop bit	—
1	Other than PS1 = PS0 = 0				DATA	DATA	Parity bit	Stop bit	Stop bit
0	0	0	0	1	DATA	Stop bit	—	—	—
0	Other than PS1 = PS0 = 0				DATA	Parity bit	Stop bit	—	—
1	0	0			DATA	DATA	DATA	Stop bit	—
1	Other than PS1 = PS0 = 0				DATA	DATA	Parity bit	Stop bit	—
0	0	0	1	1	DATA	Stop bit	Stop bit	—	—
0	Other than PS1 = PS0 = 0				DATA	Parity bit	Stop bit	Stop bit	—
1	0	0			DATA	DATA	DATA	Stop bit	Stop bit
1	Other than PS1 = PS0 = 0				DATA	DATA	Parity bit	Stop bit	Stop bit

(2) Transmission operation

The transmission operation is started by writing data to 2-frame continuous transmission shift register 1 (TXS1)/transmit shift register L1 (TXSL1).

Following data write, the start bit is transmitted from the next shift timing.

Since the UART1 does not have a CTS (transmission enable signal) input pin, use a port when the other party confirms the reception enabled status.

(a) Transmission operation start

The transmission operation is started by writing transmit data to 2-frame continuous transmission shift register 1 (TXS1)/transmit shift register L1 (TXSL1). Then data is output in sequence from LSB to the TXD1 pin (transmission in sequence from the start bit). A start bit, parity bit, and stop bit(s) are automatically added.

(b) Transmission interrupt request

When the transmit shift register becomes empty upon completion of the transmission of 1 or 2 frames of data, a transmission completion interrupt request (INTST1) is generated. The INTST1 interrupt generation timing differs depending on the specification of the stop bit length. The INTST1 interrupt is generated at the same time that the last stop bit is output.

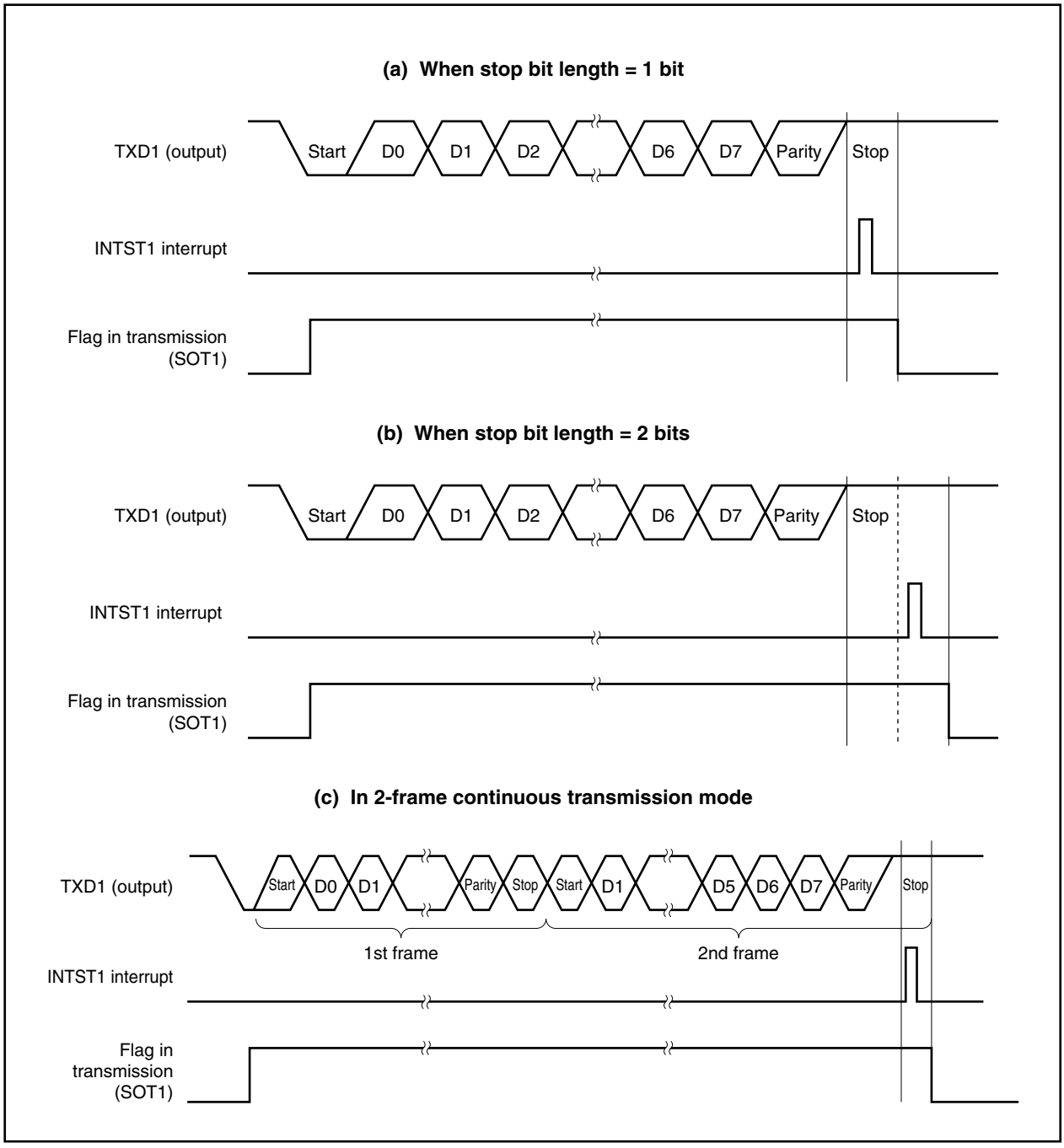
The transmission operation remains stopped until the data to be transmitted next has been written to the TXS1/TXSL1 registers.

Figure 10-18 shows the INTST1 interrupt generation timing.

Cautions 1. Normally, the transmission completion interrupt (INTST1) is generated when the transmit shift register becomes empty. However, if the transmit shift register has become empty due to input of $\overline{\text{RESET}}$, no transmission completion interrupt (INTST1) is generated.

2. No data can be written to the TXS1 or TXSL1 registers during a transmission operation until INTST1 is generated. Even if data is written, this does not affect the transmission operation.

Figure 10-18. Asynchronous Serial Interface Transmission Completion Interrupt Timing



(3) Continuous transmission of 3 or more frames

In addition to the 1-frame/2-frame transmission function, UART1 also enables continuous transmission of 3 or more frames, using the method shown below.

(a) How to continuously transmit 3 or more frames (when the stop bit is 1 bit (SL bit = 0))

Three frames can be continuously transmitted by writing transmit data to the TXS1/TXSL1 register in the period between the generation of the transmission completion interrupt request (INTST1) and $4 \times 2/f_{XX}$ before the output of the last stop bit.

The INTST1 interrupt becomes high level $2/f_{XX}$ after being output and returns to low level $2/f_{XX}$ later. TXS1/TXSL1 can only be written after the INTST1 interrupt level has fallen. The time from INTST1 interrupt generation to the completion of transmit data writing (t) is therefore indicated by the following expression.

t = (Time of one stop bit) $- (2 \times 2/f_{XX} + 4 \times 2/f_{XX})$

f_{XX} = Internal system clock

Caution $4 \times 2/f_{XX}$ has a margin of double the clock that can actually be used for operation.

Example Count clock frequency = 32 MHz = 32,000,000 Hz
Target baud rate in synchronous mode = 9,600 bps

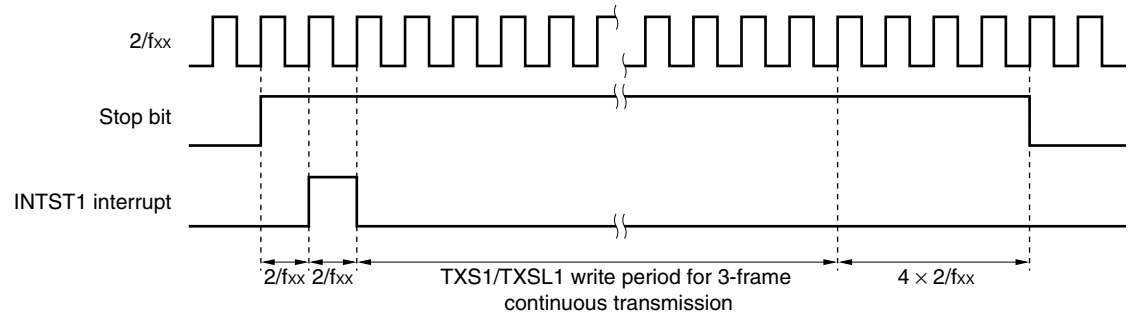
$t = (1/9615.385) - ((4 + 8)/32{,}000{,}000)$
$= 104.000 - 0.375$
$= 103.625$ [μs]

Therefore, be sure to write transmit data to TXS1/TXSL1 within 103 μs of the generation of the INTST1 interrupt.

Note, however, that because writing to TXS1/TXSL1 may be delayed depending on the priority order of the interrupt or the interrupt servicing time, be sure to allow sufficient time for writing transmit data after the INTST1 interrupt has been generated. If there is not enough time for continuous transmission due to a delay in writing to TXS1/TXSL1, a 1-bit high level is transmitted.

Note also that if the stop bit length is 2 bits (SL = 1), the INTST1 interrupt will be generated when the second stop bit is output.

Figure 10-19. Continuous Transmission of 3 or More Frames



(4) Reception operation

The reception wait status is entered by setting the RXE1 bit of the ASIM10 register to 1. To start the reception operation, first perform start bit detection. Start bit detection is done by performing sampling of the RXD1 pin. When the reception operation is started, serial data is stored in the receive shift register in order at the set baud rate. Each time reception of 2 frames or 1 frame of RXB1 or RXBL1 data has been completed, a reception completion interrupt (INTSR1) is generated. Receive data is transmitted from the reception buffer (RXB1/RXBL1) to memory when this interrupt is serviced.

(a) Reception enabled status

The reception operation is enabled by setting (1) the RXE1 bit of the ASIM10 register.

- RXE1 = 1: Reception enabled status
- RXE1 = 0: Reception disabled status

In the reception disabled status, the reception hardware is in standby in an initialized state. At this time, no reception completion interrupt is generated, and the contents of the reception buffer are held.

(b) Start of reception operation

The reception operation is started by detection of the start bit.

- **In asynchronous mode (MOD bit of ASIM11 register = 0)**
  The RXD1 pin is sampled using the serial clock from the baud rate generator. After 8 serial clocks have been output following detection of the falling edge of the RXD1 pin, the RXD1 pin is again sampled. If a low level is detected at this time, the falling edge of the RXD1 pin is interpreted as a start bit, the operation shifts to reception processing, and the RXD1 pin input is sampled from this point on in units of 16 serial clock output.
  If the high level is detected during sampling after 8 serial clocks from detection of the falling edge of the RXD1 pin, this falling edge is not recognized as a start bit. The serial clock counter that generates the sample timing is initialized and stops, and input of the next falling edge is waited for.

- **In synchronous mode (MOD bit of ASIM11 register = 1)**
  The RXD1 pin is sampled using the serial clock from the baud rate generator or at the rising edge of serial clock input/output. If the RXD1 pin is low level at this time, this is interpreted as a start bit and reception processing starts.

If reception data is interrupted at the fixed low level during reception, reception of this receive data (including error detection) is completed and reception completion interrupt is generated. However, even if the RXD line is fixed at low level, the next reception operation is not started (start bit detection is not performed).
Be sure to set the high level when restarting the reception operation. If the high level is not set, the start bit detection position becomes undefined, and correct reception operation cannot be performed.

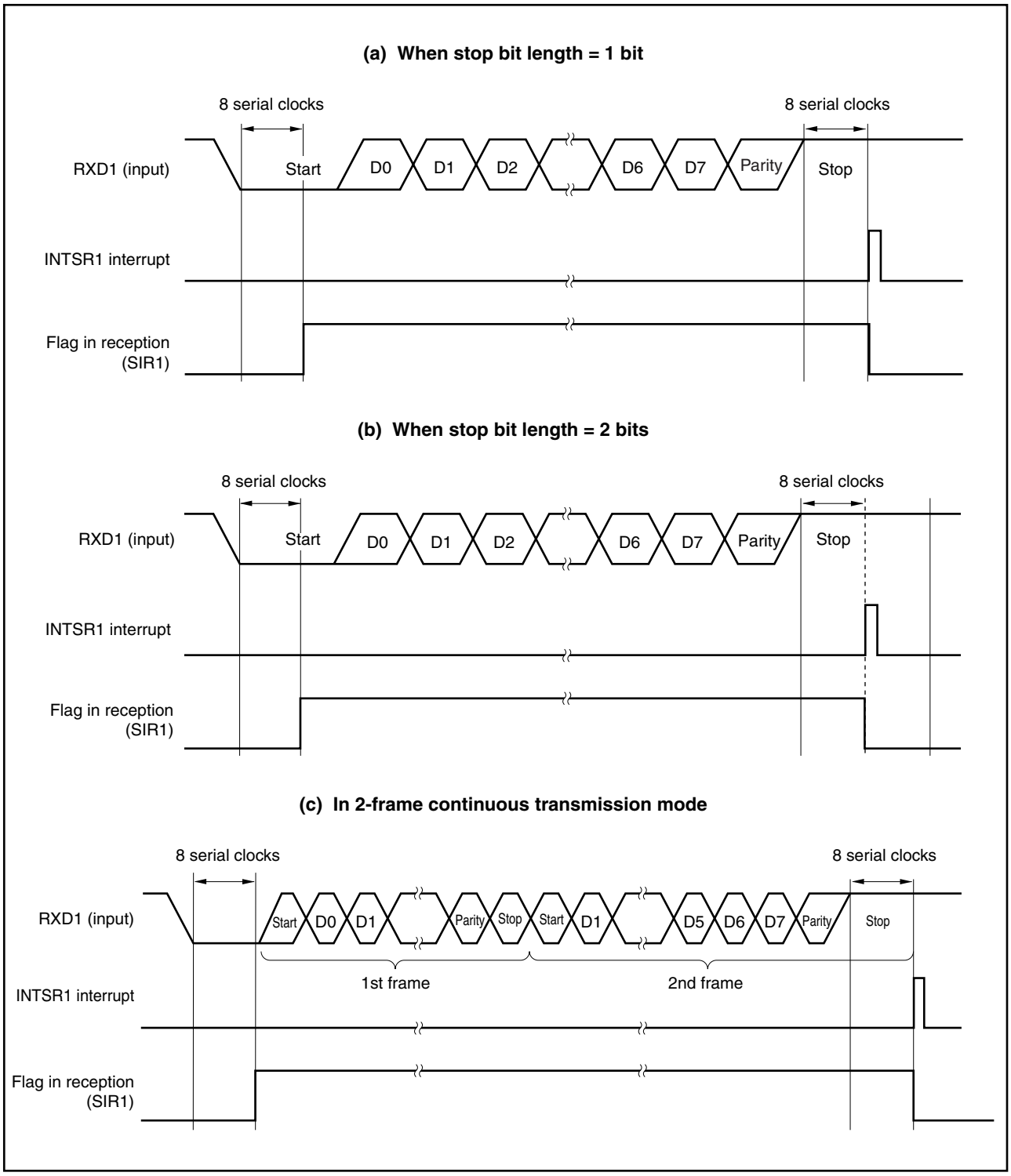
(c) Reception completion interrupt request

When reception of one frame of data has been completed (stop bit detection) when the RXE1 bit of the ASIM10 register = 1, the receive data in the shift register is transferred to RXB1/RXBL1 and a reception completion interrupt request (INTSR1) is generated after 1 frame or 2 frames of data have been transferred to RXB1/RXBL1.

A reception completion interrupt is also generated upon detection of an error.

When the RXE1 bit = 0 (reception disabled), no reception completion interrupt is generated.

★ **Figure 10-20. Asynchronous Serial Interface Reception Completion Interrupt Timing**



Cautions 1. Even if a reception error occurs, be sure to read 2-frame continuous reception buffer register 1 (RXB1)/receive buffer register 1 (RXBL1). If the RXB1 or RXBL1 register is not read, an overrun error will occur at the next data reception, and the reception error state will continue indefinitely.

2. Reception is always performed with a stop bit length of 1 bit. A second stop bit is ignored.

(5) Reception errors

The flags for the three types of errors: parity errors, framing errors, and overrun errors, are affected in synchronization with reception operation. As a result of data reception, the PE1, FE1, and OVE1 flags of the ASIS1 register are set (1) and a reception completion interrupt request (INTSR1) is generated at the same time.

The contents of error that occurred during reception can be detected by reading the contents of the PE1, FE1, and OVE1 flags of the ASIS1 register during the INTSR1 interrupt servicing.

The contents of the ASIS1 register are reset (0) by reading the ASIS1 register (if the next receive data contains an error, the corresponding error flag is set (1)).

Table 10-7. Reception Error Causes

Error Flag	Reception Error	Causes
PE1	Parity error	The parity specification during transmission did not match the parity of the reception data
FE1	Framing error	No stop bit was detected
OVE1	Overrun error	The reception of the next data was completed before data was read from the reception buffer

(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 at the transmission and reception sides.

(a) Even parity

<1> During transmission

The parity bit is controlled so that 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

<2> 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

<1> 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

<2> 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.

10.3.6 Synchronous mode

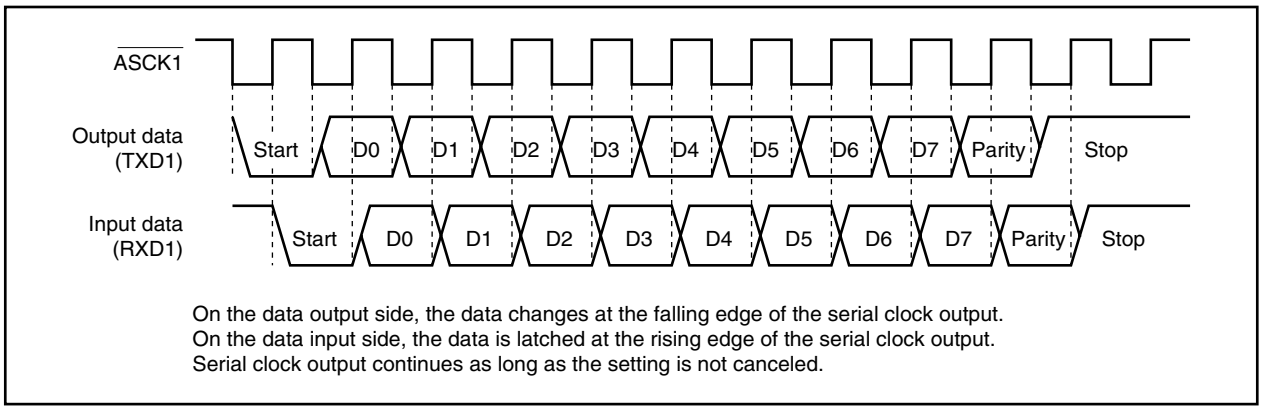
The synchronous mode can be set with the $\overline{\text{ASCK1}}$ pin, which is the serial clock I/O pin.

The synchronous mode is set with the MOD bit of the ASIM11 register, and the serial clock to be used for synchronization is selected with the SCLS bit of the ASIM10 register.

In the synchronous mode, external clock input is selected when the value of the SCLS bit is 0 (default), and the serial clock output is selected in the case of all other settings. Therefore, when performing settings, make sure that outputs between connection nodes do not conflict.

In the synchronous mode, the falling edge of the serial clock is used as the transmission timing, and the rising edge as the reception timing, but transmit data is output with a delay of 1 system clock (serial clock) (in the external clock synchronous mode, the maximum delay is 2.5 system clocks).

Figure 10-21. Transmission/Reception Timing in Synchronous Mode



On the data output side, the data changes at the falling edge of the serial clock output.
On the data input side, the data is latched at the rising edge of the serial clock output.
Serial clock output continues as long as the setting is not canceled.

Figure 10-22. Transmission/Reception Timing Chart for Synchronous Mode (1/3)

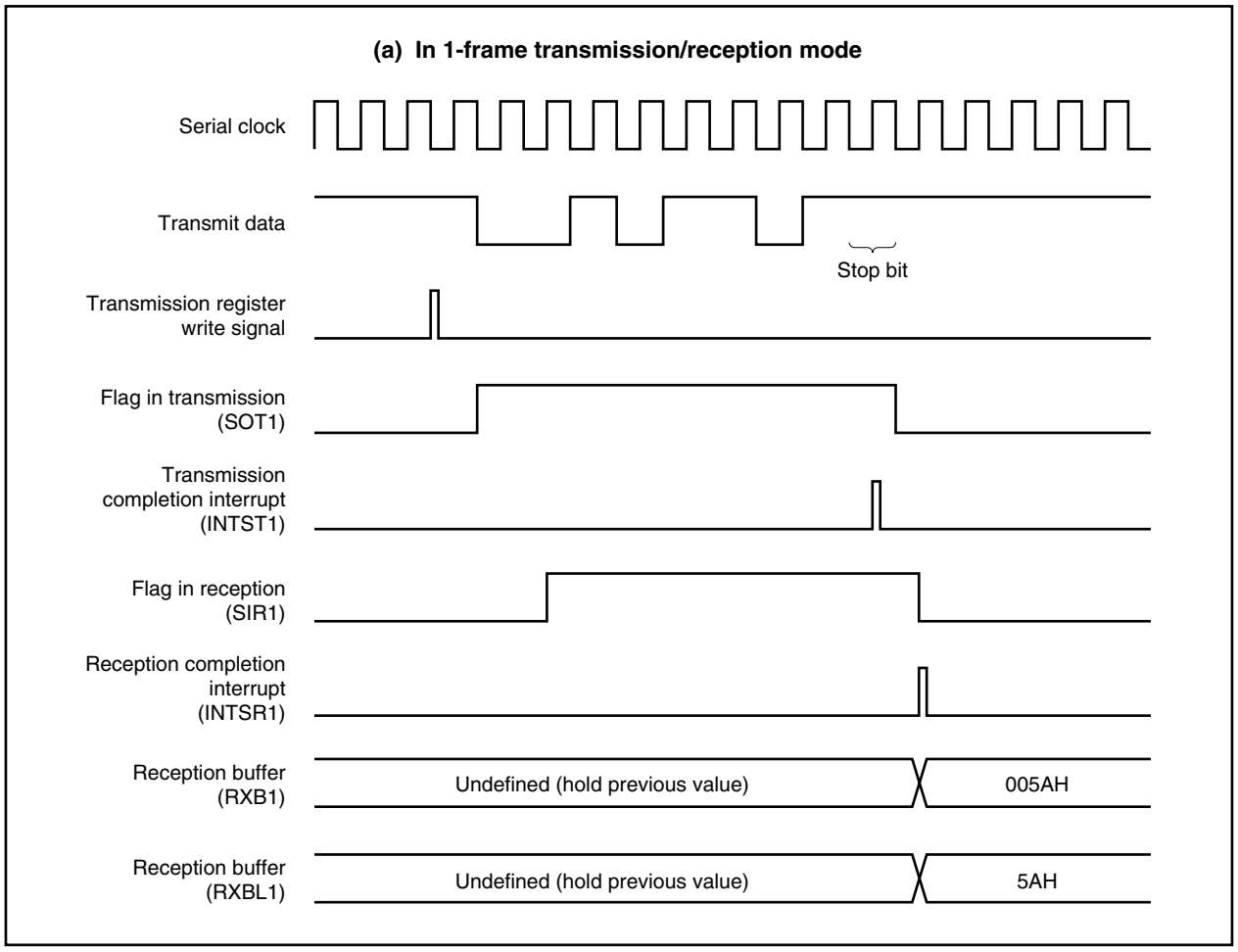


Figure 10-22. Transmission/Reception Timing Chart for Synchronous Mode (2/3)

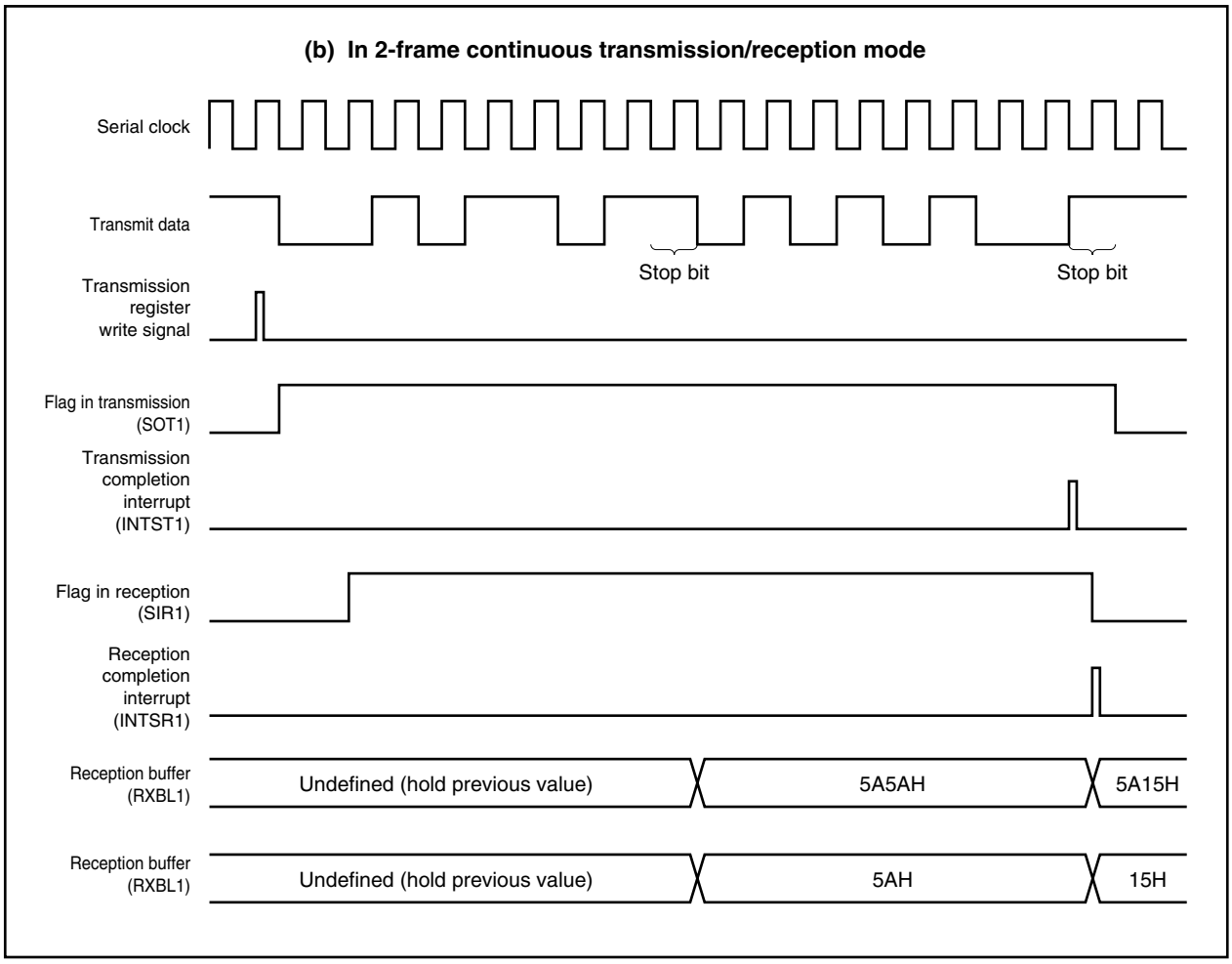
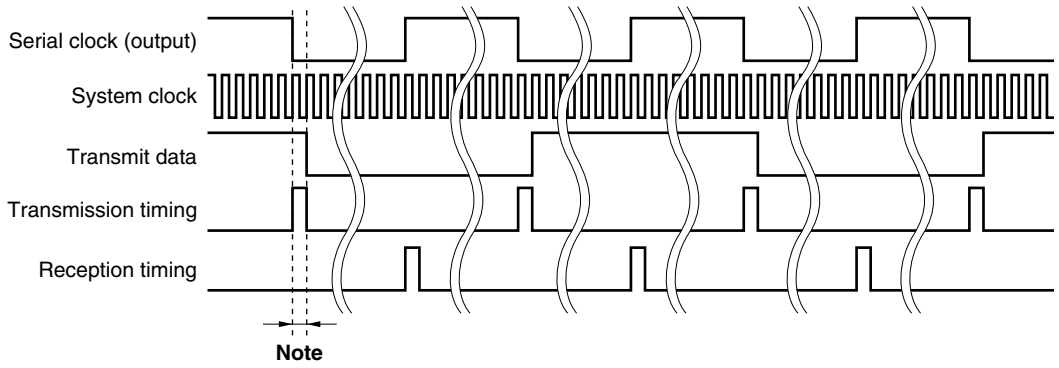


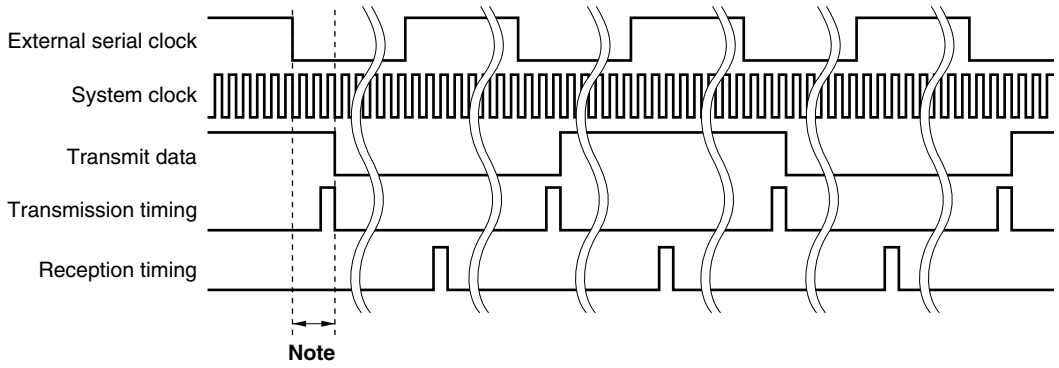
Figure 10-22. Transmission/Reception Timing Chart for Synchronous Mode (3/3)

(c) Transmission/reception timing and transmit data timing during serial clock output



Note The transmit data is delayed by 1 system clock in relation to the serial clock.

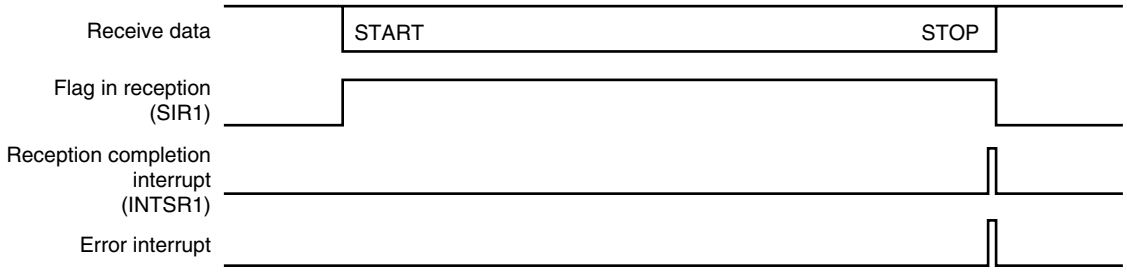
(d) Transmission/reception timing and transmit data timing using external serial clock



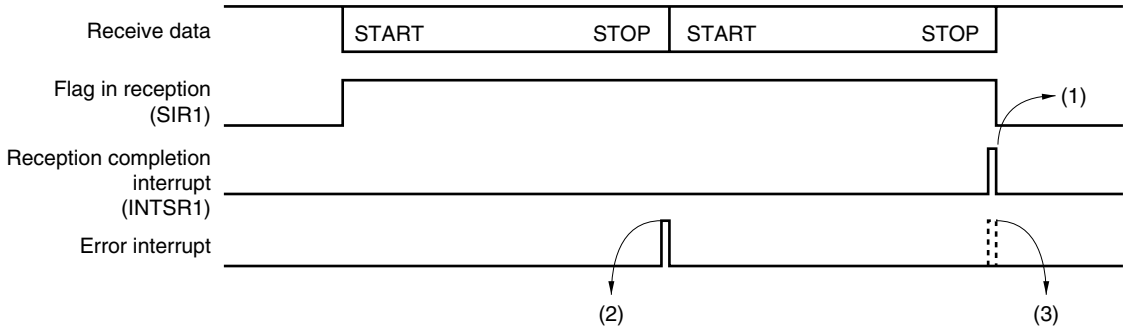
Note Since, during external serial clock synchronization, synchronization is done with the internal system clock when feeding the external serial clock to the internal circuit, a delay ranging from 1 system clock to a maximum of 2.5 system clocks results.

Figure 10-23. Reception Completion Interrupt and Error Interrupt Generation Timing During Synchronous Mode Reception

(a) During normal operation (in 1-frame reception mode)



(b) In 2-frame continuous reception mode



<Explanation>

(1) If the start bit of the second frame is not detected, no reception completion interrupt is generated.

(2) If an error occurs in the first frame, an error interrupt is generated following detection of the stop bit of the first frame (at the calculated position).

(3) If an error occurs in the second frame, an error interrupt is generated simultaneously with a reception completion interrupt.

If an error occurs in the first frame, no error interrupt is generated even if an error occurs in the second frame.

10.3.7 Dedicated baud rate generator 1 (BRG1)

(1) Configuration of baud rate generator 1 (BRG1)

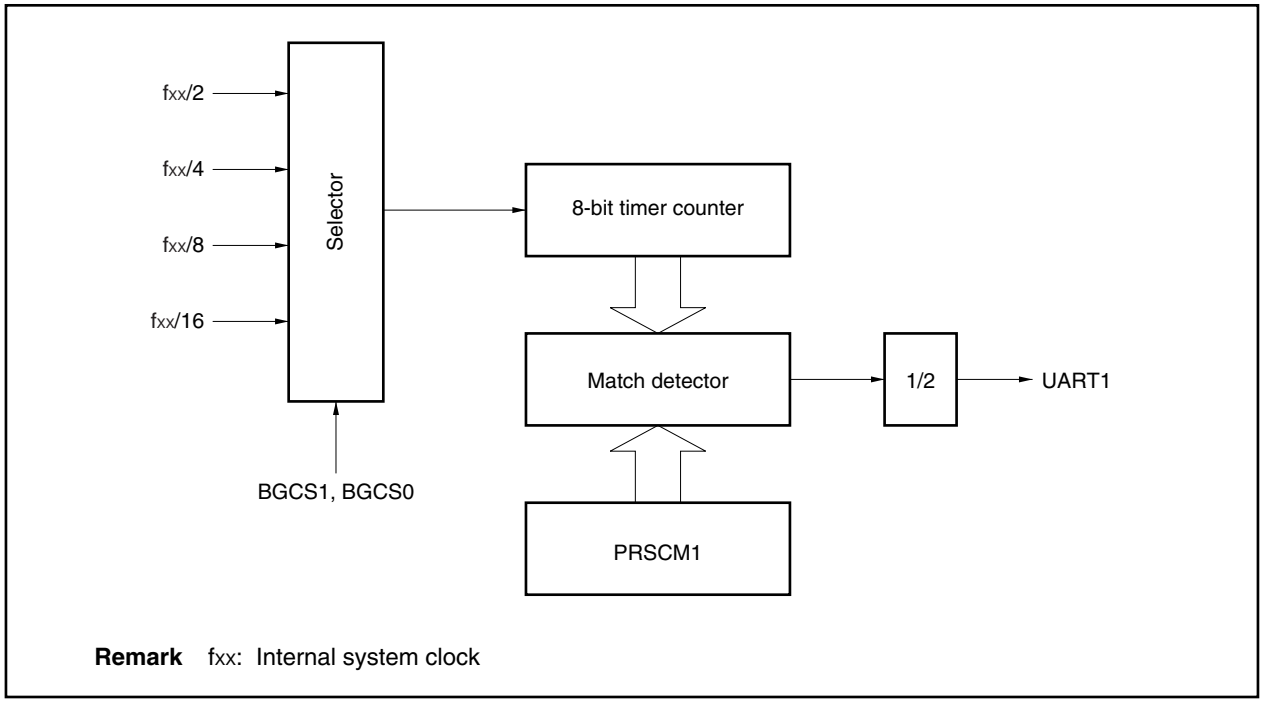
For UART1, the serial clock can be selected from the dedicated baud rate generator output or internal system clock (fxx) for each channel.

The serial clock source is specified by register ASIM10.

If dedicated baud rate generator output is specified, BRG1 is selected as the clock source.

Since the same serial clock can be shared for transmission and reception for one channel, baud rate is the same for the transmission/reception.

Figure 10-24. Block Diagram of Baud Rate Generator 1 (BRG1)



Remark fxx: Internal system clock

(2) Dedicated baud rate generator 1 (BRG1)

BRG1 is configured of an 8-bit timer counter for baud rate signal generation, a prescaler mode register that controls the generation of the baud rate signal (PRSM1), a prescaler compare register that sets the value of the 8-bit timer counter (PRSCM1), and a prescaler.

(a) Input clock

The internal system clock (fxx) is input to BRG1.

(b) Prescaler mode register 1 (PRSM1)

The PRSM1 register controls generation of the UART1 baud rate signal.

These registers can be read/written in 8-bit or 1-bit units.

Cautions 1. Do not change the values of the BGCS1 and BGCS0 bits during transmission/ reception operations.

2. Set PRSM1 bits other than the UARTCE1 bit prior to setting the UARTCE1 bit to 1.

	<7>	6	5	4	3	2	1	0	Address	After reset
PRSM1	UARTCE1	0	0	0	0	0	BGCS1	BGCS0	FFFFFA2EH	00H

Bit position	Bit name	Function
7	UARTCE1	Enables baud rate counter operation. 0: Stops baud rate counter operation and fixes baud rate output signal to 0. 1: Enables baud rate counter operation and starts baud rate output.
1, 0	BGCS1, BGCS0	Selects count clock to baud rate counter.

BGCS1	BGCS0	Count clock selection
0	0	fxx/2
0	1	fxx/4
1	0	fxx/8
1	1	fxx/16

Remark fxx: Internal system clock

(c) Prescaler compare register 1 (PRSCM1)

PRSCM1 is an 8-bit compare register that sets the value of the 8-bit timer counter.

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

Cautions 1. The internal timer counter is cleared by writing to the PRSCM1 register. Therefore, do not overwrite the PRSCM1 register during a transmission operation.

2. Perform PRSCM1 register settings prior to setting the UARTCE1 bit to 1. If the contents of the PRSCM1 register are overwritten when the value of the UARTCE1 bit is 1, the cycle of the baud rate signal is not guaranteed.

3. Set the baud rate in the asynchronous mode to 153600 bps or lower. Set the baud rate in the synchronous mode to 1000000 bps or lower.

	7	6	5	4	3	2	1	0	Address	After reset
PRSCM1	PRSCM7	PRSCM6	PRSCM5	PRSCM4	PRSCM3	PRSCM2	PRSCM1	PRSCM0	FFFFFA30H	00H

(d) Baud rate generation

First, when the UARTCE1 bit of the PRSM1 register is overwritten by 1, the 8-bit timer counter for baud rate signal generation starts counting up with the clock selected by bits BGCS1 and BGCS0 of the PRSM1 register. The count value of the 8-bit timer counter is compared with the value of the PRSCM1 register, and if these values match, a timer count clock pulse of 1 cycle is output to the output controller for the baud rate.

The output controller for the baud rate reverses the baud rate signal in synchronization with the rising edge of the timer count clock when this pulse is “1”.

(e) Cycle of baud rate signal

The cycle of the baud rate signal is calculated as follows.

- **When setting value of PRSCM1 register is 00H**
  (Cycle of signal selected by bits BGCS1, BGCS0 of PRSM1 register) × 256 × 2
- **In cases other than above**
  (Cycle of signal selected by bits BGCS1, BGCS0 of PRSM1 register) × (setting value of PRSCM1 register) × 2

(f) Baud rate setting value

The formulas for calculating the baud rate in the asynchronous mode and the synchronous mode and the formula for calculating the error are as follows.

<1> Formula for calculating baud rate in asynchronous mode

$$\text{Baud rate} = \frac{f_{XX}}{2 \times m \times 2^k \times 16} \text{ [bps]}$$

f_{XX} = Internal system clock frequency [Hz]
= CPU clock/2 [Hz]
m: Setting value of PRSCM1 register ($1 \leq m \leq 256$[Note])
k: Value set by bits BGCS1, BGCS0 of PRSM1 register (k = 0, 1, 2, 3)

Note The setting of m = 256 is performed by writing 00H to the PRSCM1 register.

<2> Formula for calculating the baud rate in synchronous mode

$$\text{Baud rate} = \frac{f_{XX}}{2 \times m \times 2^k} \text{ [bps]}$$

f_{XX} = Internal system clock frequency [Hz]
= CPU clock/2 [Hz]
m: Setting value of PRSCM1 register ($1 \leq m \leq 256$[Note])
k: Value set by bits BGCS1, BGCS0 of PRSM1 register (k = 0, 1, 2, 3)

Note The setting of m = 256 is performed by writing 00H to the PRSCM1 register.

<3> Formula for calculating error

$$\text{Error [\%]} = \left(\frac{\text{Actual baud rate} - \text{Target baud rate}}{\text{Desired baud rate}} \right) \times 100$$

Example $(9{,}520 - 9{,}600)/9{,}600 \times 100 = -0.833$ [%]

Remark Actual baud rate: Baud rate with error
Target baud rate: Normal baud rate

<4> Baud rate setting example

In an actual system, the output of a prescaler module, etc. is connected to the input clock. Table 10-8 shows the baud rate generator setting data at this time.

Table 10-8. Baud Rate Generator Setting Data (BRG = fxx/2)

(a) When fxx = 32 MHz

Target Baud Rate		Actual Baud Rate		BGCSm Bit (m = 0, 1)	PRSCM1 Register Setting Value	Error
Synchronous Mode	Asynchronous Mode	Synchronous Mode	Asynchronous Mode			
4,800	300	4,807.692	300.4808	3	208	0.16
9,600	600	9,615.385	600.9615	3	104	0.16
19,200	1,200	19,230.77	1,201.923	3	52	0.16
38,400	2,400	38,461.54	2,403.846	3	26	0.16
76,800	4,800	76,923.08	4,807.692	3	13	0.16
153,600	9,600	153,846.2	9,615.385	2	13	0.16
166,400	10,400	166,666.7	10,416.67	1	24	0.16
307,200	19,200	307,692.3	19,230.77	1	13	0.16
614,400	38,400	615,384.6	38,461.54	0	13	0.16
Not possible	76,800	–	71,428.57	0	7	–6.99
Not possible	153,600	–	166,666.7	0	3	8.51

(b) When fxx = 40 MHz

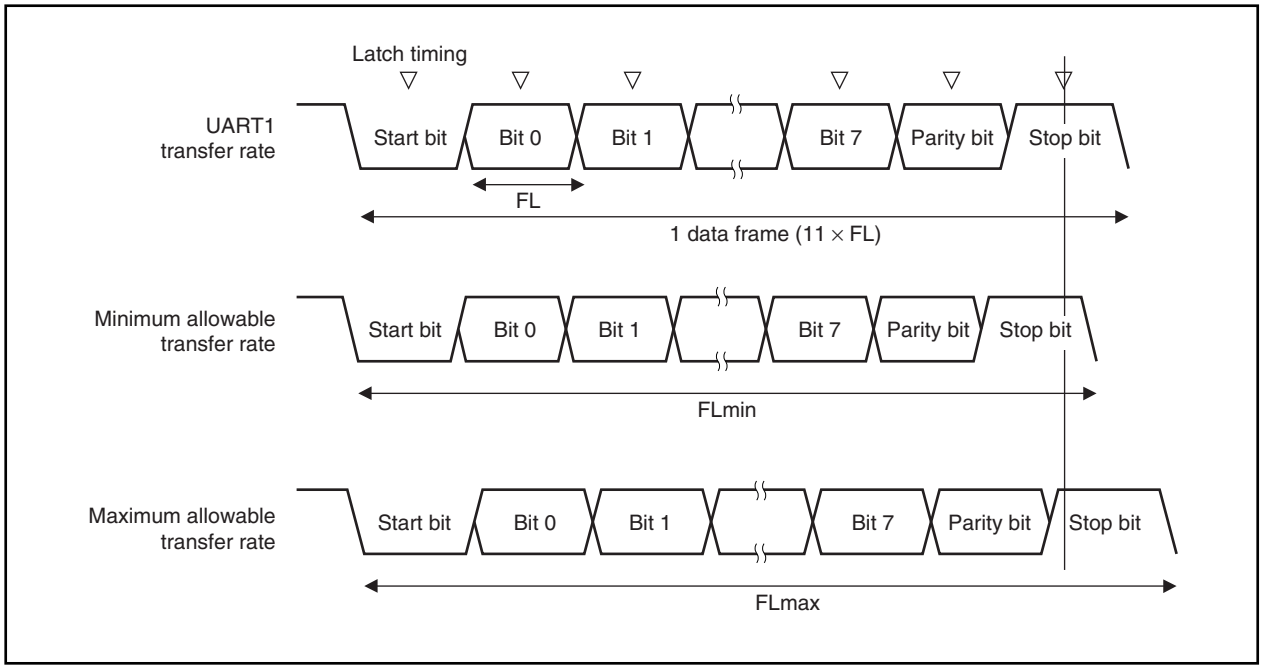
Target Baud Rate		Actual Baud Rate		BGCSm Bit (m = 0, 1)	PRSCM1 Register Setting Value	Error
Synchronous Mode	Asynchronous Mode	Synchronous Mode	Asynchronous Mode			
4,800	300	4,882.813	305.1758	3	256	1.73
9,600	600	9,615.385	600.9615	3	130	0.16
19,200	1,200	19,230.77	1,201.923	3	65	0.16
38,400	2,400	38,461.54	2,403.846	2	65	0.16
76,800	4,800	76,923.08	4,807.692	1	65	0.16
153,600	9,600	153,846.2	9,615.385	0	65	0.16
166,400	10,400	166,666.7	10,416.67	0	60	0.16
307,200	19,200	303,030.3	18,939.39	0	33	–1.36
614,400	38,400	625,000	39,062.5	0	16	1.73
Not possible	76,800	–	78,125	0	8	1.73
Not possible	153,600	–	156,250	0	4	1.73

(3) 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 10-25. Allowable Baud Rate Range During Reception



As shown in Figure 10-25, after the start bit is detected, the receive data latch timing is determined according to the counter that was set by the PRSCM1 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: UART1 baud rate
k: PRSCM1 register setting value
FL: 1-bit data length

When the latch timing margin is 2 clocks of fxx/2, the minimum allowable transfer rate (FLmin) is as follows (fxx: Internal system clock).

$$FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

Therefore, the transfer destination's maximum receivable baud rate (BRmax) is as follows.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k + 2} \text{ Brate}$$

Similarly, the maximum allowable transfer rate (FLmax) can be obtained as follows.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k + 2}{2 \times k} \times FL = \frac{21k - 2}{2 \times k} FL$$

$$FLmax = \frac{21k - 2}{20k} FL \times 11$$

Therefore, the transfer destination's minimum receivable baud rate (BRmin) is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k - 2} \text{ Brate}$$

(4) Transfer rate in 2-frame continuous reception

In 2-frame continuous reception, the timing is initialized by detecting the start bit of the second frame, so the transfer results are not affected.

10.4 Clocked Serial Interfaces 0, 1 (CSI0, CSI1)

10.4.1 Features

- High-speed transfer: Maximum 5 Mbps
- Half-duplex communications
- Master mode or slave mode can be selected
- Transmission data length: 8 bits or 16 bits can be set
- Transfer data direction can be switched between MSB first and LSB first
- Eight clock signals can be selected (7 master clocks and 1 slave clock)
- 3-wire type SOn: Serial transmit data output
  SIn: Serial receive data input
  $\overline{\text{SCK}}$n: Serial clock I/O
- Interrupt sources: 1 type
  - Transmission/reception completion interrupt (INTCSIn)
- Transmission/reception mode and reception-only mode can be specified
- Two transmit buffers (SOTBFn/SOTBFLn, SOTBn/SOTBLn) and two receive buffers (SIRBn/SIRBLn, SIRBEn/SIRBELn) are provided on chip
- Single transfer mode and repeat transfer mode can be specified

Remark n = 0, 1

10.4.2 Configuration

CSIn is controlled via the clocked serial interface mode register (CSIMn) (n = 0, 1). Transmission/reception of data is performed by reading/writing the SIOn register (n = 0, 1).

(1) Clocked serial interface mode registers 0, 1 (CSIM0, CSIM1)

The CSIMn register is an 8-bit register that specifies the operation of CSIn.

(2) Clocked serial interface clock selection registers 0, 1 (CSIC0, CSIC1)

The CSICn register is an 8-bit register that controls the CSIn serial transfer operation.

(3) Serial I/O shift registers 0, 1 (SIO0, SIO1)

The SIOn register is a 16-bit shift register that converts parallel data into serial data.
The SIOn register is used for both transmission and reception.
Data is shifted in (reception) and shifted out (transmission) from the MSB or LSB side.
The actual transmission/reception operations are started up by accessing the buffer register.

(4) Serial I/O shift registers L0, L1 (SIOL0, SIOL1)

The SIOLn register is an 8-bit shift register that converts parallel data into serial data.
The SIOLn register is used for both transmission and reception.
Data is shifted in (reception) and shifted out (transmission) from the MSB or LSB side.
The actual transmission/reception operations are started up by access of the buffer register .

(5) Clocked serial interface receive buffer registers 0, 1 (SIRB0, SIRB1)

The SIRBn register is a 16-bit buffer register that stores receive data.

(6) Clocked serial interface receive buffer registers L0, L1 (SIRBL0, SIRBL1)

The SIRBLn register is an 8-bit buffer register that stores receive data.

(7) Clocked serial interface read-only receive buffer registers 0, 1 (SIRBE0, SIRBE1)

The SIRBEn register is a 16-bit buffer register that stores receive data.
The SIRBEn register is the same as the SIRBn register. It is used to read the contents of the SIRBn register.

(8) Clocked serial interface read-only receive buffer registers L0, L1 (SIRBEL0, SIRBEL1)

The SIRBELn register is an 8-bit buffer register that stores receive data.
The SIRBELn register is the same as the SIRBLn register. It is used to read the contents of the SIRBLn register.

(9) Clocked serial interface transmit buffer registers 0, 1 (SOTB0, SOTB1)

The SOTBn register is a 16-bit buffer register that stores transmit data.

(10) Clocked serial interface transmit buffer registers L0, L1 (SOTBL0, SOTBL1)

The SOTBLn register is an 8-bit buffer register that stores transmit data.

(11) Clocked serial interface initial transmit buffer registers (SOTBF0, SOTBF1)

The SOTBFn register is a 16-bit buffer register that stores the initial transmit data in the repeat transfer mode.

(12) Clocked serial interface initial transmit buffer register L (SOTBFL0, SOTBFL1)

The SOTBFLn register is an 8-bit buffer register that stores initial transmit data in the repeat transfer mode.

(13) Selector

The selector selects the serial clock to be used.

(14) Serial clock controller

Controls the serial clock supply to the shift register. Also controls the clock output to the $\overline{\text{SCKn}}$ pin when the internal clock is used.

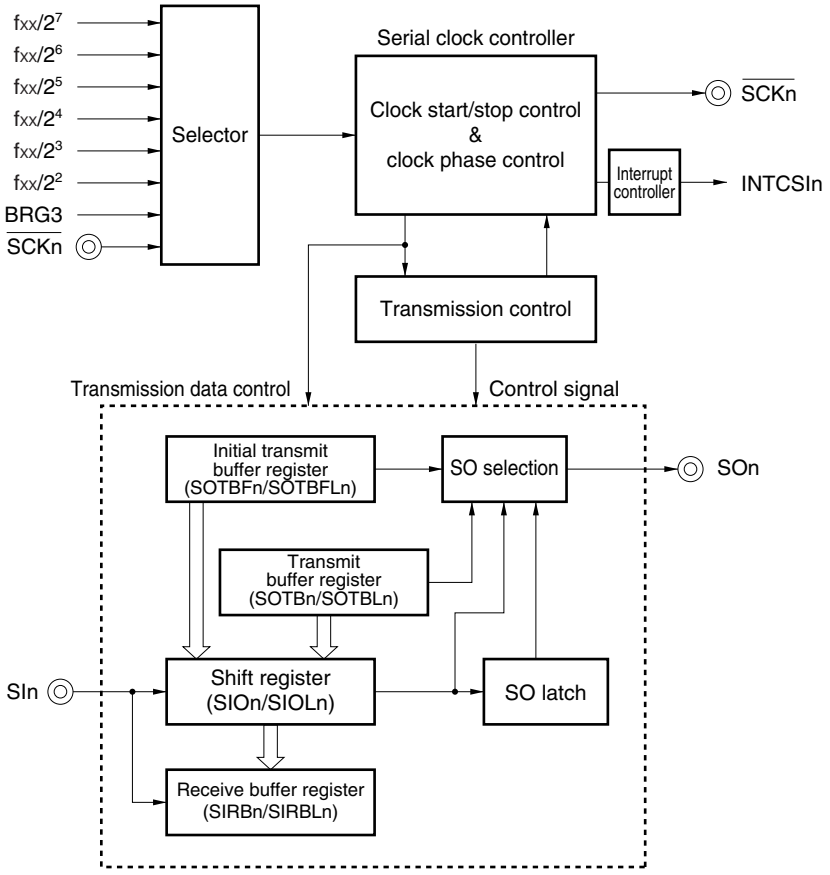
(15) Serial clock counter

Counts the serial clock output or input during transmission/reception operation, and checks whether 8-bit or 16-bit data transmission/reception has been performed.

(16) Interrupt controller

Controls the interrupt request timing.

Figure 10-26. Block Diagram of Clocked Serial Interface



Remarks 1. n = 0, 1

2. fxx: Internal system clock

3. The SO1, SI1, and $\overline{\text{SCK1}}$ pins function alternately as the TXD1, RXD1, and $\overline{\text{ASCK1}}$ pins.

10.4.3 Control registers

Because CSI1 shares its pins with UART1, the CSI1 mode must be preset by using the PMC3 and RFC3 registers (refer to **10.1.1 Selecting mode of UART1 or CSI1**).

(1) Clocked serial interface mode registers 0, 1 (CSIM0, CSIM1)

The CSIMn register controls the CSIn operation (n = 0, 1).

These registers can be read/written in 8-bit or 1-bit units (however, bit 0 is read-only).

Caution Overwriting the TRMDn, CCL, DIRn, CSIT, and AUTO bits of the CSIMn register can be done only when the CSOTn bit = 0. If these bits are overwritten at any other time, the operation cannot be guaranteed.

	<7>	<6>	5	<4>	3	2	1	<0>	Address	After reset
CSIM0	CSICAE0	TRMD0	CCL	DIR0	CSIT	AUTO	0	CSOT0	FFFFF900H	00H

	<7>	<6>	5	<4>	3	2	1	<0>	Address	After reset
CSIM1	CSICAE1	TRMD1	CCL	DIR1	CSIT	AUTO	0	CSOT1	FFFFF910H	00H

Bit position	Bit name	Function
7	CSICAEn	Enables/disables CSIn operation. 0: Enables CSIn operation. 1: Disables CSIn operation. The internal CSIn circuit can be reset asynchronously by setting the CSICAEn bit to 0. For the $\overline{SCKn}$ and SOn pin output status when the CSICAEn bit = 0, refer to **10.4.5 Output pins**.
6	TRMDn	Specifies transmission/reception mode. 0: Receive-only mode 1: Transmission/reception mode When the TRMDn bit = 0, receive-only transfer is performed and the SOn pin output is fixed to low level. Data reception is started by reading the SIRBn register. When the TRMDn bit = 1, transmission/reception is started by writing data to the SOTBn register.
5	CCL	Specifies data length. 0: 8 bits 1: 16 bits
4	DIRn	Specifies transfer direction mode (MSB/LSB). 0: First bit of transfer data is MSB 1: First bit of transfer data is LSB
3	CSIT	Controls delay of interrupt request signal. 0: No delay 1: Delay mode (interrupt request signal is delayed 1/2 cycle). The delay mode (CSIT bit = 1) is valid only in the master mode (CKS2 to CSK0 bits of the CSICn register are not 11B). In the slave mode (CKS2 to CKS0 bits are 11B), do not set the delay mode. **Caution The delay mode (CSIT bit = 1) is valid only in the master mode (CKS2 to CSK0 bits of the CSICn register are not 111B). In the slave mode (CKS2 to CKS0 bits are 111B), do not set the delay mode.**
2	AUTO	Specifies single transfer mode or repeat transfer mode. 0: Single transfer mode 1: Repeat transfer mode
0	CSOTn	Flag indicating transfer status. 0: Idle status 1: Transfer execution status **Caution The CSOTn bit is cleared (0) by writing 0 to the CSICAEn bit.**

Remark n = 0, 1

(2) Clocked serial interface clock selection registers 0, 1 (CSIC0, CSIC1)

The CSICn register is an 8-bit register that controls the CSIn transfer operation (n = 0, 1).

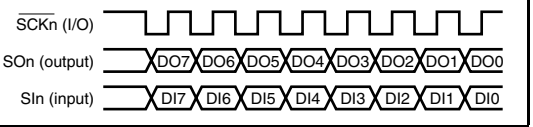
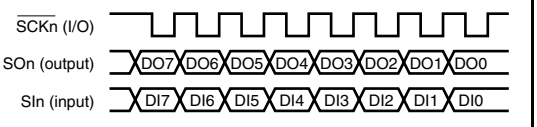
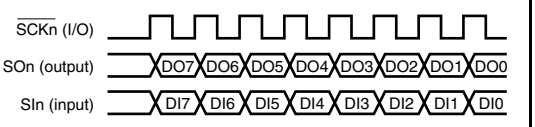
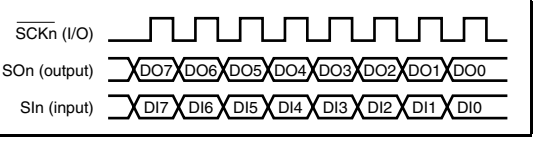
These registers can be read/written in 8-bit or 1-bit units.

Caution The CSICn register can be overwritten only when the CSICAEn bit of the CSIMn register = 0.

	7	6	5	4	3	2	1	0	Address	After reset
CSIC0	0	0	0	CKP	DAP	CKS2	CKS1	CKS0	FFFFF901H	00H

	7	6	5	4	3	2	1	0	Address	After reset
CSIC1	0	0	0	CKP	DAP	CKS2	CKS1	CKS0	FFFFF911H	00H

Bit position	Bit name	Function
4, 3	CKP, DAP	Specifies operation mode.

CKP	DAP	Operation mode
0	0	
0	1	
1	0	
1	1	

Remark n = 0, 1

Bit position	Bit name	Function
2 to 0	CKS2 to CKS0	Specifies serial clock.

CKS2	CKS1	CKS0	Serial clock	Mode
0	0	0	$f_{XX}/2^7$	Master mode
0	0	1	$f_{XX}/2^6$	Master mode
0	1	0	$f_{XX}/2^5$	Master mode
0	1	1	$f_{XX}/2^4$	Master mode
1	0	0	$f_{XX}/2^3$	Master mode
1	0	1	$f_{XX}/2^2$	Master mode
1	1	0	Clock generated by BRG3	Master mode
1	1	1	External clock ($\overline{SCKn}$)	Slave mode

Remark f_{XX}: Internal system clock frequency
n = 0, 1

(3) Clocked serial interface receive buffer registers 0, 1 (SIRB0, SIRB1)

The SIRBn register is a 16-bit buffer register that stores receive data (n = 0, 1).

When the receive-only mode is set (TRMDn bit of CSIMn register = 0), the reception operation is started by reading data from the SIRBn register.

These registers are read-only, in 16-bit units.

In addition to reset input, these registers can also be initialized by clearing (0) the CSICAEn bit of the CSIMn register.

Cautions 1. Read the SIRBn register only when the 16-bit data length has been set (CCL bit of CSIMn register = 1).

2. When the single transfer mode has been set (AUTO bit of CSIMn register = 0), perform a read operation only in the idle state (CSOTn bit of CSIMn register = 0). If the SIRBn register is read during data transfer, the data cannot be guaranteed.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
SIRB0	SIRB 15	SIRB 14	SIRB 13	SIRB 12	SIRB 11	SIRB 10	SIRB 9	SIRB 8	SIRB 7	SIRB 6	SIRB 5	SIRB 4	SIRB 3	SIRB 2	SIRB 1	SIRB 0	FFFFF902H	0000H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
SIRB1	SIRB 15	SIRB 14	SIRB 13	SIRB 12	SIRB 11	SIRB 10	SIRB 9	SIRB 8	SIRB 7	SIRB 6	SIRB 5	SIRB 4	SIRB 3	SIRB 2	SIRB 1	SIRB 0	FFFFF912H	0000H

Bit position	Bit name	Function
15 to 0	SIRB15 to SIRB0	Stores receive data.

(4) Clocked serial interface receive buffer registers L0, L1 (SIRBL0, SIRBL1)

The SIRBLn register is an 8-bit buffer register that stores receive data (n = 0, 1).

When the receive-only mode is set (TRMDn bit of CSIMn register = 0), the reception operation is started by reading data from the SIRBLn register.

These registers are read-only, in 8-bit or 1-bit units.

In addition to reset input, these registers can also be initialized by clearing (0) the CSICAEn bit of the CSIMn register.

The SIRBLn register is the same as the lower bytes of the SIRBn register.

Cautions 1. Read the SIRBLn register only when the 8-bit data length has been set (CCL bit of CSIMn register = 0).

2. When the single transfer mode is set (AUTO bit of CSIMn register = 0), perform a read operation only in the idle state (CSOTn bit of CSIMn register = 0). If the SIRBLn register is read during data transfer, the data cannot be guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
SIRBL0	SIRB7	SIRB6	SIRB5	SIRB4	SIRB3	SIRB2	SIRB1	SIRB0	FFFFF902H	00H

	7	6	5	4	3	2	1	0	Address	After reset
SIRBL1	SIRB7	SIRB6	SIRB5	SIRB4	SIRB3	SIRB2	SIRB1	SIRB0	FFFFF912H	00H

Bit position	Bit name	Function
7 to 0	SIRB7 to SIRB0	Stores receive data.

(5) Clocked serial interface read-only receive buffer registers 0, 1 (SIRBE0, SIRBE1)

The SIRBEn register is a 16-bit buffer register that stores receive data (n = 0, 1).

These registers are read-only, in 16-bit units.

In addition to reset input, this register can also be initialized by clearing (0) the CSICAEn bit of the CSIMn register.

The SIRBEn register is the same as the SIRBn register. It is used to read the contents of the SIRBn register.

Cautions 1. The receive operation is not started even if data is read from the SIRBEn register.

2. The SIRBEn register can be read only if the 16-bit data length is set (CCL bit of CSIMn register = 1).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
SIRBE0	SIRBE 15	SIRBE 14	SIRBE 13	SIRBE 12	SIRBE 11	SIRBE 10	SIRBE 9	SIRBE 8	SIRBE 7	SIRBE 6	SIRBE 5	SIRBE 4	SIRBE 3	SIRBE 2	SIRBE 1	SIRBE 0	FFFFF906H	0000H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
SIRBE1	SIRBE 15	SIRBE 14	SIRBE 13	SIRBE 12	SIRBE 11	SIRBE 10	SIRBE 9	SIRBE 8	SIRBE 7	SIRBE 6	SIRBE 5	SIRBE 4	SIRBE 3	SIRBE 2	SIRBE 1	SIRBE 0	FFFFF916H	0000H

Bit position	Bit name	Function
15 to 0	SIRBE15 to SIRBE0	Stores receive data.

(6) Clocked serial interface read-only receive buffer registers L0, L1 (SIRBEL0, SIRBEL1)

The SIRBELn register is an 8-bit buffer register that stores receive data (n = 0, 1).

These registers are read-only, in 8-bit or 1-bit units.

In addition to reset input, this register can also be initialized by clearing (0) the CSICAEn bit of the CSIMn register.

The SIRBELn register is the same as the SIRBLn register. It is used to read the contents of the SIRBLn register.

Cautions 1. The receive operation is not started even if data is read from the SIRBELn register.

2. The SIRBELn register can be read only if the 8-bit data length has been set (CCL bit of CSIMn register = 0).

	7	6	5	4	3	2	1	0	Address	After reset
SIRBEL0	SIRBE7	SIRBE6	SIRBE5	SIRBE4	SIRBE3	SIRBE2	SIRBE1	SIRBE0	FFFFF906H	00H

	7	6	5	4	3	2	1	0	Address	After reset
SIRBEL1	SIRBE7	SIRBE6	SIRBE5	SIRBE4	SIRBE3	SIRBE2	SIRBE1	SIRBE0	FFFFF916H	00H

Bit position	Bit name	Function
7 to 0	SIRBE7 to SIRBE0	Stores receive data.

(7) Clocked serial interface transmit buffer registers 0, 1 (SOTB0, SOTB1)

The SOTBn register is a 16-bit buffer register that stores transmit data (n = 0, 1).

When the transmission/reception mode is set (TRMDn bit of CSIMn register = 1), the transmission operation is started by writing data to the SOTBn register.

This register can be read/written in 16-bit units.

Cautions 1. Access the SOTBn register only when the 16-bit data length is set (CCL bit of CSIMn register = 1).

2. When the single transfer mode is set (AUTO bit of CSIMn register = 0), perform access only in the idle state (CSOTn bit of CSIMn register = 0). If the SOTBn register is accessed during data transfer, the data cannot be guaranteed.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
SOTB0	SOTB 15	SOTB 14	SOTB 13	SOTB 12	SOTB 11	SOTB 10	SOTB 9	SOTB 8	SOTB 7	SOTB 6	SOTB 5	SOTB 4	SOTB 3	SOTB 2	SOTB 1	SOTB 0	FFFFF904H	0000H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
SOTB1	SOTB 15	SOTB 14	SOTB 13	SOTB 12	SOTB 11	SOTB 10	SOTB 9	SOTB 8	SOTB 7	SOTB 6	SOTB 5	SOTB 4	SOTB 3	SOTB 2	SOTB 1	SOTB 0	FFFFF914H	0000H

Bit position	Bit name	Function
15 to 0	SOTB15 to SOTB0	Stores transmit data.

(8) Clocked serial interface transmit buffer registers L0, L1 (SOTBL0, SOTBL1)

The SOTBLn register is an 8-bit buffer register that stores transmit data (n = 0, 1).

When the transmission/reception mode is set (TRMDn bit of CSIMn register = 1), the transmission operation is started by writing data to the SOTBLn register.

These registers can be read/written in 8-bit or 1-bit units.

The SOTBLn register is the same as the lower bytes of the SOTBn register.

Cautions 1. Access the SOTBLn register only when the 8-bit data length has been set (CCL bit of CSIMn register = 0).

2. When the single transfer mode is set (AUTO bit of CSIMn register = 0), perform access only in the idle state (CSOTn bit of CSIMn register = 0). If the SOTBLn register is accessed during data transfer, the data cannot be guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
SOTBL0	SOTB7	SOTB6	SOTB5	SOTB4	SOTB3	SOTB2	SOTB1	SOTB0	FFFFF904H	00H

	7	6	5	4	3	2	1	0	Address	After reset
SOTBL1	SOTB7	SOTB6	SOTB5	SOTB4	SOTB3	SOTB2	SOTB1	SOTB0	FFFFF914H	00H

Bit position	Bit name	Function
7 to 0	SOTB7 to SOTB0	Stores transmit data.

(9) Clocked serial interface initial transmit buffer registers 0, 1 (SOTBF0, SOTBF1)

The SOTBFn register is a 16-bit buffer register that stores initial transmission data in the repeat transfer mode (n = 0, 1).

The transmission operation is not started even if data is written to the SOTBFn register.

These registers can be read/written in 16-bit units.

Caution Access the SOTBFn register only when the 16-bit data length has been set (CCL bit of CSIMn register = 1), and only in the idle state (CSOTn bit of CSIMn register = 0). If the SOTBFn register is accessed during data transfer, the data cannot be guaranteed.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
SOTBF0	SOTBF15	SOTBF14	SOTBF13	SOTBF12	SOTBF11	SOTBF10	SOTBF9	SOTBF8	SOTBF7	SOTBF6	SOTBF5	SOTBF4	SOTBF3	SOTBF2	SOTBF1	SOTBF0	FFFFF908H	0000H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
SOTBF1	SOTBF15	SOTBF14	SOTBF13	SOTBF12	SOTBF11	SOTBF10	SOTBF9	SOTBF8	SOTBF7	SOTBF6	SOTBF5	SOTBF4	SOTBF3	SOTBF2	SOTBF1	SOTBF0	FFFFF918H	0000H

Bit position	Bit name	Function
15 to 0	SOTBF15 to SOTBF0	Stores initial transmission data in repeat transfer mode.

(10) Clocked serial interface initial transmit buffer registers L0, L1 (SOTBFL0, SOTBFL1)

The SOTBFLn register is an 8-bit buffer register that stores initial transmission data in the repeat transfer mode (n = 0, 1).

The transmission operation is not started even if data is written to the SOTBFLn register.

These registers can be read/written in 8-bit or 1-bit units.

The SOTBFLn register is the same as the lower bytes of the SOTBFn register.

Caution Access the SOTBFLn register only when the 8-bit data length has been set (CCL bit of CSIMn register = 0), and only in the idle state (CSOTn bit of CSIMn register = 0). If the SOTBFLn register is accessed during data transfer, the data cannot be guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
SOTBFL0	SOTBF7	SOTBF6	SOTBF5	SOTBF4	SOTBF3	SOTBF2	SOTBF1	SOTBF0	FFFFF908H	00H

	7	6	5	4	3	2	1	0	Address	After reset
SOTBFL1	SOTBF7	SOTBF6	SOTBF5	SOTBF4	SOTBF3	SOTBF2	SOTBF1	SOTBF0	FFFFF918H	00H

Bit position	Bit name	Function
7 to 0	SOTBF7 to SOTBF0	Stores initial transmission data in repeat transfer mode.

(11) Serial I/O shift registers 0, 1 (SIO0, SIO1)

The SIOn register is a 16-bit shift register that converts parallel data into serial data (n = 0, 1).

The transfer operation is not started even if the SIOn register is read.

These registers are read-only, in 16-bit units.

In addition to reset input, this register can also be initialized by clearing (0) the CSICAEn bit of the CSIMn register.

Caution Access the SIOn register only when the 16-bit data length has been set (CCL bit of CSIMn register = 1), and only in the idle state (CSOTn bit of CSIMn register = 0). If the SIOn register is accessed during data transfer, the data cannot be guaranteed.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
SIO0	SIO15	SIO14	SIO13	SIO12	SIO11	SIO10	SIO9	SIO8	SIO7	SIO6	SIO5	SIO4	SIO3	SIO2	SIO1	SIO0	FFFFF90AH	0000H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
SIO1	SIO15	SIO14	SIO13	SIO12	SIO11	SIO10	SIO9	SIO8	SIO7	SIO6	SIO5	SIO4	SIO3	SIO2	SIO1	SIO0	FFFFF91AH	0000H

Bit position	Bit name	Function
15 to 0	SIO15 to SIO0	Data is shifted in (reception) or shifted out (transmission) from the MSB or LSB side.

(12) Serial I/O shift registers L0, L1 (SIOL0, SIOL1)

The SIOLn register is an 8-bit shift register that converts parallel data into serial data (n = 0, 1).

The transfer operation is not started even if the SIOLn register is read.

These registers are read-only, in 8-bit or 1-bit units.

In addition to reset input, this register can also be initialized by clearing (0) the CSICAEn bit of the CSIMn register.

The SIOLn register is the same as the lower bytes of the SIOn register.

Caution Access the SIOLn register only when the 8-bit data length has been set (CCL bit of CSIMn register = 0), and only in the idle state (CSOTn bit of CSIMn register = 0). If the SIOLn register is accessed during data transfer, the data cannot be guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
SIOL0	SIO7	SIO6	SIO5	SIO4	SIO3	SIO2	SIO1	SIO0	FFFFF90AH	00H

	7	6	5	4	3	2	1	0	Address	After reset
SIOL1	SIO7	SIO6	SIO5	SIO4	SIO3	SIO2	SIO1	SIO0	FFFFF91AH	00H

Bit position	Bit name	Function
7 to 0	SIO7 to SIO0	Data is shifted in (reception) or shifted out (transmission) from the MSB or LSB side.

10.4.4 Operation

(1) Single transfer mode

(a) Usage

In the receive-only mode (TRMDn bit of CSIMn register = 0), transfer is started by reading[Note 1] the receive data buffer register (SIRBn/SIRBLn) (n = 0, 1).

In the transmission/reception mode (TRMDn bit of CSIMn register = 1), transfer is started by writing[Note 2] to the transmit data buffer register (SOTBn/SOTBLn).

In the slave mode, the operation must be enabled beforehand (CSICAEn bit of CSIMn register = 1).

When transfer is started, the value of the CSOTn bit of the CSIMn register becomes 1 (transmission execution status).

Upon transfer completion, the transmission/reception completion interrupt (INTCSIn) is set (1), and the CSOTn bit is cleared (0). The next data transfer request is then waited for.

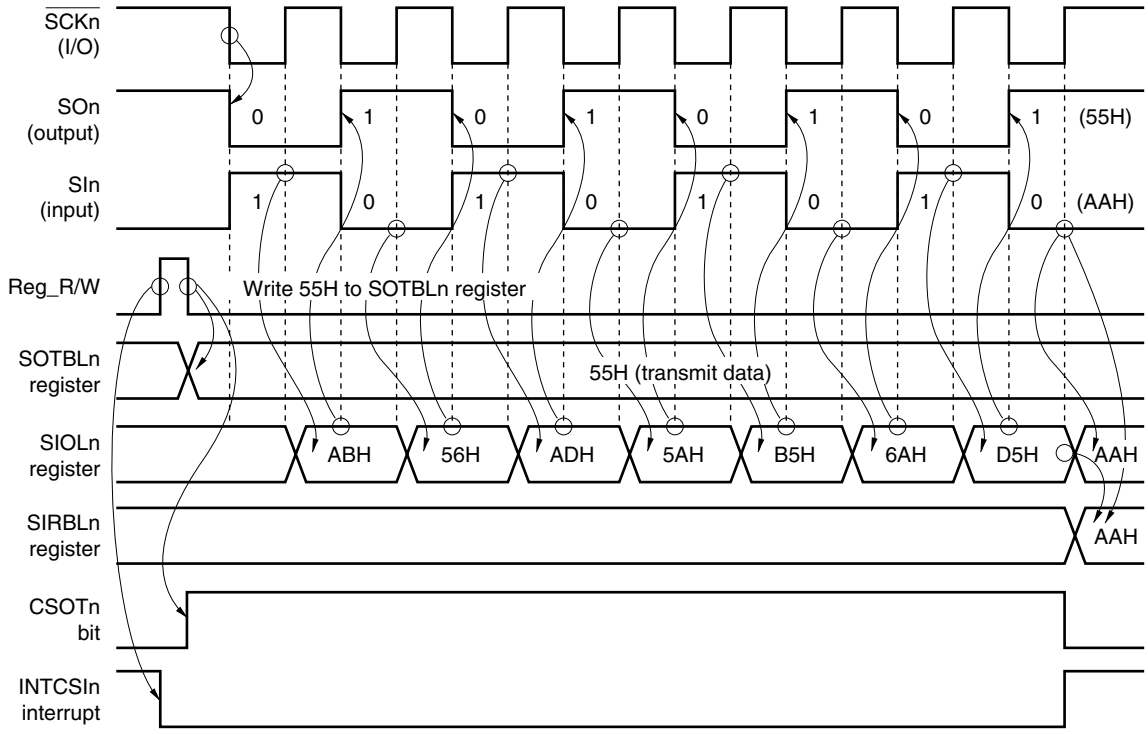
Notes 1. When the 16-bit data length (CCL bit of CSIMn register = 1) has been set, read the SIRBn register. When the 8-bit data length (CCL bit of CSIMn register = 0) has been set, read the SIRBLn register.

2. When the 16-bit data length (CCL bit of CSIMn register = 1) has been set, write to the SOTBn register. When the 8-bit data length (CCL bit of CSIMn register = 0) has been set, write to the SOTBLn register.

Caution When the CSOTn bit of the CSIMn register = 1, do not manipulate the CSIn register.

Figure 10-27. Timing Chart in Single Transfer Mode (1/2)

(a) In transmission/reception mode, data length: 8 bits, transfer direction: MSB first, no interrupt delay, single transfer mode, operation mode: CKP bit = 0, DAP bit = 0

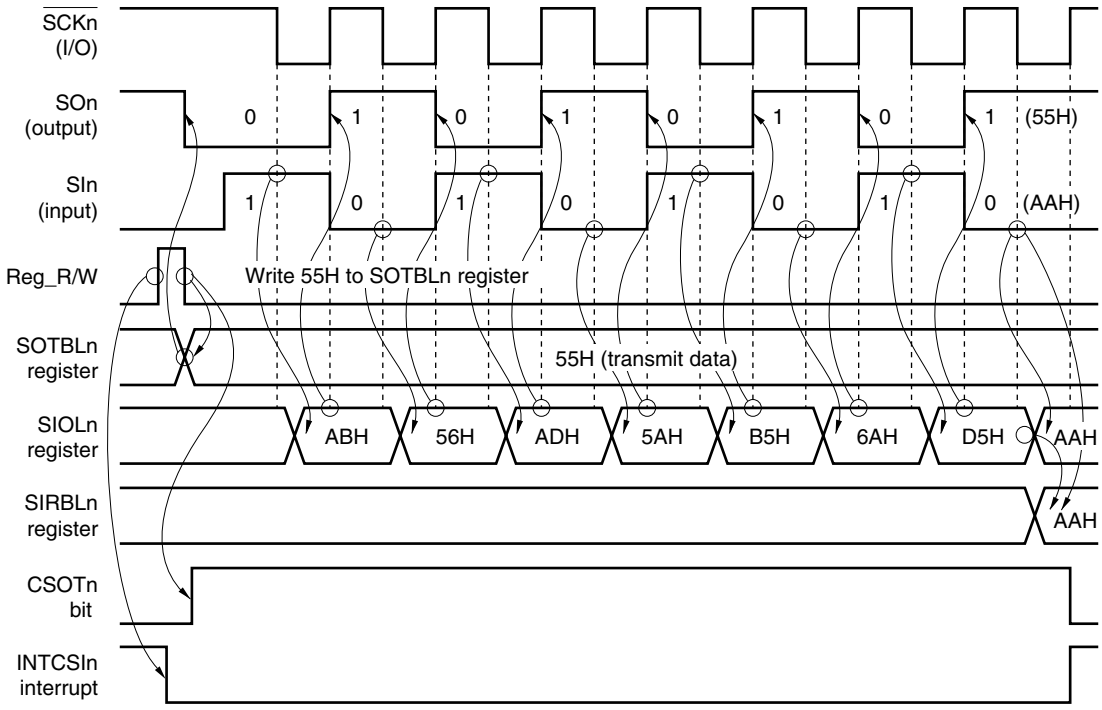


Remarks 1. n = 0, 1

2. Reg_R/W: Internal signal. This signal indicates that receive data buffer register (SIRBn/SIRBLn) read or transmit data buffer register (SOTBn/SOTBLn) write was performed.

Figure 10-27. Timing Chart in Single Transfer Mode (2/2)

(b) In transmission/reception mode, data length: 8 bits, transfer direction: MSB first, no interrupt delay, single transfer mode, operation mode: CKP bit = 0, DAP bit = 1



Remarks 1. n = 0, 1

2. Reg_R/W: Internal signal. This signal indicates that receive data buffer register (SIRBn/SIRBLn) read or transmit data buffer register (SOTBn/SOTBLn) write was performed.

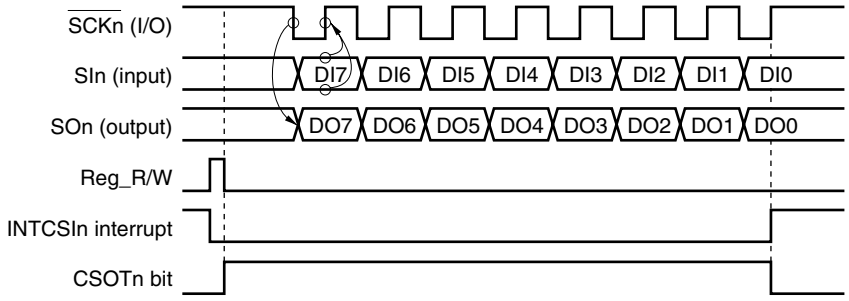
(b) Clock phase selection

The following shows the timing when changing the conditions for clock phase selection (CKP bit of CSICn register) and data phase selection (DAP bit of CSICn register) under the following conditions.

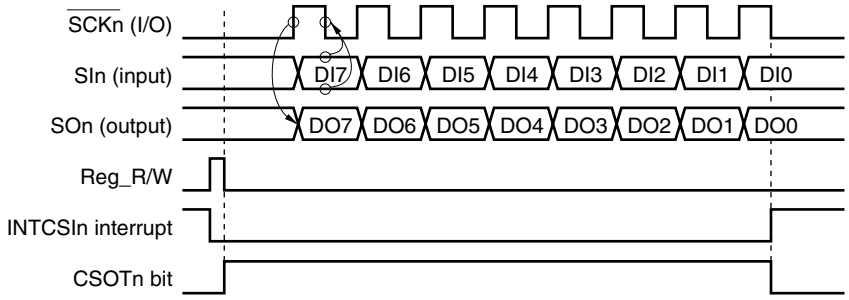
- Data length = 8 bits (CCL bit of CSIMn register = 0)
- First bit of transfer data = MSB (DIRn bit of CSIMn register = 0)
- No interrupt request signal delay control (CSIT bit of CSIMn register = 0)

Figure 10-28. Timing Chart According to Clock Phase Selection (1/2)

(a) When CKP bit = 0, DAP bit = 0



(b) When CKP bit = 1, DAP bit = 0

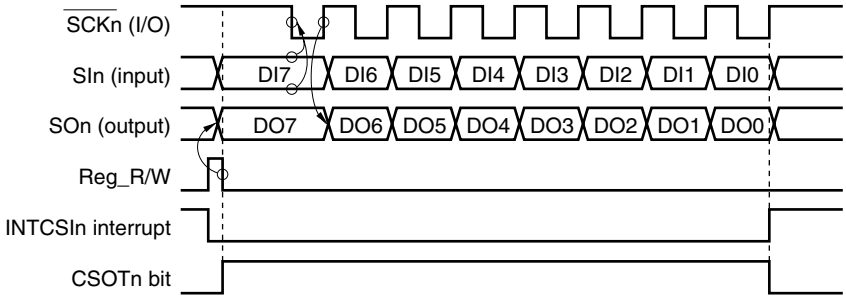


Remarks 1. n = 0, 1

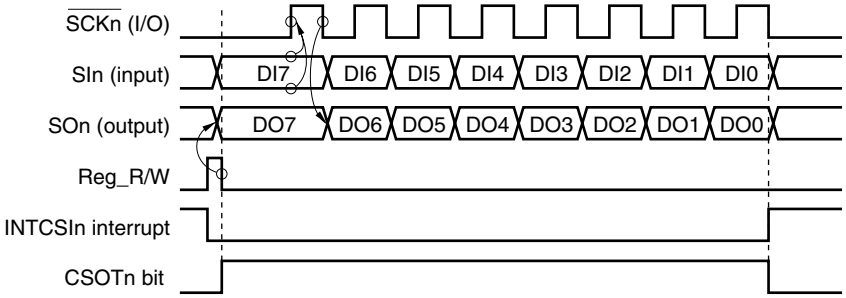
2. Reg_R/W: Internal signal. This signal indicates that receive data buffer register (SIRBn/SIRBLn) read or transmit data buffer register (SOTBn/SOTBLn) write was performed.

Figure 10-28. Timing Chart According to Clock Phase Selection (2/2)

(c) When CKP bit = 0, DAP bit = 1



(d) When CKP bit = 1, DAP bit = 1



Remarks 1. n = 0, 1

2. Reg_R/W: Internal signal. This signal indicates that receive data buffer register (SIRBn/SIRBLn) read or transmit data buffer register (SOTBn/SOTBLn) write was performed.

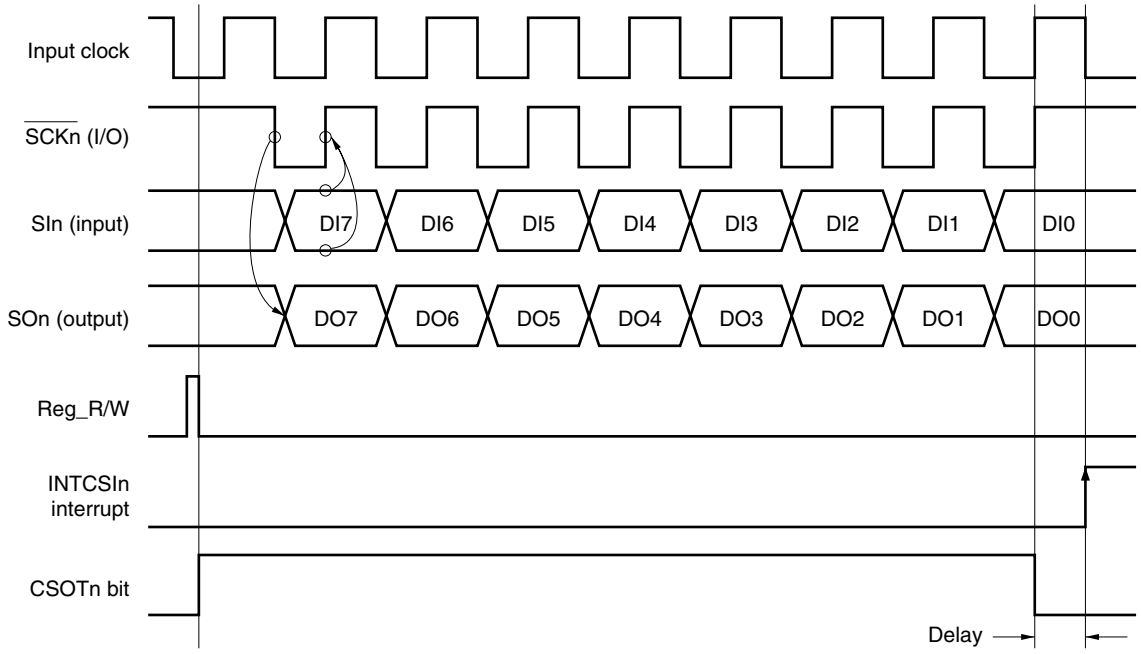
(c) Transmission/reception completion interrupt request signals (INTCSI0, INTCSI1)

INTCSIn is set (1) upon completion of data transmission/reception.

Caution The delay mode (CSIT bit = 1) is valid only in the master mode (bits CKS2 to CKS0 of the CSICn register are not 111B). The delay mode cannot be set when the slave mode is set (bits CKS2 to CKS0 = 111B).

Figure 10-29. Timing Chart of Interrupt Request Signal Output in Delay Mode (1/2)

(a) When CKP bit = 0, DAP bit = 0

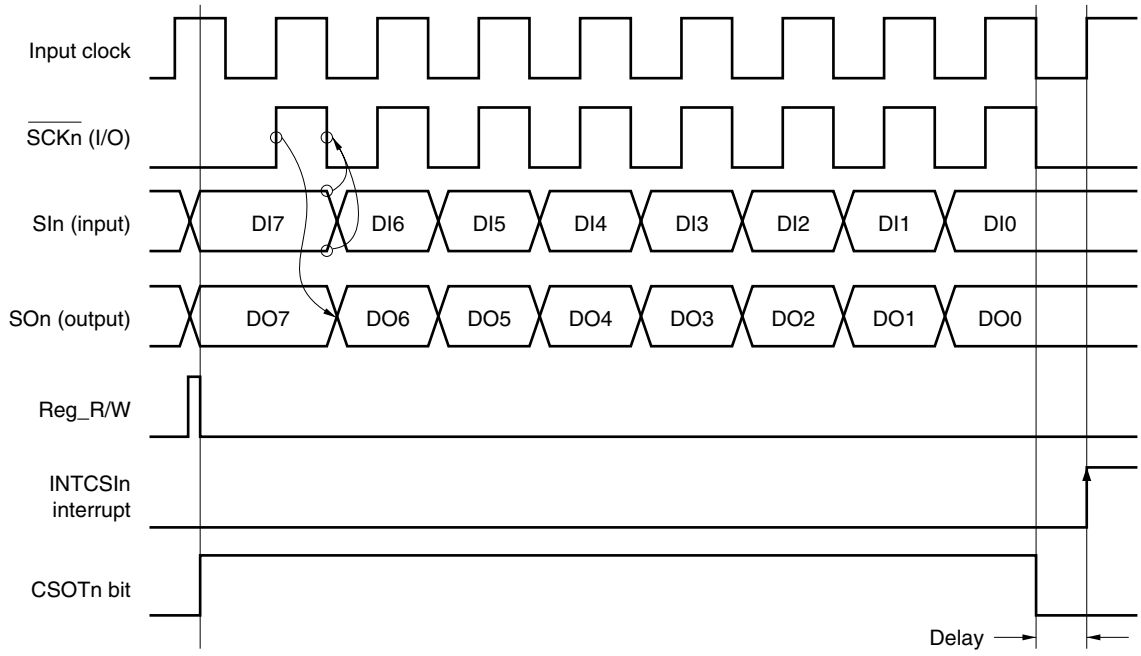


Remarks 1. n = 0, 1

2. Reg_R/W: Internal signal. This signal indicates that receive data buffer register (SIRBn/SIRBLn) read or transmit data buffer register (SOTBn/SOTBLn) write was performed.

Figure 10-29. Timing Chart of Interrupt Request Signal Output in Delay Mode (2/2)

(b) When CKP bit = 1, DAP bit = 1



Remarks 1. n = 0, 1

2. Reg_R/W: Internal signal. This signal indicates that receive data buffer register (SIRBn/SIRBLn) read or transmit data buffer register (SOTBn/SOTBLn) write was performed.

(2) Repeat transfer mode

(a) Usage (receive-only)

<1> Set the repeat transfer mode (AUTO bit of CSIMn register = 1) and the receive-only mode (TRMDn bit of CSIMn register = 0).

<2> Read the SIRBn register (start transfer with dummy read).

<3> Wait for the transmission/reception completion interrupt request (INTCSIn).

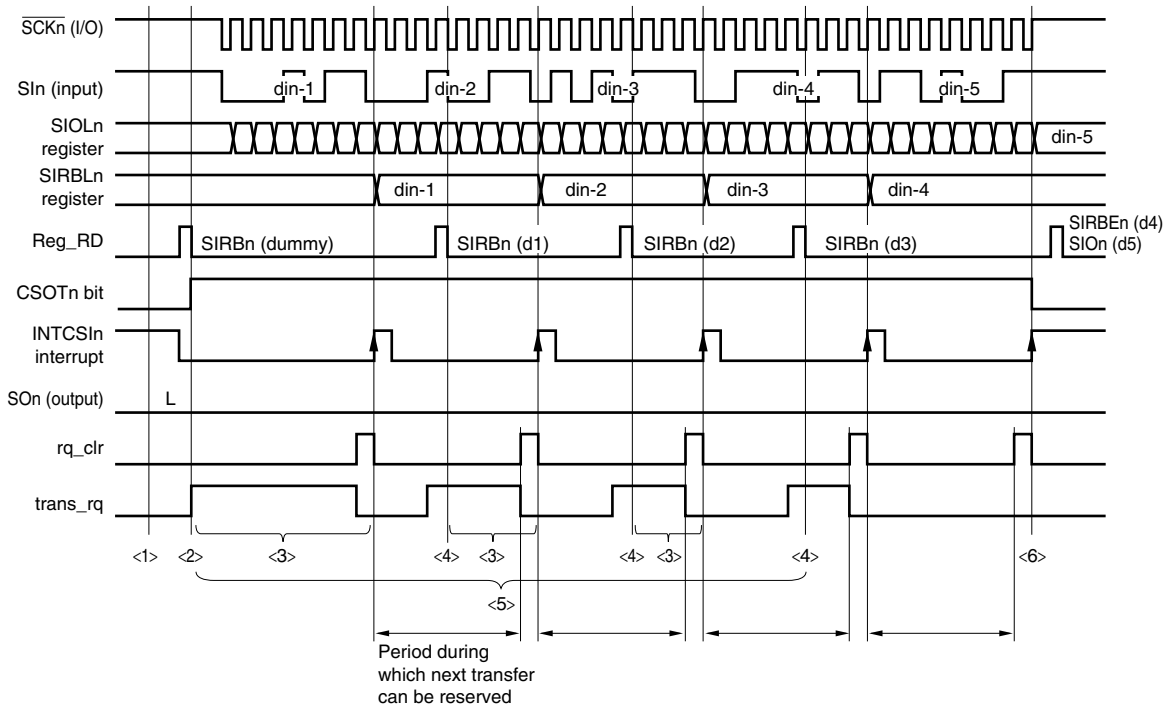
<4> When the transmission/reception completion interrupt request (INTCSIn) has been set (1), read the SIRBn register[Note] (reserve next transfer).

<5> Repeat steps <3> and <4> (N – 2) times. (N: Number of transfer data)

<6> Following output of the last transmission/reception completion interrupt request (INTCSIn), read the SIRBEn register and the SIOn register[Note].

Note When transferring N number of data, receive data is loaded by reading the SIRBn register from the first data to the (N – 2)th data. The (N – 1)th data is loaded by reading the SIRBEn register, and the Nth (last) data is loaded by reading the SIOn register.

Figure 10-30. Repeat Transfer (Receive-Only) Timing Chart



Remarks 1. n = 0, 1

2. Reg_RD: Internal signal. This signal indicates that the receive data buffer register (SIRBn/SIRBLn) has been read.

rq_clr: Internal signal. Transfer request clear signal.

trans_rq: Internal signal. Transfer request signal.

In the case of the repeat transfer mode, two transfer requests are set at the start of the first transfer. Following the transmission/reception completion interrupt request (INTCSIn), transfer is continued if the SIRBn register can be read within the next transfer reservation period. If the SIRBn register cannot be read, transfer ends and the SIRBn register does not receive the new value of the SIOn register.

The last data can be obtained by reading the SIOn register following completion of the transfer.

(b) Usage (transmission/reception)

<1> Set the repeat transfer mode (AUTO bit of CSIMn register = 1) and the transmission/reception mode (TRMDn bit of CSIMn register = 1)

<2> Write the first data to the SOTBFn register.

<3> Write the 2nd data to the SOTBn register (start transfer).

<4> Wait for the transmission/reception completion interrupt request (INTCSIn).

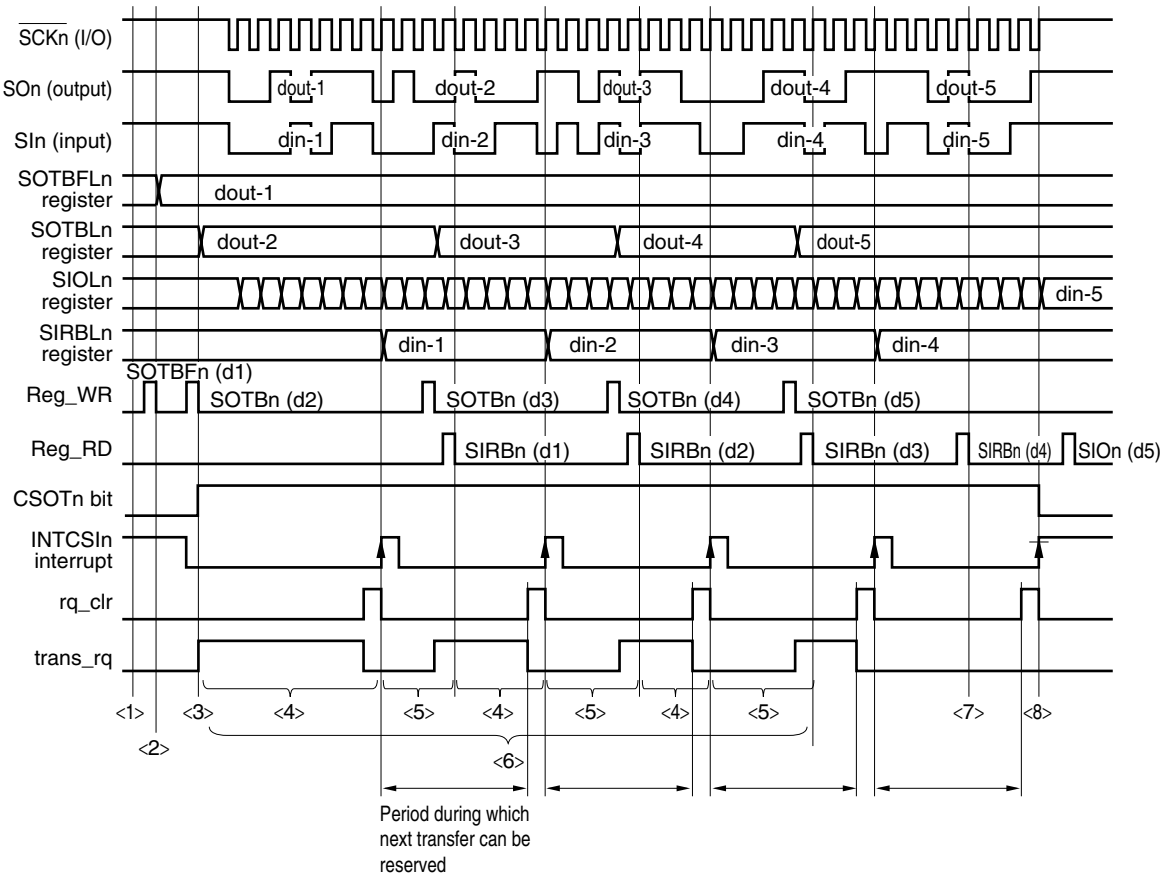
<5> When the transmission/reception completion interrupt request (INTCSIn) has been set (1), write the next data to the SOTBn register (reserve next transfer), and read the SIRBn register to load the receive data.

<6> Repeat steps <4> and <5> as long as data to be sent remains.

<7> Wait for the INTCSIn interrupt. When the interrupt request signal is set (1), read the SIRBn register to load the (N – 1)th receive data (N: Number of transfer data).

<8> Following the last transmission/reception completion interrupt request (INTCSIn), read the SIOn register to load the Nth (last) receive data.

Figure 10-31. Repeat Transfer (Transmission/Reception) Timing Chart



Remarks 1. n = 0, 1

2. Reg_WR: Internal signal. This signal indicates that the transmit data buffer register (SOTBn/SOTBLn) has been written.

Reg_RD: Internal signal. This signal indicates that the receive data buffer register (SIRBn/SIRBLn) has been read.

rq_clr: Internal signal. Transfer request clear signal.

trans_rq: Internal signal. Transfer request signal.

In the case of the repeat transfer mode, two transfer requests are set at the start of the first transfer. Following the transmission/reception completion interrupt request (INTCSIn), transfer is continued if the SOTBn register can be written within the next transfer reservation period. If the SOTBn register cannot be written, transfer ends and the SIRBn register does not receive the new value of the SIOn register.

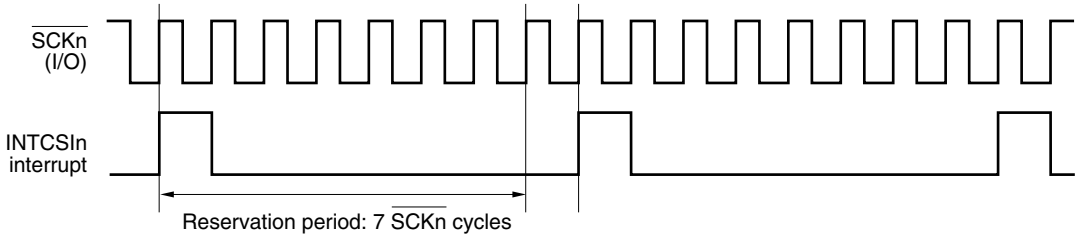
The last receive data can be obtained by reading the SIOn register following completion of the transfer.

(c) Next transfer reservation period

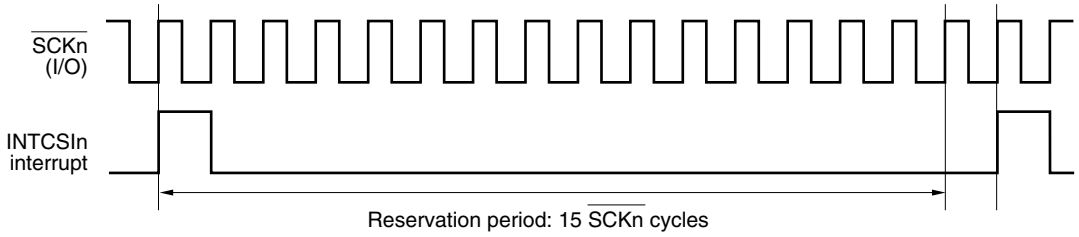
In the repeat transfer mode, the next transfer must be prepared with the period shown in Figure 10-32.

Figure 10-32. Timing Chart of Next Transfer Reservation Period (1/2)

(a) When data length: 8 bits, operation mode: CKP bit = 0, DAP bit = 0



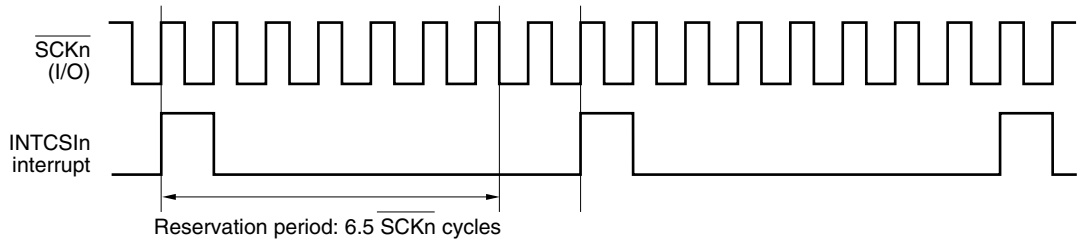
(b) When data length: 16 bits, operation mode: CKP bit = 0, DAP bit = 0



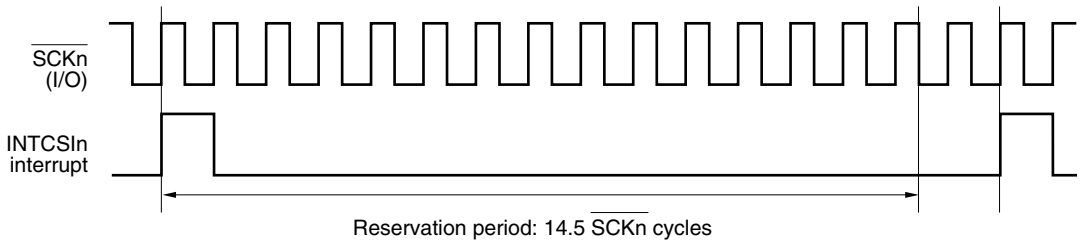
Remark n = 0, 1

Figure 10-32. Timing Chart of Next Transfer Reservation Period (2/2)

(c) When data length: 8 bits, operation mode: CKP bit = 0, DAP bit = 1



(d) When data length: 16 bits, operation mode: CKP bit = 0, DAP bit = 1



Remark n = 0, 1

(d) Cautions

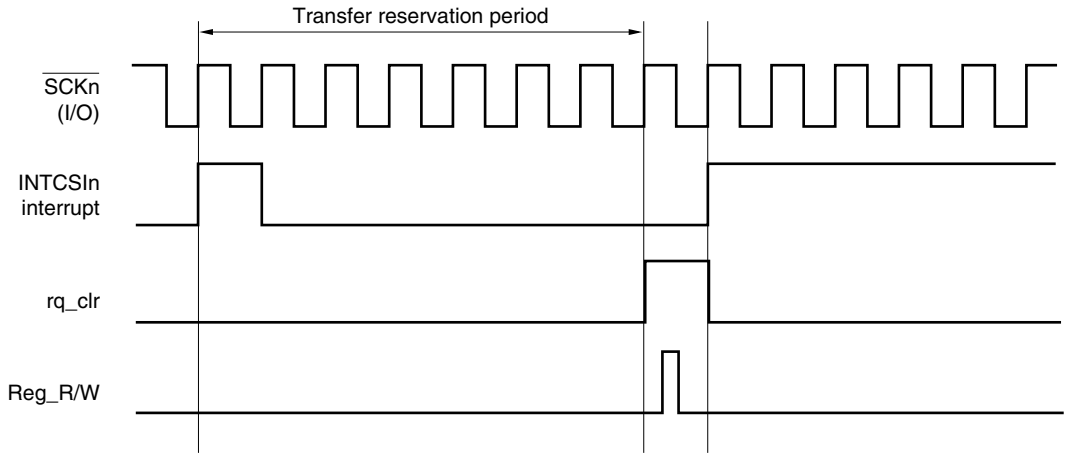
To continue repeat transfers, it is necessary to either read the SIRBn register or write to the SOTBn register during the transfer reservation period.

If access is performed to the SIRBn register or the SOTBn register when the transfer reservation period is over, the following occurs.

(i) In case of conflict between transfer request clear and register access

Since request cancellation has higher priority, the next transfer request is ignored. Therefore, transfer is interrupted, and normal data transfer cannot be performed.

Figure 10-33. Transfer Request Clear and Register Access Conflict



Remarks 1. n = 0, 1

2. rq_clr: Internal signal. Transfer request clear signal.

Reg_R/W: Internal signal. This signal indicates that the receive data buffer register (SIRBn/SIRBLn) read or transmit data buffer register (SOTBn/SOTBLn) write was performed.

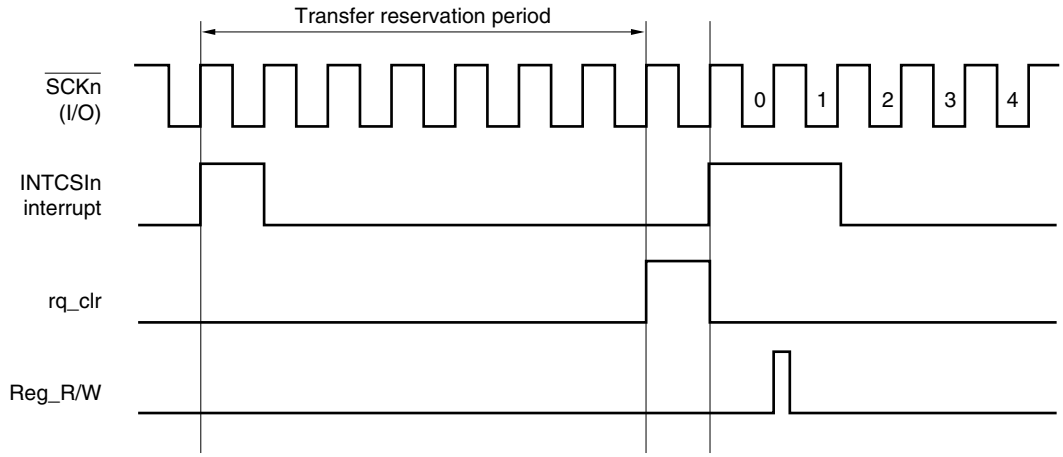
(ii) In case of conflict between interrupt request and register access

Since continuous transfer has stopped once, executed as a new repeat transfer.

In the slave mode, a bit phase error transfer error results (refer to **Figure 10-34**).

In the transmission/reception mode, the value of the SOTBFn register is retransmitted, and illegal data is sent.

Figure 10-34. Interrupt Request and Register Access Conflict



Remarks 1. n = 0, 1

2. rq_clr: Internal signal. Transfer request clear signal.

Reg_R/W: Internal signal. This signal indicates that receive data buffer register (SIRBn/SIRBLn) read or transmit data buffer register (SOTBn/SOTBLn) write was performed.

10.4.5 Output pins

(1) $\overline{\text{SCKn}}$ pin

When the CSIn operation is disabled (CSICAEn bit of CSIMn register = 0), the $\overline{\text{SCKn}}$ pin output status is as follows (n = 0, 1).

Table 10-9. $\overline{\text{SCKn}}$ Pin Output Status

CKP	CKS2	CKS1	CKS0	$\overline{\text{SCKn}}$ Pin Output
0	Don't care	Don't care	Don't care	Fixed to high level
1	1	1	1	Fixed to high level
	Other than above			Fixed to low level

Remarks 1. n = 0, 1

2. When any of the CKP and CKS2 to CKS0 bits of the CSICn register is overwritten, the $\overline{\text{SCKn}}$ pin output changes.

(2) SOn pin

When the CSIn operation is disabled (CSICAEn bit of CSIMn register = 0), the SOn pin output status is as follows (n = 0, 1).

Table 10-10. SOn Pin Output Status

TRMDn	DAP	AUTO	CCL	DIRn	SOn Pin Output
0	Don't care	Don't care	Don't care	Don't care	Fixed to low level
1	0	Don't care	Don't care	Don't care	SO latch value (low level)
	1	0	0	0	SOTB7 value
				1	SOTB0 value
			1	0	SOTB15 value
				1	SOTB0 value
		1	0	0	SOTBF7 value
				1	SOTBF0 value
			1	0	SOTBF15 value
				1	SOTBF0 value

Remarks 1. n = 0, 1

2. When any of the TRMDn, CCL, DIRn, and AUTO bits of the CSIMn register or DAP bit of the CSICn register is overwritten, the SOn pin output changes.

3. SOTBm: Bit m of SOTBn register (m = 0, 7, 15)

4. SOTBFm: Bit m of SOTBFn register (m = 0, 7, 15)

10.4.6 Dedicated baud rate generator 3 (BRG3)

(1) Configuration of baud rate generator 3 (BRG3)

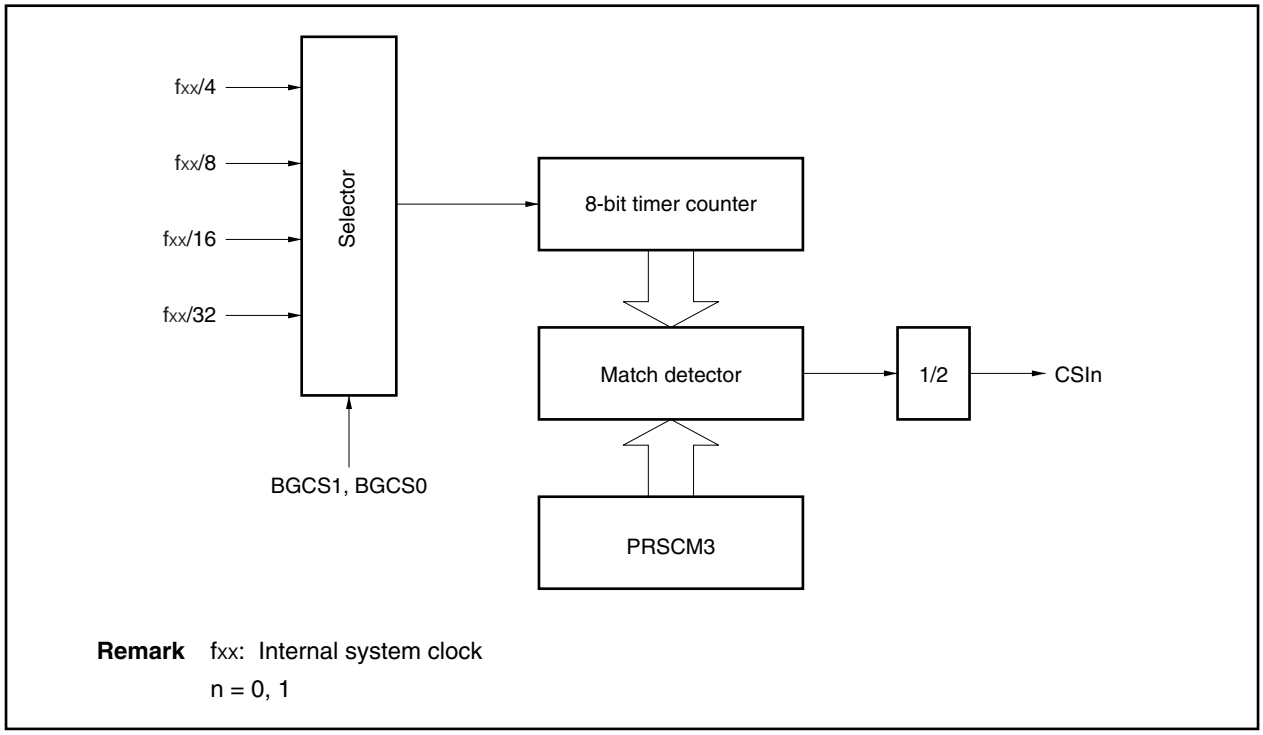
Dedicated baud rate generator output or the internal system clock (fxx) can be selected for the CSI0 and CSI1 serial clocks.

The serial clock source is specified by registers CSIC0 and CSIC1.

If dedicated baud rate generator output is specified, BRG3 is selected as the clock source.

Since the same serial clock can be shared for transmission and reception, baud rate is the same for both transmission and reception.

Figure 10-35. Block Diagram of Baud Rate Generator 3 (BRG3)



Remark fxx: Internal system clock
n = 0, 1

(2) Dedicated baud rate generator 3 (BRG3)

BRG3 is configured by an 8-bit timer counter that generates the baud rate signal, prescaler mode register 3 (PRSM3), which controls baud rate signal generation, prescaler compare register 3 (PRSCM3), which sets the value of the 8-bit timer counter, and a prescaler.

(a) Input clock

The internal system clock (fxx) is input to BRG3.

(b) Prescaler mode register 3 (PRSM3)

The PRSM3 register controls generation of the CSI0 and CSI1 baud rate signals.

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

Cautions 1. Do not change the value of the BGCS1, BGCS0 bits during a transmission/ reception operation.

2. Set the PRSM3 register prior to setting the CSICAEn bit of the CSIMn register to 1 (n = 0, 1).

	7	6	5	4	3	2	1	0	Address	After reset
PRSM3	0	0	0	CE	0	0	BGCS1	BGCS0	FFFFF920H	00H

Bit position	Bit name	Function
4	CE	Enables baud rate counter operation. 0: Stops baud rate counter operation and fixes baud rate output signal to 0. 1: Enables baud rate counter operation and starts baud rate output operation.
1, 0	BGCS1, BGCS0	Selects count clock for baud rate counter.

BGCS1	BGCS0	Count clock selection
0	0	fxx/4
0	1	fxx/8
1	0	fxx/16
1	1	fxx/32

Remark fxx: Internal system clock

(c) Prescaler compare register 3 (PRSCM3)

PRSCM3 is an 8-bit compare register that sets the value of the 8-bit timer counter.

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

Cautions 1. The internal timer counter is cleared by writing to the PRSM3 register. Therefore, do not write to the PRSCM3 register during transmission.

2. Set the PRSCM3 register prior to setting the CSICAEn bit of the CSIMn register to 1 (n = 0, 1). If the contents of the PRSCM3 register are overwritten when the value of the CSICAEn bit is 1, the cycle of the baud rate signal is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
PRSCM3	PRSCM7	PRSCM6	PRSCM5	PRSCM4	PRSCM3	PRSCM2	PRSCM1	PRSCM0	FFFFF922H	00H

(d) Baud rate signal cycle

The baud rate signal cycle is calculated as follows.

- **When setting value of PRSCM3 register is 00H**
  (Cycle of signal selected by bits BGCS1, BGCS0 of PRSM3 register) × 256 × 2
- **In cases other than above**
  (Cycle of signal selected by bits BGCS1, BGCS2 of PRSM3 register) × (setting value of PRSCM3 register) × 2

(e) Baud rate setting value

Table 10-11. Baud Rate Generator Setting Data

(a) When f_{XX} = 32 MHz

BGCS1	BGCS0	PRSCM Register Value	Clock (Hz)
0	0	1	4,000,000
0	0	2	2,000,000
0	0	4	1,000,000
0	0	8	500,000
0	0	16	250,000
0	0	40	100,000
0	0	80	50,000
0	0	160	25,000
0	1	200	10,000
1	0	200	5,000

(b) When f_{XX} = 40 MHz

BGCS1	BGCS0	PRSCM Register Value	Clock (Hz)
0	0	2	2,500,000
0	0	5	1,000,000
0	0	10	500,000
0	0	20	250,000
0	0	50	100,000
0	0	100	50,000
0	0	200	25,000
0	1	250	10,000
1	0	250	5,000

Caution Set the transfer clock so that it does not fall below the minimum value of 200 ns of the $\overline{SCKn}$ cycle (t_{CYSK1}) prescribed in the electrical specifications.

CHAPTER 11 A/D CONVERTER

11.1 Features

- Two 10-bit resolution on-chip A/D converters (A/D converter 0 and 1)
  Simultaneous sampling by two circuits is possible.
- Analog input: Total of 14 channels for two circuits
  A/D converter 0: 6 channels
  A/D converter 1: 8 channels
- On-chip A/D conversion result registers 0m, 1n (ADCR0m, ADCR1n)
  10 bits × 6 registers + 10 bits × 8 registers
- A/D conversion trigger mode
  A/D trigger mode
  A/D trigger polling mode
  Timer trigger mode
  External trigger mode
- Successive approximation technique
- Voltage detection mode

Remark m = 0 to 5, n = 0 to 7

11.2 Configuration

A/D converters 0 and 1, which employ a successive approximation technique, perform A/D conversion operations using A/D scan mode registers 00, 01, 10, and 11 (ADSCM00, ADSCM01, ADSCM10, and ADSCM11) and registers ADCR0m and ADCR1n (m = 0 to 5, n = 0 to 7).

(1) Input circuit

The input circuit selects an analog input (ANI0m or ANI1n) according to the mode set in the ADSCM00 or ADSCM10 register and sends it to the sample and hold circuit (m = 0 to 5, n = 0 to 7).

(2) Sample and hold circuit

The sample and hold circuit individually samples analog inputs sent sequentially from the input circuit and sends them to the comparator. It holds sampled analog inputs during A/D conversion.

(3) Voltage comparator

The voltage comparator compares the analog input voltage that was input with the output voltage of the D/A converter.

(4) D/A converter

The D/A converter is used to generate the voltage that matches the analog input.

The output voltage of the D/A converter is controlled by the successive approximation register (SAR).

(5) Successive approximation register (SAR)

The SAR is a 10-bit register that controls the output value of the D/A converter for comparing with the analog input voltage value. When an A/D conversion ends, the current contents of the SAR (conversion result) are stored in an A/D conversion result register (ADCR0m, ADCR1n) (m = 0 to 5, n = 0 to 7). When all specified A/D conversions end, an A/D conversion end interrupt (INTAD0, INTAD1) is also generated.

(6) A/D conversion result registers 0m, 1n (ADCR0m, ADCR1n)

ADCR0m and ADCR1n are 10-bit registers that hold A/D conversion results (m = 0 to 5, n = 0 to 7). Whenever an A/D conversion ends, the conversion result from the successive approximation register (SAR) is loaded.

$\overline{\text{RESET}}$ input sets these registers to 0000H.

(7) Controller

The controller selects an analog input, generates sample and hold circuit operation timing, controls conversion triggers, and specifies the conversion operation time according to the mode set by the ADSCMn0 or ADSCMn1 register.

(8) ANI0m, ANI1n pins (m = 0 to 5, n = 0 to 7)

The ANI0n and ANI1n pins are the analog input pins of each channel (total of 14 channels for two circuits) for analog converters 0 and 1. They input analog signals to be A/D converted.

Caution Make sure that the voltages input to ANI0m and ANI1n are within the range of the ratings. In particular, if a voltage (including noise) higher than AV_{DD0} and AV_{DD1} or lower than AV_{SS0} and AV_{SS1} (even if within the range of absolute maximum ratings) is input, the conversion value of that channel is invalid, and the conversion values of other channels may also be affected.

(9) AV_{SS0}, AV_{SS1} pins

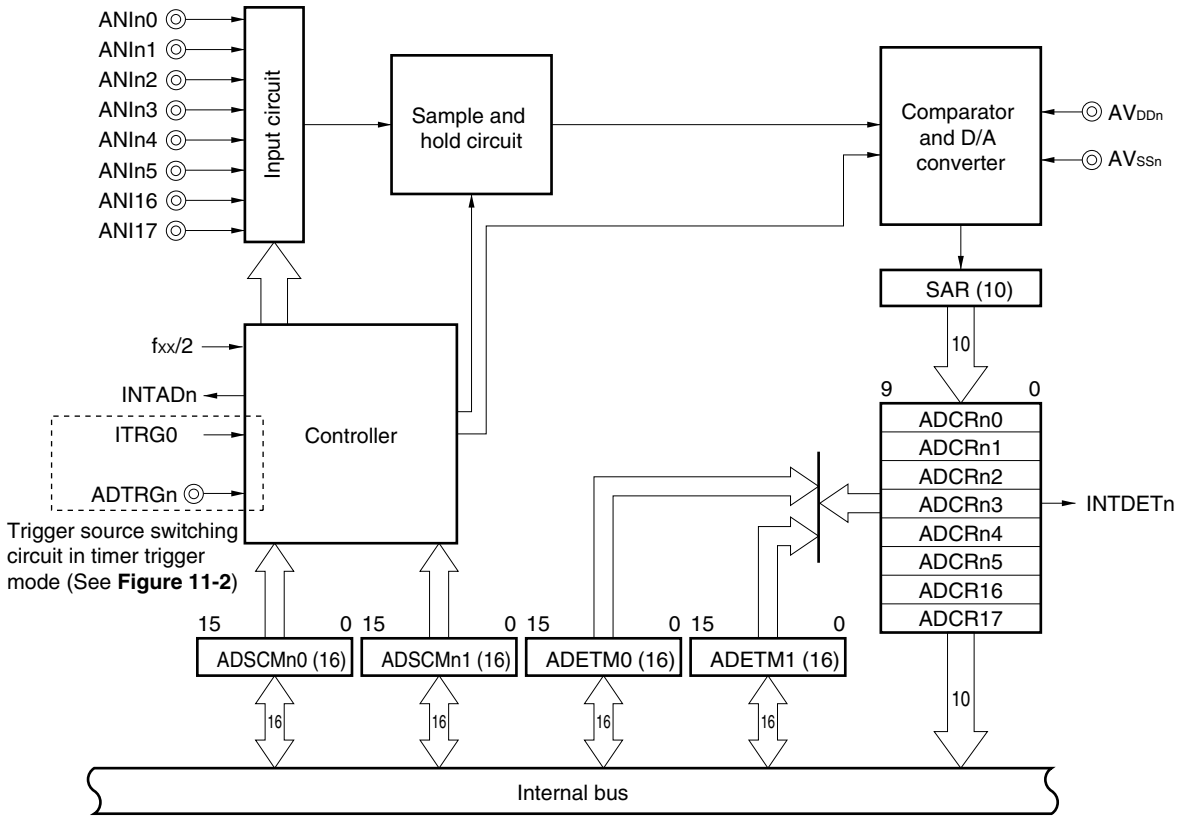
The AV_{SS0} and AV_{SS1} pins are the ground voltage pins of A/D converters 0 and 1. Even if not using A/D converters 0 and 1, always ensure these pins have the same potential as the V_{SS} pin.

(10) AV_{DD0}, AV_{DD1} pins

The AV_{DD0} and AV_{DD1} pins are the analog power supply pins of A/D converters 0 and 1. These pins are also used as pins that input a reference voltage (equivalent to the AV_{REF0} and AV_{REF1} pins of the V850E/IA1). Therefore, the signals input to the ANI0m and ANI1n pins are converted into digital signals, based on the voltage applied between AV_{DD0} and AV_{SS0} and between AV_{DD1} and AV_{SS1} (m = 0 to 5, n = 0 to 7).

Even if not using A/D converters 0 and 1, always ensure these pins have the same potential as the V_{DD} pin.

Figure 11-1. Block Diagram of A/D Converter 0 or 1



Remark n = 0, 1

fxx: Internal system clock

Cautions 1. Noise at an analog input pin (ANI0m, ANI1n) or reference voltage input pin (AV_{DD0}, AV_{DD1}) may give rise to an invalid conversion result (m = 0 to 5, n = 0 to 7).

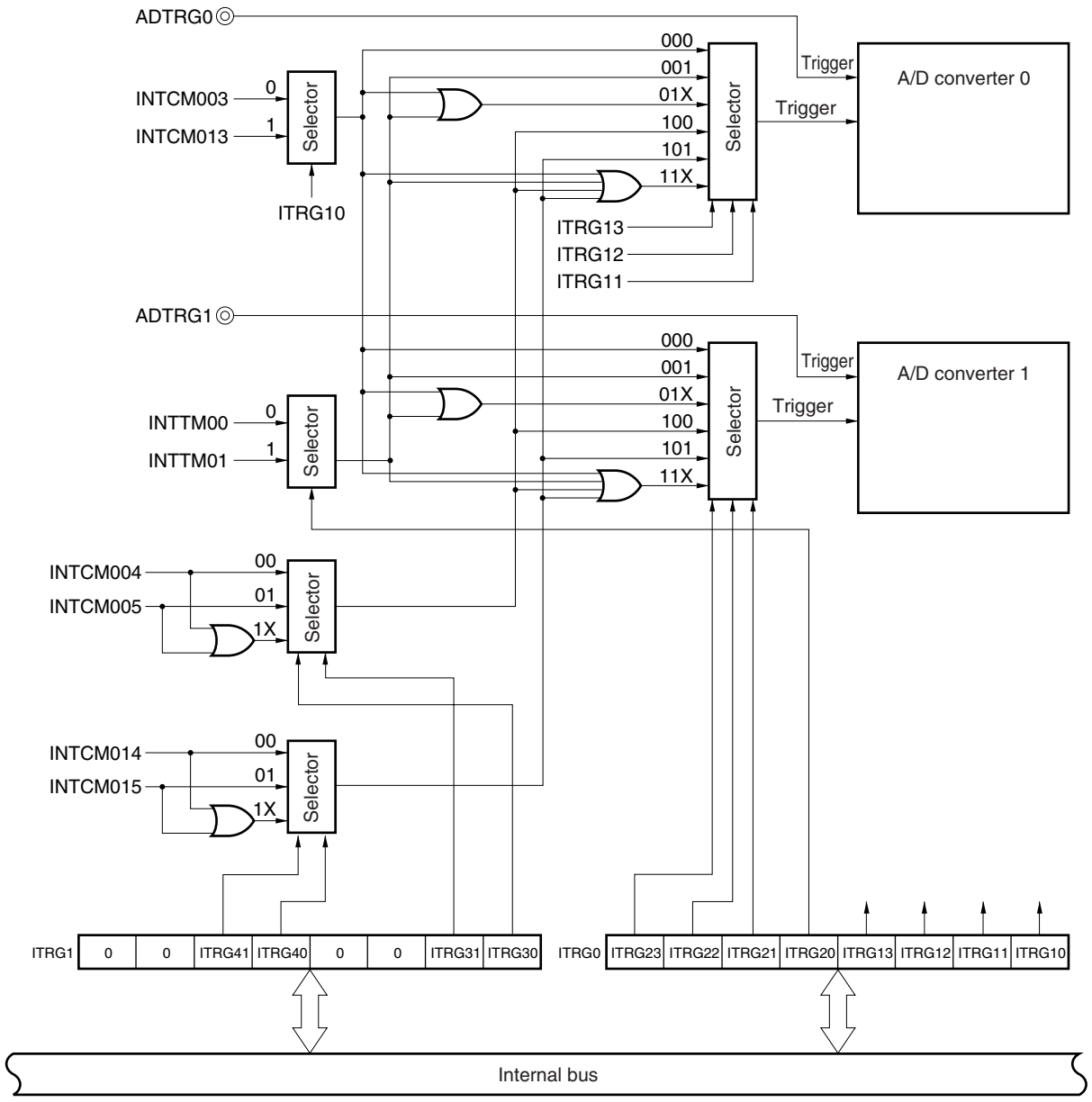
Software processing is needed in order to prevent this invalid conversion result from adversely affecting the system.

The following are examples of software processing.

- **Use the average value of the results of multiple A/D conversions as the A/D conversion result.**
- **Perform A/D conversion several times consecutively and use conversion results omitting any abnormal conversion results that are obtained.**
- **If an A/D conversion result from which it is judged that an abnormality occurred in the system is obtained, be sure to recheck the abnormality occurrence before performing malfunction processing.**

2. Be sure that voltages outside the range [AV_{SS0} to AV_{DD0}, AV_{SS1} to AV_{DD1}] are not applied to pins being used as A/D converter 0 and 1 input pins.

Figure 11-2. Block Diagram of Trigger Source Switching Circuit in Timer Trigger Made



Caution For the selection of the trigger source in timer trigger mode, refer to Table 11-4 Timer Trigger Source Selection of A/D Converters 0 and 1.

Remark X: Don't care

11.3 Functions Added to V850E/IA2

(1) Addition of INTCM004, INTCM005, INTCM014, INTCM015 as timer trigger sources

The timer trigger source (INTTM0n, INTCM0n3 to INTCM0n5) is selected using A/D internal trigger selection registers 0 and 1 (ITRG0 and ITRG1) when the timer trigger mode is set by A/D scan mode registers 00 and 10 (ADSCM00 and ADSCM10).

With the V850E/IA2, bit 3 (ITRG13) and bit 7 (ITRG23) of the ITRG0 register, as well as the ITRG1 register have been added.

(2) Changing analog input to a total of 14 channels for two circuits

(3) Multiplexing AV_{REF0} and AV_{REF1} with AV_{DD0} and AV_{DD1}

11.4 Control Registers

(1) A/D scan mode registers 00 and 10 (ADSCM00, ADSCM10)

The ADSCMn0 registers are 16-bit registers that select analog input pins, specify operation modes, and control conversion operations.

They can be read or written in 16-bit units.

When the higher 8 bits of the ADSCMn0 register are used as the ADSCMn0H register and the lower 8 bits are used as the ADSCMn0L register, they can be read/written in 8-bit or 1-bit units.

However, writing to the ADSCMn0 register during A/D conversion initializes conversion and starts the conversion operation from the beginning.

Caution Clear (0) the ADCEn bit before changing the trigger mode using the ADPLMn and TRG2 to TRG0 bits (n = 0, 1). If the changing of the trigger mode and clearing of the ADCEn bits are performed simultaneously (same instruction), operation is not guaranteed. Be sure to perform register access twice.

(1/2)

	<15>	<14>	13	<12>	<11>	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
ADSCM00	ADCE0	ADCS0	0	ADMS0	ADPLM0	TRG2	TRG1	TRG0	SANI3	SANI2	SANI1	SANI0	ANIS3	ANIS2	ANIS1	ANIS0	FFFFF200H	0000H

	<15>	<14>	13	<12>	<11>	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
ADSCM10	ADCE1	ADCS1	0	ADMS1	ADPLM1	TRG2	TRG1	TRG0	SANI3	SANI2	SANI1	SANI0	ANIS3	ANIS2	ANIS1	ANIS0	FFFFF240H	0000H

Bit position	Bit name	Function
15	ADCEn	Specifies enabling or disabling A/D conversion. 0: Disable 1: Enable
14	ADCSn	Shows status of A/D converter 0 or 1. This bit is read-only. 0: Stopped 1: Operating ADCSn bit is 0 during the period of 6 × fxx/2 immediately after the start of A/D conversion, and then set to 1. This operation is performed each time an analog input pin has been switched for A/D conversion in the scan mode.
12	ADMSn	Specifies operation mode of A/D converter 0 or 1. 0: Scan mode 1: Select mode
11 to 8	ADPLMn, TRG2 to TRG0	ADPLMn: Specifies polling mode. TRG2 to TRG0: Specifies trigger mode.

ADPLMn	TRG2	TRG1	TRG0	Trigger mode
0	0	0	0	A/D trigger mode
0	0	0	1	Timer trigger mode
0	1	1	1	External trigger mode
1	0	0	0	A/D trigger polling mode
Other than above				Setting prohibited

Remark n = 0, 1

(2/2)

Bit position	Bit name	Function
7 to 4	SANI3 to SANI0	Specifies conversion start analog input pin in scan mode. These bits are ignored in select mode.

SANI3	SANI2	SANI1	SANI0	Scan start analog input pin
0	0	0	0	ANIn0
0	0	0	1	ANIn1
0	0	1	0	ANIn2
0	0	1	1	ANIn3
0	1	0	0	ANIn4
0	1	0	1	ANIn5
0	1	1	0	ANI16
0	1	1	1	ANI17
Other than above				Setting prohibited

Caution Always set the conversion start analog input pin number that is set by bits SANI3 to SANI0 to a smaller pin number than the conversion end analog input pin number that is set by bits ANIS3 to ANIS0.

Bit position	Bit name	Function
3 to 0	ANIS3 to ANIS0	Specifies analog input pin in select mode. In scan mode, specifies conversion termination analog input pin.

ANIS3	ANIS2	ANIS1	ANIS0	In select mode	In scan mode
0	0	0	0	ANIn0	ANIn0
0	0	0	1	ANIn1	SANI → ANIn1
0	0	1	0	ANIn2	SANI → ANIn2
0	0	1	1	ANIn3	SANI → ANIn3
0	1	0	0	ANIn4	SANI → ANIn4
0	1	0	1	ANIn5	SANI → ANIn5
0	1	1	0	ANI16	SANI → ANI16
0	1	1	1	ANI17	SANI → ANI17
Other than above				Setting prohibited	

Remark SANI < ANInm
Where n = 0: m = 1 to 5
Where n = 1: m = 1 to 7

Remark n = 0, 1

(2) A/D scan mode registers 01 and 11 (ADSCM01, ADSCM11)

The ADSCMn1 registers are 16-bit registers that set the conversion time of the A/D converter.

They can be read or written in 16-bit units.

When the higher 8 bits of the ADSCMn1 register are used as the ADSCMn1H register, and the lower 8 bits are used as the ADSCMn1L register, the ADSCMn1H register can be read/written in 8-bit units, and the ADSCMn1L register is read-only, in 8-bit units.

Caution Do not write to the ADSCMn1 registers during an A/D conversion operation. If a write is performed, the conversion operation is suspended and subsequently terminates.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
ADSCM01	0	0	0	0	0	FR2	FR1	FR0	0	0	0	0	0	0	0	0	FFFFF202H	0000H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
ADSCM11	0	0	0	0	0	FR2	FR1	FR0	0	0	0	0	0	0	0	0	FFFFF242H	0000H

Bit position	Bit name	Function
10 to 8	FR2 to FR0	Specifies conversion time.

FR2	FR1	FR0	Conversion clocks	Conversion time (μs)[Note]	
				fxx = 40 MHz	fxx = 33 MHz
0	0	0	344	8.60	–
0	0	1	248	6.20	7.51
0	1	0	176	–	5.33
0	1	1	128	–	–
1	0	0	104	–	–
1	0	1	80	–	–
1	1	0	56	–	–
1	1	1	Setting prohibited	–	

Note This is the time from sampling until conversion end.
Sampling time = (Conversion clocks – 8)/6 × fxx

Caution Be sure to secure the conversion time within a range of 5 to 10 μs.
Conversion time = fxx × Conversion clocks

Remark fxx: Internal system clock

(3) A/D voltage detection mode registers 0 and 1 (ADETM0, ADETM1)

The ADETMn registers are 16-bit registers that set the voltage detection mode. In the voltage detection mode, the analog input pin for which voltage detection is being performed and a reference voltage value are compared and an interrupt is set in response to the comparison result.

These registers can be read or written in 16-bit units.

When the higher 8 bits of the ADETMn register are used as the ADETMnH register, and the lower 8 bits are used as the ADETMnL register, they can be read/written in 8-bit or 1-bit units.

Caution Do not write to an ADETMn register during an A/D conversion operation. If a write is performed, conversion is suspended and it subsequently terminates.

	<15>	<14>	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
ADETM0	ADETEN0	ADETLH0	DETANI3	DETANI2	DETANI1	DETANI0	DETCMP9	DETCMP8	DETCMP7	DETCMP6	DETCMP5	DETCMP4	DETCMP3	DETCMP2	DETCMP1	DETCMP0	FFFFF204H	0000H

	<15>	<14>	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
ADETM1	ADETEN1	ADETLH1	DETANI3	DETANI2	DETANI1	DETANI0	DETCMP9	DETCMP8	DETCMP7	DETCMP6	DETCMP5	DETCMP4	DETCMP3	DETCMP2	DETCMP1	DETCMP0	FFFFF244H	0000H

Bit position	Bit name	Function
15	ADETENn	Specifies voltage detection mode. 0: Operates in normal mode 1: Operates in voltage detection mode
14	ADETLHn	Sets voltage comparison detection. 0: Generates INTDETn interrupt if reference voltage value > analog input pin voltage. 1: Generates INTDETn interrupt if reference voltage value < analog input pin voltage.
13 to 10	DETANI3 to DETANI0	Selects analog input pin to compare to reference voltage value set by DETCMP9 to DETCMP0 when in voltage detection mode.
9 to 0	DETCMP9 to DETCMP0	Sets reference voltage value to compare with analog input pin selected by DETANI3 to DETANI0.

DETANI3	DETANI2	DETANI1	DETANI0	Voltage detection analog input pin
0	0	0	0	ANIn0
0	0	0	1	ANIn1
0	0	1	0	ANIn2
0	0	1	1	ANIn3
0	1	0	0	ANIn4
0	1	0	1	ANIn5
0	1	1	0	ANI16
0	1	1	1	ANI17
1	×	×	×	Setting prohibited

Remark ×: Don't care

Remark n = 0, 1

(4) A/D conversion result registers 00 to 05 and 10 to 17 (ADCR00 to ADCR05, ADCR10 to ADCR17)

The ADCR0m and ADCR1n registers are 10-bit registers that hold the results of A/D conversions (m = 0 to 5, n = 0 to 7). A/D converter 0 has six 10-bit registers for six channels and A/D converter 1 has eight 10-bit registers for eight channels. In all, fourteen 10-bit registers are available.

These registers are read-only, in 16-bit units.

When reading 10 bits of data of an A/D conversion result from the ADCR0m or ADCR1n register, only the lower 10 bits are valid and the higher 6 bits are always read as 0.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
ADCR0m	0	0	0	0	0	0	ADCRm9	ADCRm8	ADCRm7	ADCRm6	ADCRm5	ADCRm4	ADCRm3	ADCRm2	ADCRm1	ADCRm0	See **Table 11-1**	0000H
(m = 0 to 5)																		
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
ADCR1n	0	0	0	0	0	0	ADCRn9	ADCRn8	ADCRn7	ADCRn6	ADCRn5	ADCRn4	ADCRn3	ADCRn2	ADCRn1	ADCRn0	See **Table 11-2**	0000H
(n = 0 to 7)																		

Table 11-1. Correspondence Between ADCR0m (m = 0 to 5) Register Names and Addresses

Register Name	Address
ADCR00	FFFFF210H
ADCR01	FFFFF212H
ADCR02	FFFFF214H
ADCR03	FFFFF216H
ADCR04	FFFFF218H
ADCR05	FFFFF21AH

Table 11-2. Correspondence Between ADCR1n (n = 0 to 7) Register Names and Addresses

Register Name	Address
ADCR10	FFFFF250H
ADCR11	FFFFF252H
ADCR12	FFFFF254H
ADCR13	FFFFF256H
ADCR14	FFFFF258H
ADCR15	FFFFF25AH
ADCR16	FFFFF25CH
ADCR17	FFFFF25EH

The correspondence between the analog input pins and the ADCR0m and ADCR1n registers is shown below.

Table 11-3. Correspondence Between Analog Input Pins and ADCR0m and ADCR1n Registers

A/D Converter	Analog Input Pin	A/D Conversion Result Register
A/D converter 0	ANI00	ADCR00
	ANI01	ADCR01
	ANI02	ADCR02
	ANI03	ADCR03
	ANI04	ADCR04
	ANI05	ADCR05
A/D converter 1	ANI10	ADCR10
	ANI11	ADCR11
	ANI12	ADCR12
	ANI13	ADCR13
	ANI14	ADCR14
	ANI15	ADCR15
	ANI16	ADCR16
	ANI17	ADCR17

The relationship between the analog voltage input to an analog input pin (ANI0m or ANI1n) and the value of the A/D conversion result register (ADCR0m or ADCR1n) is as follows (m = 0 to 5, n = 0 to 7):

$$ADCR = INT\left(\frac{V_{IN}}{AV_{DD}} \times 1{,}024 + 0.5\right)$$

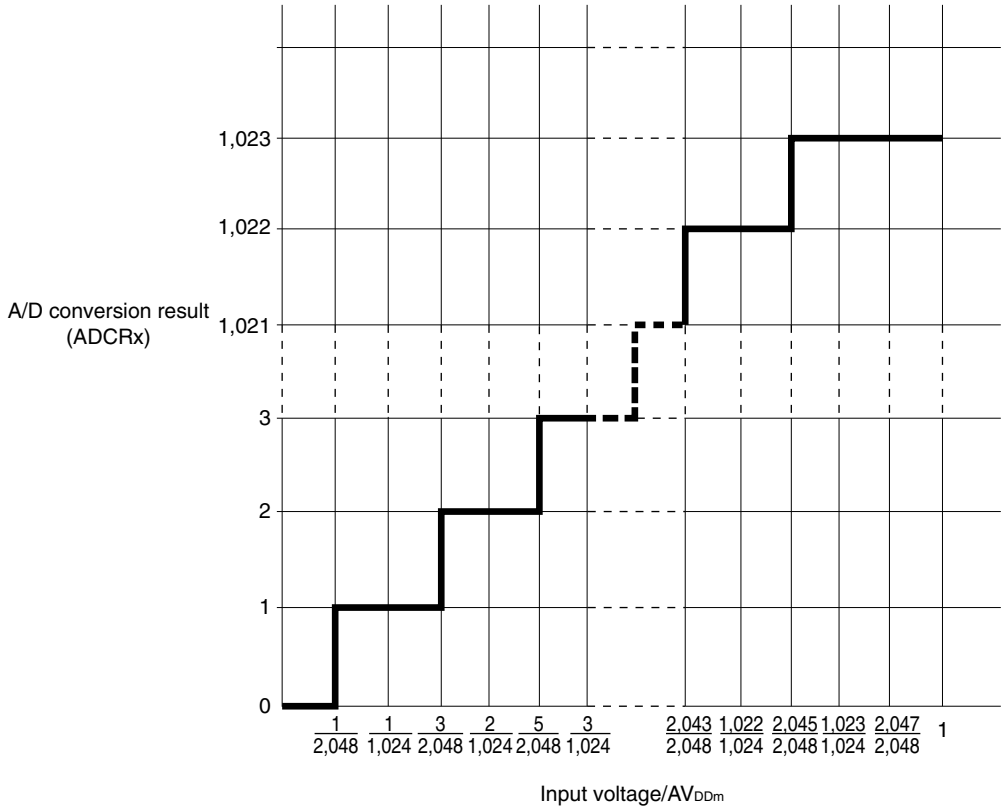
Or,

$$(ADCR - 0.5) \times \frac{AV_{DD}}{1{,}024} \leq V_{IN} < (ADCR + 0.5) \times \frac{AV_{DD}}{1{,}024}$$

INT ( ): Function that returns integer of value in ( )
V_{IN}: Analog input voltage
AV_{DD}: AV_{DD0} or AV_{DD1} pin voltage
ADCR: Value of A/D conversion result register (ADCR0m or ADCR1n)

Figure 11-3 illustrates the relationship between the analog input voltages and A/D conversion results.

Figure 11-3. Relationship Between Analog Input Voltages and A/D Conversion Results



Remark m = 0, 1
x = 00 to 05, 10 to 17

(5) A/D internal trigger selection registers 0, 1 (ITRG0, ITRG1)

The ITRGn register switches the trigger source in timer trigger mode. The timer trigger source of A/D converters 0 and 1 can be set using the ITRGn register.

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

	7	6	5	4	3	2	1	0	Address	After reset
ITRG0	ITRG23	ITRG22	ITRG21	ITRG20	ITRG13	ITRG12	ITRG11	ITRG10	FFFFF280H	00H

	7	6	5	4	3	2	1	0	Address	After reset
ITRG1	0	0	ITRG41	ITRG40	0	0	ITRG31	ITRG30	FFFFF288H	00H

Bit position	Bit name	Function
7 to 0 (ITRG0) 5, 4, 1, 0 (ITRG1)	ITRG23 to ITRG20, ITRG13 to ITRG10 (ITRG0) ITRG41, ITRG40, ITRG31, ITRG30 (ITRG1)	Specifies timer trigger source of A/D converters 0 and 1 (refer to **Table 11-4 Timer Trigger Source Selection of A/D Converters 0 and 1**).

Table 11-4. Timer Trigger Source Selection of A/D Converters 0 and 1 (1/3)

ITRGm3	ITRGm2	ITRGm1	ITRG41	ITRG40	ITRG31	ITRG30	ITRG20	ITRG10	Trigger Source of A/D Converter n
0	0	0	×	×	×	×	×	0	Selects INTCM003
0	0	0	×	×	×	×	×	1	Selects INTCM013
0	0	1	×	×	×	×	0	×	Selects INTTM00
0	0	1	×	×	×	×	1	×	Selects INTTM01
0	1	×	×	×	×	×	0	0	Selects INTCM003, INTTM00
0	1	×	×	×	×	×	0	1	Selects INTCM013, INTTM00
0	1	×	×	×	×	×	1	0	Selects INTCM003, INTTM01
0	1	×	×	×	×	×	1	1	Selects INTCM013, INTTM01
1	0	0	×	×	0	0	×	×	Selects INTCM004
1	0	0	×	×	0	1	×	×	Selects INTCM005
1	0	0	×	×	1	×	×	×	Selects INTCM004, INTCM005
1	0	1	0	0	×	×	×	×	Selects INTCM014
1	0	1	0	1	×	×	×	×	Selects INTCM015
1	0	1	1	×	×	×	×	×	Selects INTCM014, INTCM015
1	1	×	0	0	0	0	0	0	Selects INTCM003, INTTM00, INTCM004, INTCM014
1	1	×	0	0	0	0	0	1	Selects INTCM013, INTTM00, INTCM004, INTCM014
1	1	×	0	0	0	0	1	0	Selects INTCM003, INTTM01, INTCM004, INTCM014
1	1	×	0	0	0	0	1	1	Selects INTCM013, INTTM01, INTCM004, INTCM014
1	1	×	0	0	0	1	0	0	Selects INTCM003, INTTM00, INTCM005, INTCM014
1	1	×	0	0	0	1	0	1	Selects INTCM013, INTTM00, INTCM005, INTCM014
1	1	×	0	0	0	1	1	0	Selects INTCM003, INTTM01, INTCM005, INTCM014
1	1	×	0	0	0	1	1	1	Selects INTCM013, INTTM01, INTCM005, INTCM014
1	1	×	0	0	1	×	0	0	Selects INTCM003, INTTM00, INTCM004, INTCM005, INTCM014
1	1	×	0	0	1	×	0	1	Selects INTCM013, INTTM00, INTCM004, INTCM005, INTCM014
1	1	×	0	0	1	×	1	0	Selects INTCM003, INTTM01, INTCM004, INTCM005, INTCM014

Remarks 1. n = 0, 1

Where n = 0: m = 1

Where n = 1: m = 2

2. ×: Don't care

Table 11-4. Timer Trigger Source Selection of A/D Converters 0 and 1 (2/3)

ITRGm3	ITRGm2	ITRGm1	ITRG41	ITRG40	ITRG31	ITRG30	ITRG20	ITRG10	Trigger Source of A/D Converter n
1	1	×	0	0	1	×	1	1	Selects INTCM013, INTTM01, INTCM004, INTCM005, INTCM014
1	1	×	0	1	0	0	0	0	Selects INTCM003, INTTM00, INTCM004, INTCM015
1	1	×	0	1	0	0	0	1	Selects INTCM013, INTTM00, INTCM004, INTCM015
1	1	×	0	1	0	0	1	0	Selects INTCM003, INTTM01, INTCM004, INTCM015
1	1	×	0	1	0	0	1	1	Selects INTCM013, INTTM01, INTCM004, INTCM015
1	1	×	0	1	0	1	0	0	Selects INTCM003, INTTM00, INTCM005, INTCM015
1	1	×	0	1	0	1	0	1	Selects INTCM013, INTTM00, INTCM005, INTCM015
1	1	×	0	1	0	1	1	0	Selects INTCM003, INTTM01, INTCM005, INTCM015
1	1	×	0	1	0	1	1	1	Selects INTCM013, INTTM01, INTCM005, INTCM015
1	1	×	0	1	1	×	0	0	Selects INTCM003, INTTM00, INTCM004, INTCM005, INTCM015
1	1	×	0	1	1	×	0	1	Selects INTCM013, INTTM00, INTCM004, INTCM005, INTCM015
1	1	×	0	1	1	×	1	0	Selects INTCM003, INTTM01, INTCM004, INTCM005, INTCM015
1	1	×	0	1	1	×	1	1	Selects INTCM013, INTTM01, INTCM004, INTCM005, INTCM015
1	1	×	1	×	0	0	0	0	Selects INTCM003, INTTM00, INTCM004, INTCM014, INTCM015
1	1	×	1	×	0	0	0	1	Selects INTCM013, INTTM00, INTCM004, INTCM014, INTCM015
1	1	×	1	×	0	0	1	0	Selects INTCM003, INTTM01, INTCM004, INTCM014, INTCM015
1	1	×	1	×	0	0	1	1	Selects INTCM013, INTTM01, INTCM004, INTCM014, INTCM015
1	1	×	1	×	0	1	0	0	Selects INTCM003, INTTM00, INTCM005, INTCM014, INTCM015
1	1	×	1	×	0	1	0	1	Selects INTCM013, INTTM00, INTCM005, INTCM014, INTCM015
1	1	×	1	×	0	1	1	0	Selects INTCM003, INTTM01, INTCM005, INTCM014, INTCM015

Remarks 1. n = 0, 1

Where n = 0: m = 1

Where n = 1: m = 2

2. ×: Don't care

Table 11-4. Timer Trigger Source Selection of A/D Converters 0 and 1 (3/3)

ITRGm3	ITRGm2	ITRGm1	ITRG41	ITRG40	ITRG31	ITRG30	ITRG20	ITRG10	Trigger Source of A/D Converter n
1	1	×	1	×	0	1	1	1	Selects INTCM013, INTTM01, INTCM005, INTCM014, INTCM015
1	1	×	1	×	1	×	0	0	Selects INTCM003, INTTM00, INTCM004, INTCM005, INTCM014, INTCM015
1	1	×	1	×	1	×	0	1	Selects INTCM013, INTTM00, INTCM004, INTCM005, INTCM014, INTCM015
1	1	×	1	×	1	×	1	0	Selects INTCM003, INTTM01, INTCM004, INTCM005, INTCM014, INTCM015
1	1	×	1	×	1	×	1	1	Selects INTCM013, INTTM01, INTCM004, INTCM005, INTCM014, INTCM015

Remarks 1. n = 0, 1

Where n = 0: m = 1

Where n = 1: m = 2

2. ×: Don't care

11.5 Interrupt Requests

A/D converters 0 and 1 generate two kinds of interrupts.

- A/D conversion end interrupts (INTAD0, INTAD1)
- Voltage detection interrupts (INTDET0, INTDET1)

(1) A/D conversion end interrupts (INTAD0, INTAD1)

In the A/D conversion enabled status, an A/D conversion end interrupt is generated when a specified number of A/D conversions have been completed.

A/D Converter	A/D Conversion End Interrupt Signal
0	Generates INTAD0
1	Generates INTAD1

(2) Voltage detection interrupt (INTDET0, INTDET1)

In the voltage detection mode (ADETEN0 or ADETEN1 bit of ADETM0 or ADETM1 = 1), the value of the ADCR0m or ADCR1n register of the relevant analog input pin is compared with the reference voltage set in the DETCMP9 to DETCMP0 bits of the ADETM0 or ADETM1 register and a voltage detection interrupt is generated in response to the value of the ADETLH0 or ADETLH1 bit of the ADETM0 or ADETM1 register (m = 0 to 5, n = 0 to 7).

A/D Converter	Voltage Detection Interrupt Signal
0	Generates INTDET0
1	Generates INTDET1

11.6 A/D Converter Operation

11.6.1 A/D converter basic operation

A/D conversion is performed using the following procedure.

(1) Set the analog input selection and the operation mode and trigger mode specifications using the ADSCM00 or ADSCM10 register[Note 1]. Setting (1) the ADCE0 or ADCE1 bit of the ADSCM00 or ADSCM10 register when in A/D trigger mode or A/D trigger polling mode starts A/D conversion. In timer trigger mode or external trigger mode, the status becomes trigger standby[Note 2].

(2) When A/D conversion starts, compare the analog input with the voltage generated by the D/A converter.

(3) When 10-bit comparison ends, store the conversion result in the ADCR0m or ADCR1n register. When the specified number of A/D conversions have ended, generate the A/D conversion end interrupt (INTAD0, INTAD1) (m = 0 to 5, n = 0 to 7).

Notes 1. If the contents of the ADSCM00 or ADSCM10 register are changed during an A/D conversion operation, the A/D conversion operation preceding the change stops and a conversion result is not stored in the ADCR0m or ADCR1n register. The conversion operation is initialized and conversion starts from the beginning.

2. In timer trigger mode or external trigger mode, there is a transition to trigger standby status when the ADCE0 or ADCE1 bit of the ADSCM00 or ADSCM10 register is set to 1. An A/D conversion operation is activated by a trigger signal and there is a return to trigger standby status when the A/D conversion operation ends.

The timer trigger is selected by the ITRG0 and ITRG1 registers.

11.6.2 Operation modes and trigger modes

Diverse conversion operations can be specified for A/D converters 0 and 1 by specifying the operation mode and trigger mode. The operation mode and trigger mode are set using the ADSCM00 or ADSCM10 register.

The relationship between the operation mode and the trigger mode is shown below.

Trigger Mode	Operation Mode	Setting	
		ADSCM00	ADSCM10
AD trigger	Select	XX010000XXXXXXXXB	XX010000XXXXXXXXB
	Scan	XX000000XXXXXXXXB	XX000000XXXXXXXXB
AD trigger polling	Select	XX011000XXXXXXXXB	XX011000XXXXXXXXB
	Scan	XX001000XXXXXXXXB	XX001000XXXXXXXXB
Timer trigger	Select	XX010001XXXXXXXXB	XX010001XXXXXXXXB
	Scan	XX000001XXXXXXXXB	XX000001XXXXXXXXB
External trigger	Select	XX010111XXXXXXXXB	XX010111XXXXXXXXB
	Scan	XX000111XXXXXXXXB	XX000111XXXXXXXXB

(1) Trigger modes

Four trigger modes that serve as the start timing of A/D conversion processing are available: A/D trigger mode, A/D trigger polling mode, timer trigger mode, and external trigger mode.

These trigger modes are set using the ADSCM00 and ADSCM10 registers.

(a) A/D trigger mode

A/D trigger mode, which starts the conversion timing for the analog input set for the ANI0m or ANI1n pin (m = 0 to 5, n = 0 to 7), is a mode in which A/D conversion is started by setting the ADCE0 or ADCE1 bit of the ADSCM00 or ADSCM10 register to 1. In this mode, it is necessary to set the ADCE0 or ADCE1 bit to 1 as an A/D conversion restart operation after the INTAD0 or INTAD1 interrupt (ADCS0, ADCS1 = 0).

(b) A/D trigger polling mode

A/D trigger polling mode, which starts the conversion timing of the analog input set for the ANI0m or ANI1n pin (m = 0 to 5, n = 0 to 7), is a mode in which A/D conversion is started by setting the ADCE0 or ADCE1 bit of the ADSCM00 or ADSCM10 register to 1. In this mode, it is not necessary to set the ADCE0 or ADCE1 bit to 1 as an A/D conversion restart operation after the INTAD0 or INTAD1 interrupt (ADCS0, ADCS1 = 1). The specified analog input is converted serially until the ADCE0 or ADCE1 bit is set to 0. The INTAD0 or INTAD1 interrupt occurs each time a conversion ends.

(c) Timer trigger mode

Timer trigger mode, which starts the conversion timing of the analog input set for the ANI0m or ANI1n pin (m = 0 to 5, n = 0 to 7), is a mode governed by the trigger specified by the A/D internal trigger selection registers 0 and 1 (ITRG0, ITRG1).

(d) External trigger mode

External trigger mode, which starts the conversion timing of the analog input set using the ANI0m and ANI1n pins, is a mode specified using the ADTRG0 or ADTRG1 pin (m = 0 to 5, n = 0 to 7).

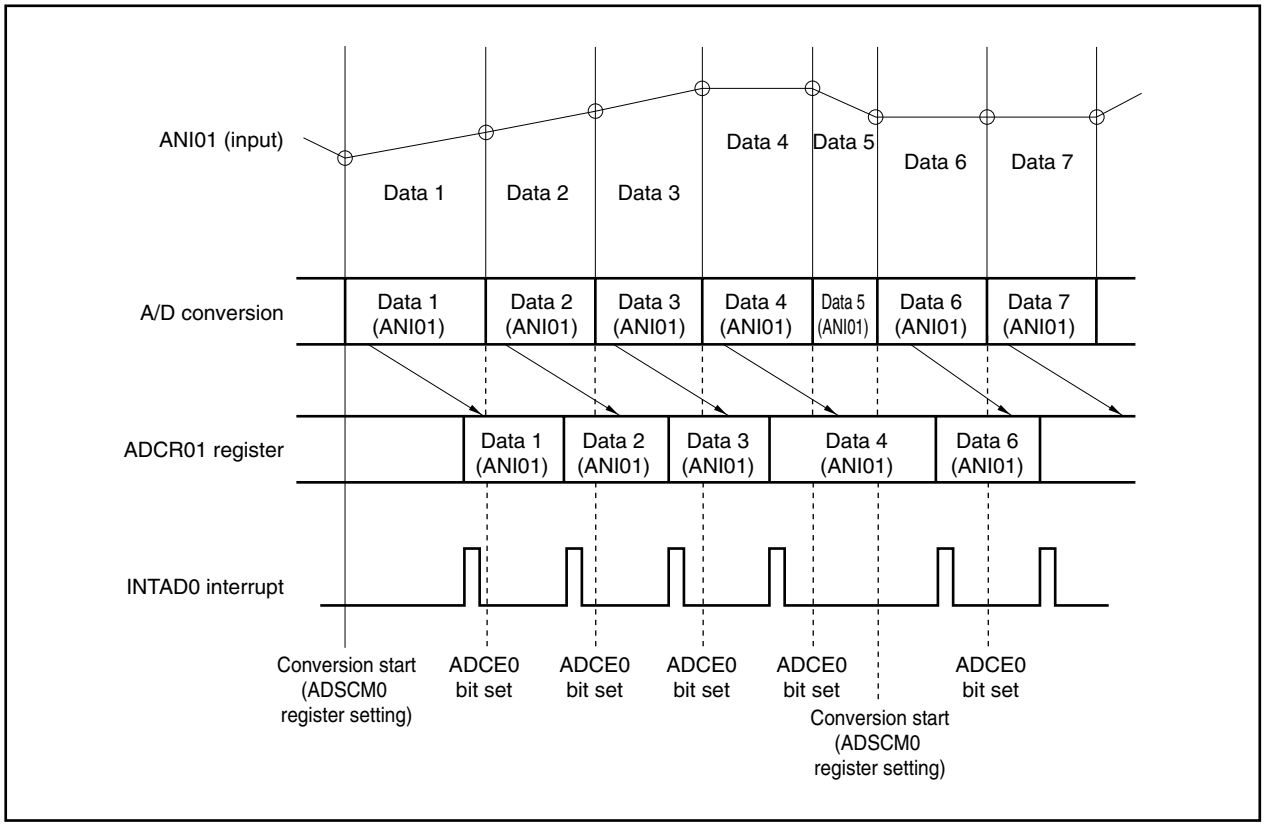
(2) Operation modes

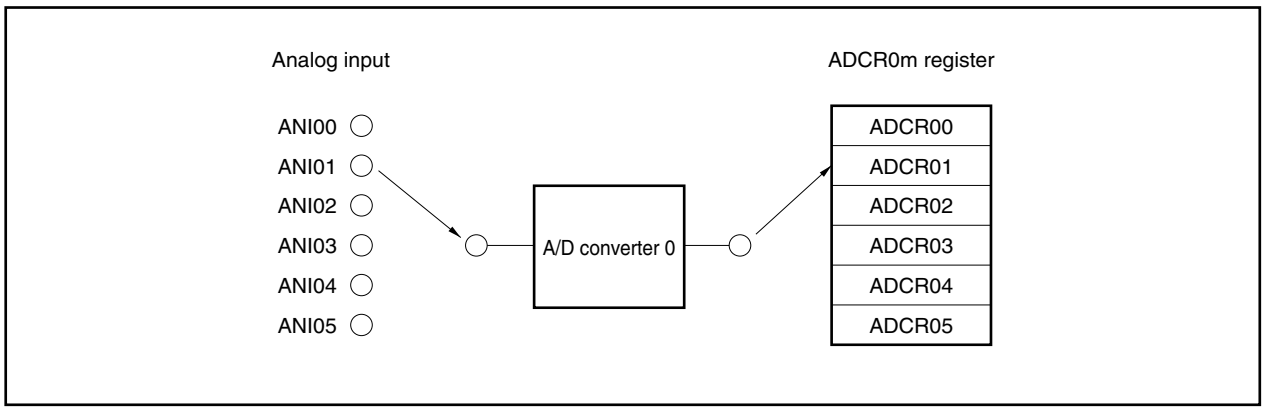
The two operation modes, which are the modes that set the ANI00 to ANI05 and ANI10 to ANI17 pins, are select mode and scan mode. These modes are set using the ADSCM00 and ADSCM10 registers.

(a) Select mode

In select mode, one analog input specified by the ADSCM00 or ADSCM10 register is A/D converted. The conversion result is stored in the ADCR0m or ADCR1n register corresponding to the analog input (ANI0m or ANI1n) (m = 0 to 5, n = 0 to 7).

Figure 11-4. Example of Select Mode Operation Timing (ANI01): For A/D Converter 0

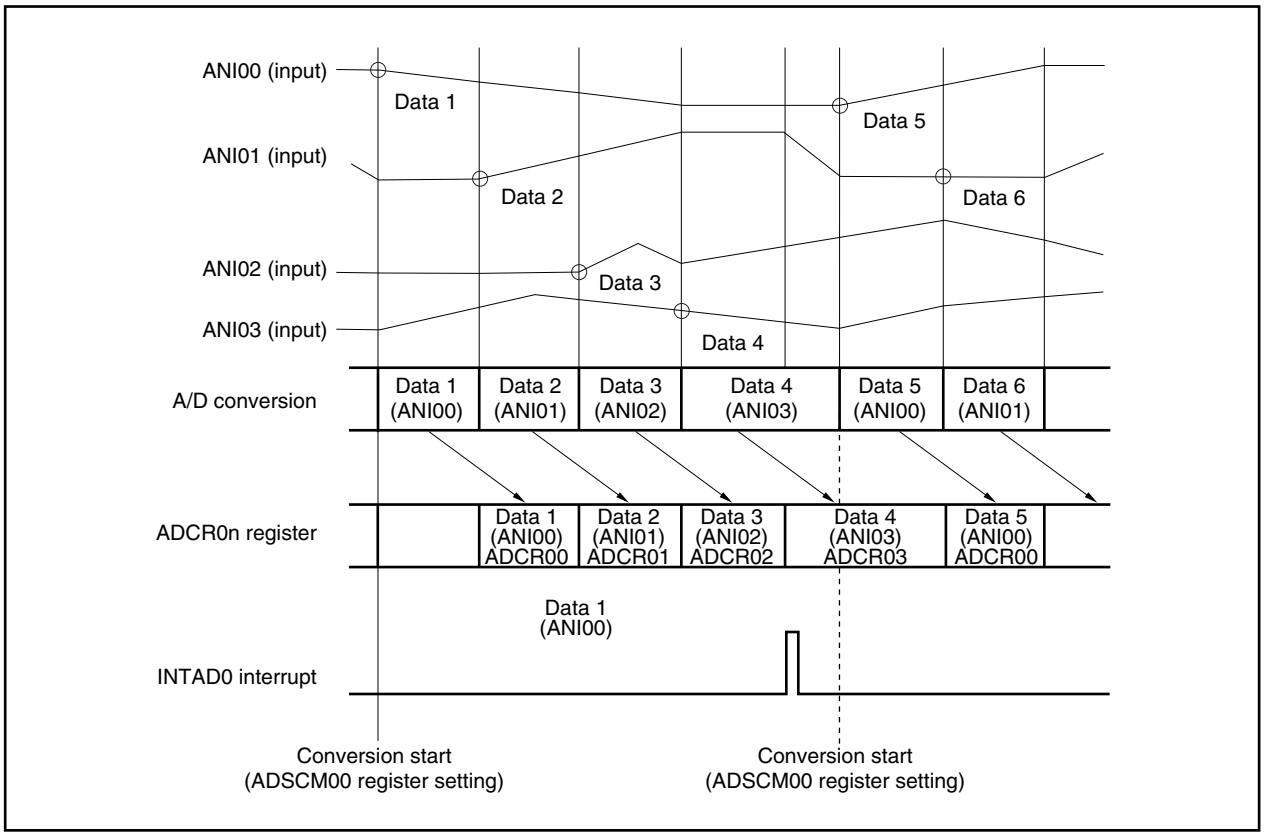


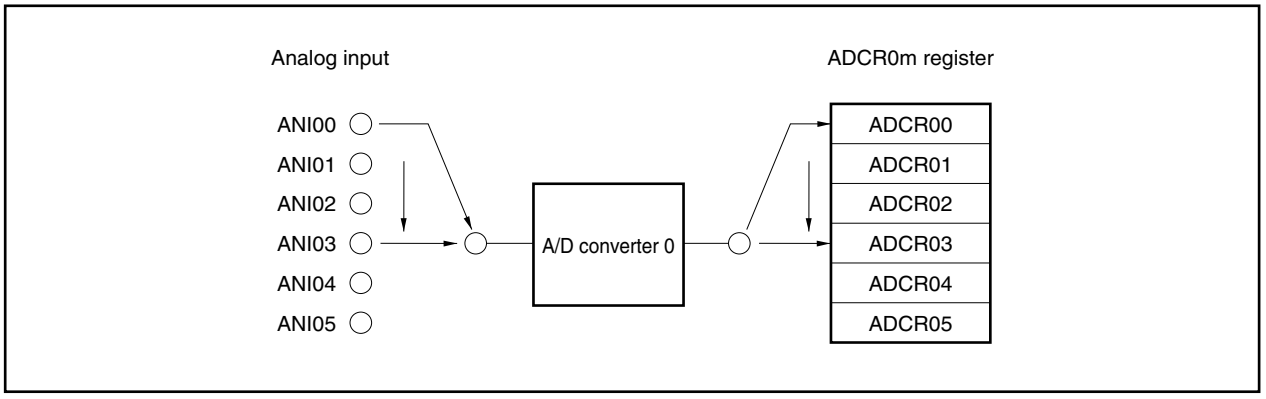


(b) Scan mode

In scan mode, pins from the A/D conversion start analog input pin to the A/D conversion termination analog input pin specified by the ADSCM00 or ADSCM10 register are sequentially selected and A/D converted. The A/D conversion result is stored in the ADCR0m or ADCR1n register corresponding to the analog input (m = 0 to 5, n = 0 to 7). When the specified analog input conversion ends, the A/D conversion end interrupt (INTAD0 or INTAD1) is generated.

Figure 11-5. Example of Scan Mode Operation Timing: For A/D Converter 0 (4-Channel Scan (ANI00 to ANI03))





11.7 Operation in A/D Trigger Mode

Setting the ADCE0 or ADCE1 bit of the ADSCM00 or ADSCM10 register to 1 starts A/D conversion.

11.7.1 Operation in select mode

One analog input specified by the ADSCM00 or ADSCM10 register is A/D converted at a time and the result is stored in the ADCR0m or ADCR1n register. Analog inputs correspond one-to-one with the ADCR0m or ADCR1n register (m = 0 to 5, n = 0 to 7).

The A/D conversion end interrupt (INTAD0, INTAD1) is generated at the end of each A/D conversion, which terminates A/D conversion (ADCS0, ADCS1 bit = 0).

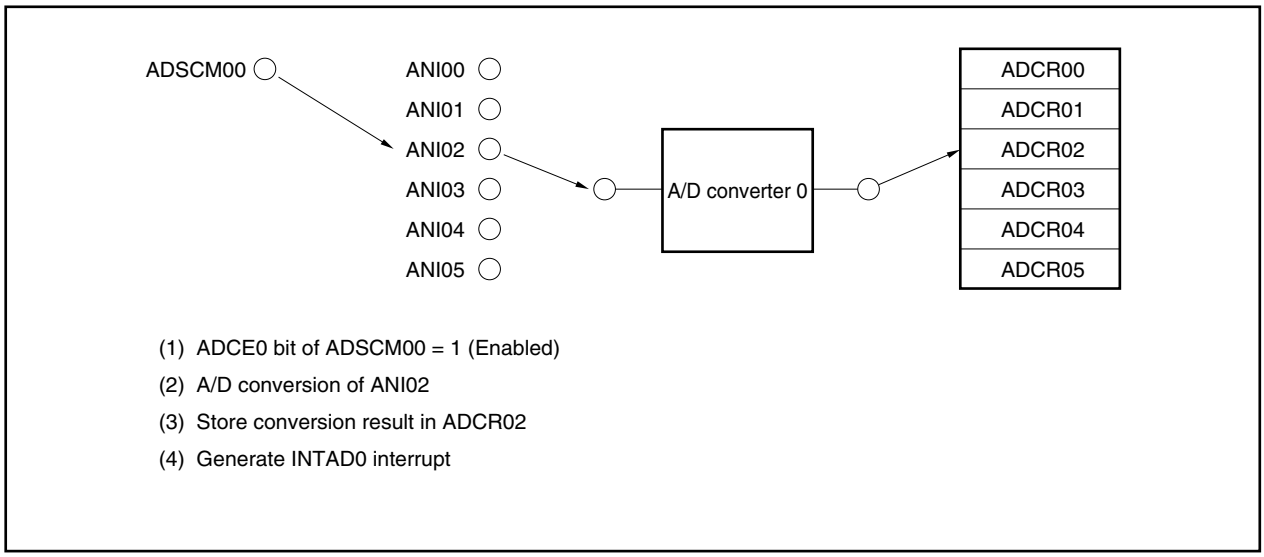
Analog Input	A/D Conversion Result Register
ANIx	ADCRx

Remark x = 00 to 05, 10 to 17

To restart A/D conversion, write 1 in the ADCE0 or ADCE1 bit of the ADSCM00 or ADSCM10 register.

This is optimal for an application that reads a result for each A/D conversion.

Figure 11-6. Example of Select Mode (A/D Trigger Select) Operation (ANI02): For A/D Converter 0



11.7.2 Operation in scan mode

Pins from the conversion start analog input pin to the conversion termination analog input pin specified by ADSCM00 or ADSCM10 register are sequentially selected and A/D converted. An A/D conversion result is stored in the ADCR0m or ADCR1n register corresponding to the analog input (m = 0 to 5, n = 0 to 7). When conversion ends for all analog inputs up to the conversion termination analog input pin, the A/D conversion end interrupt (INTAD0, INTAD1) is generated, which terminates A/D conversion (ADCS0 or ADCS1 bit of ADSCM0 or ADSCM1 register = 0).

Analog Input	A/D Conversion Result Register
ANIx[Note 1]	ADCRx
|	|
ANIx[Note 2]	ADCRx

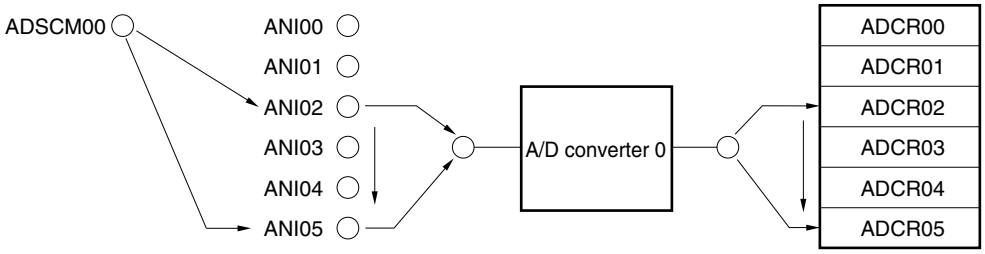
Notes 1. Set using the SANI3 to SANI0 bits of the ADSCM00 or ADSCM10 register.
Be sure to set a pin number that is smaller than the conversion termination analog input pin number set according to Note 2.

2. Set using the ANIS3 to ANIS0 bits of the ADSCM00 or ADSCM10 register.

Remark x = 00 to 05, 10 to 17

To restart A/D conversion, write 1 in the ADCE0 or ADCE1 bit of the ADSCM00 or ADSCM10 register. This is optimal for an application that regularly monitors multiple analog inputs.

Figure 11-7. Example of Scan Mode (A/D Trigger Scan) Operation (ANI02 to ANI05): For A/D Converter 0



(1) ADCE0 bit of ADSCM00 = 1 (Enabled)
(2) A/D conversion of ANI02
(3) Store conversion result in ADCR02
(4) A/D conversion of ANI03
(5) Store conversion result in ADCR03
(6) A/D conversion of ANI04
(7) Store conversion result in ADCR04
(8) A/D conversion of ANI05
(9) Store conversion result in ADCR05
(10) Generate INTAD0 interrupt

11.8 Operation in A/D Trigger Polling Mode

Setting the ADCE0 or ADCE1 bit of the ADSCM00 or ADSCM10 register to 1 starts A/D conversion.

Both select mode and scan mode are available in A/D trigger polling mode. Since the ADCS0 or ADCS1 bit of the ADSCM00 or ADSCM10 register remains 1 after the INTAD0 or INTAD1 interrupt in this mode, it is not necessary to write 1 in the ADCE0 or ADCE1 bit as an A/D conversion restart operation.

11.8.1 Operation in select mode

The analog input specified in the ADSCM00 or ADSCM10 register is A/D converted. The conversion result is stored in the ADCR0m or ADCR1n register (m = 0 to 5, n = 0 to 7).

One analog input is A/D converted at a time and the result is stored in one ADCR0m or ADCR1n register. Analog inputs correspond one-to-one with the ADCR0m or ADCR1n register.

An A/D conversion end interrupt (INTAD0 or INTAD1) is generated at the end of each A/D conversion. A/D conversion operations are repeated until the ADCE0 or ADCE1 bit = 0 (ADCS0, ADCS1 bit = 1).

Analog Input	A/D Conversion Result Register
ANIx	ADCRx

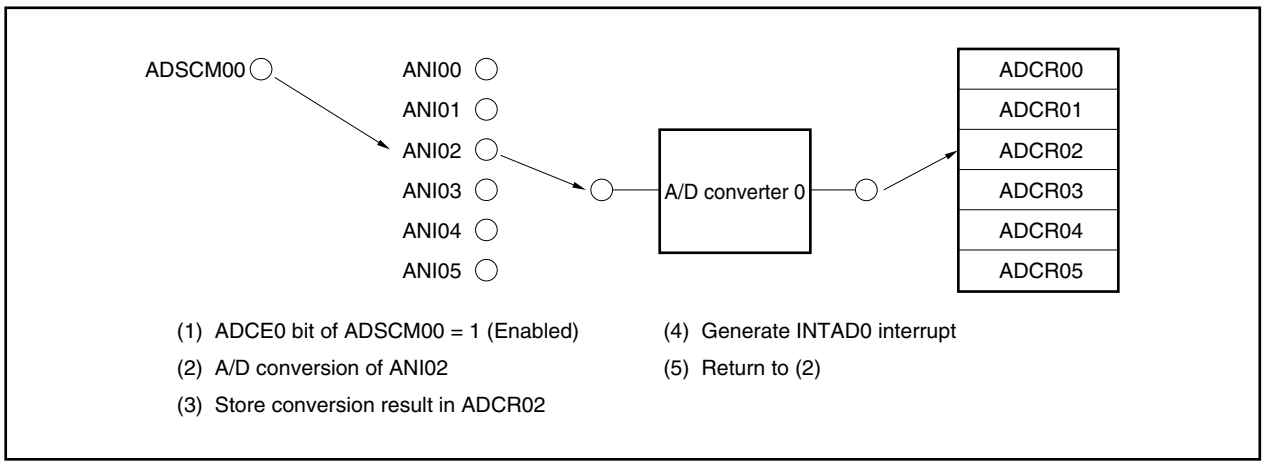
Remark x = 00 to 05, 10 to 17

In A/D trigger polling mode, it is not necessary to write 1 in the ADCE0 or ADCE1 bit of the ADSCM00 or ADSCM10 register as an A/D conversion restart operation[Note].

This is optimal for applications that regularly read A/D conversion values.

Note In A/D trigger polling mode, the fact that the ADCE0 or ADCE1 bit of the ADSCM00 or ADSCM10 register is 0 means that A/D conversion does not stop as long as the ADCS0 or ADCS1 bit is not 0. Therefore, if the ADCR0m or ADCR1n register is not read before the next A/D conversion, it is overwritten.

Figure 11-8. Example of Select Mode (A/D Trigger Polling Select) Operation (ANI02): For A/D Converter 0



11.8.2 Operation in scan mode

Pins from the conversion start analog input pin to the conversion termination analog input pin specified by the ADSCM00 or ADSCM10 register are sequentially selected and A/D converted. The A/D conversion result is stored in the ADCR0m or ADCR1n register corresponding to the analog input (m = 0 to 5, n = 0 to 7). When conversion ends for all analog inputs up to the conversion termination analog input pin, the A/D conversion end interrupt (INTAD0, INTAD1) is generated. A/D conversion repeats until the ADCE0 or ADCE1 bit = 0 (ADCS0, ADCS1 bit = 1).

Analog Input	A/D Conversion Result Register
ANIx[Note 1]	ADCRx
|	|
ANIx[Note 2]	ADCRx

Notes 1. Set using the SANI3 to SANI0 bits of the ADSCM00 or ADSCM10 register.
Be sure to set a pin number that is smaller than the conversion termination analog input pin number set according to Note 2.

2. Set using the ANIS3 to ANIS0 of the ADSCM00 or ADSCM10 register.

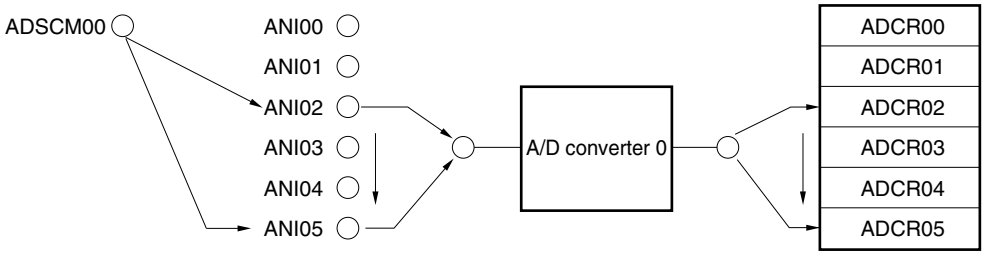
Remark x = 00 to 05, 10 to 17

It is not necessary to write 1 in the ADCE0 or ADCE1 bit of the ADSCM00 or ADSCM10 register as an A/D conversion restart operation in A/D trigger polling mode[Note].

This is optimal for applications that regularly read A/D conversion values.

Note In A/D trigger polling mode, the fact that the ADCE0 or ADCE1 bit of the ADSCM00 or ADSCM10 register is 0 means that A/D conversion operation does not stop as long as the ADCS0 or ADCS1 bit is not 0. Therefore, if the ADCR0m or ADCR1n register is not read before the next A/D conversion, it is overwritten.

Figure 11-9. Example of Scan Mode (A/D Trigger Polling Scan) Operation (ANI02 to ANI05): For A/D Converter 0



(1) ADCE0 bit of ADSCM00 = 1 (Enabled)
(2) A/D conversion of ANI02
(3) Store conversion result in ADCR02
(4) A/D conversion of ANI03
(5) Store conversion result in ADCR03
(6) A/D conversion of ANI04
(7) Store conversion result in ADCR04
(8) A/D conversion of ANI05
(9) Store conversion result in ADCR05
(10) Generate INTAD0 interrupt
(11) Return to (2)

11.9 Operation in Timer Trigger Mode

A/D converters 0 and 1 have a total of 14 channels of analog inputs (ANI00 to ANI05 and ANI10 to ANI17). For these channels, an interrupt signal specified by A/D internal trigger selection registers 0 and 1 (ITRG0, INTRG1) can be set as a conversion trigger.

The eight interrupt signals that can be selected as triggers are the TM0n timer 0 register underflow interrupt signals (INTTM00 and INTTM01) and the CM003 to CM005 and CM013 to CM015 match interrupt signals (INTCM003 to INTCM005 and INTCM013 to INTCM015) (n = 0, 1).

11.9.1 Operation in select mode

Taking the interrupt signal specified by A/D internal trigger selection registers 0 and 1 (ITRG0, ITRG1) as a trigger, one analog input (ANI00 to ANI05, ANI10 to ANI17) specified by the ADSCM00 or ADSCM10 register is A/D converted once. The conversion result is stored in the ADCR0m or ADCR1n register corresponding to the analog input (m = 0 to 5, n = 0 to 7). The A/D conversion end interrupt (INTAD0 or INTAD1) is generated at the end of each A/D conversion, which terminates A/D conversion (ADCS0, ADCS1 = 0).

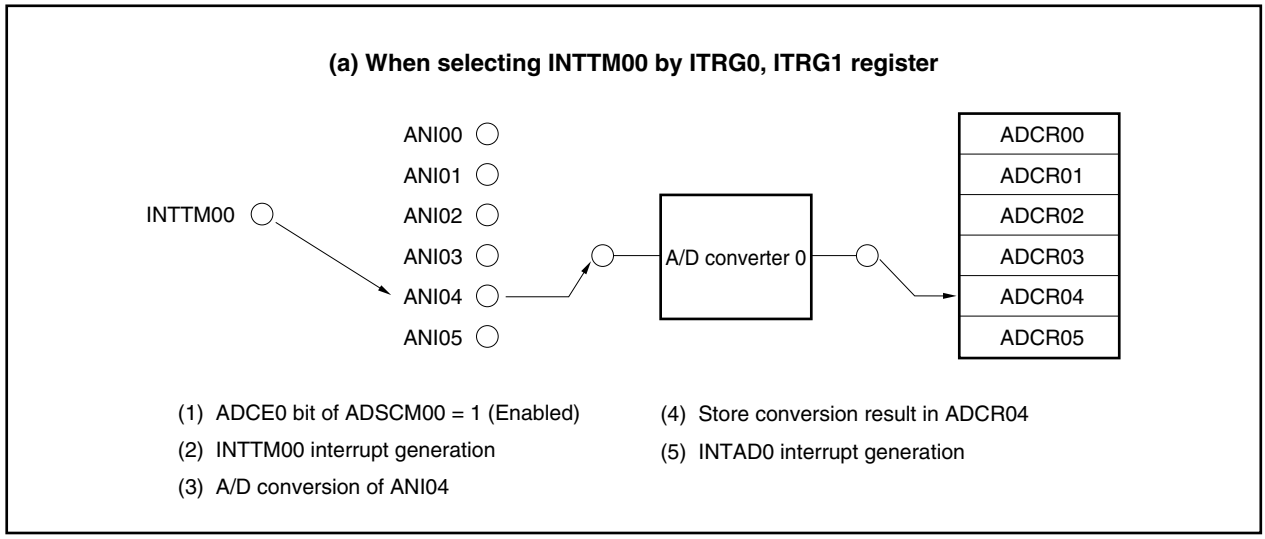
This is optimal for applications that read A/D conversion values synchronized to a timer trigger.

Trigger	Analog Input	A/D Conversion Result Register
Interrupt specified by ITRG0, ITRG1 register	ANIx	ADCRx

Remark n = 00 to 05, 10 to 17

After the end of A/D conversion, A/D converter 0 or 1 changes to the trigger wait status (ADCE0, ADCE1 = 1). A/D conversion is performed again when the interrupt signal specified by the ITRG0 or ITRG1register is generated.

Figure 11-10. Example of Timer Trigger Select Mode Operation (ANI04): For A/D Converter 0



11.9.2 Operation in scan mode

Using the interrupt signal specified by A/D internal trigger selection registers 0 and 1 (ITRG0, ITRG1) as a trigger, pins from the conversion start analog input pin to the conversion termination analog input pin specified by the ADSCM00 or ADSCM10 register are sequentially selected and A/D converted. Conversion results are stored in the ADCR0m or ADCR1n register corresponding to the analog input (m = 0 to 5, n = 0 to 7). When all of the specified A/D conversions are complete, the A/D conversion end interrupt (INTAD0 or INTAD1) is generated, which terminates A/D conversion (ADCS0, ADCS1 = 0).

This is optimal for applications that regularly monitor multiple analog inputs in synchronization with a timer trigger.

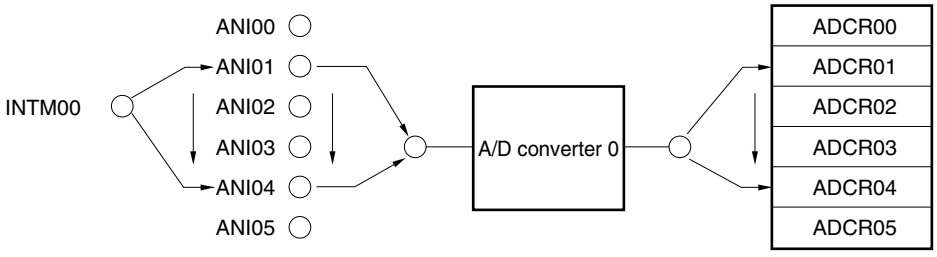
Trigger	Analog Input	A/D Conversion Result Register
Interrupt specified by ITRG0, ITRG1 register	ANIn0	ADCRn0
	ANIn1	ADCRn1
	ANIn2	ADCRn2
	ANIn3	ADCRn3
	ANIn4	ADCRn4
	ANIn5	ADCRn5
	ANI16	ADCR16
	ANI17	ADCR17

Remark n = 0, 1

After all of the specified A/D conversions have ended, the A/D converter changes to the trigger wait status (ADCE0, ADCE1 = 1). A/D conversion is performed again when the interrupt signal specified by the ITRG0 or ITRG1 register is generated.

Figure 11-11. Example of Timer Trigger Scan Mode Operation (for A/D Converter 0): INTTM00 Selected by ITRG0, ITRG1 Register

(a) Set to scan ANI01 to ANI04



(1) ADCE0 bit of ADSCM00 = 1 (Enabled)
(2) INTTM00 interrupt generation
(3) A/D conversion of ANI01
(4) Store conversion result in ADCR01
(5) A/D conversion of ANI02
(6) Store conversion result in ADCR02
(7) A/D conversion of ANI03
(8) Store conversion result in ADCR03
(9) A/D conversion of ANI04
(10) Store conversion result in ADCR04
(11) INTAD0 interrupt generation

11.10 Operation in External Trigger Mode

In external trigger mode, an analog input (ANI00 to ANI05, ANI10 to ANI17) is A/D converted at the ADTRG0 or ADTRG1 pin input timing.

The valid edge of an external input signal in external trigger mode can be specified as the rising edge, falling edge, or both rising and falling edges using the ES21 or ES20 bit of the INTM1 register for A/D converter 0 and the ES31 or ES30 bit of the INTM1 register for A/D converter 1.

11.10.1 Operation in select mode

One analog input (ANI00 to ANI05, ANI10 to ANI17) specified by the ADSCM00 or ADSCM10 register is A/D converted. The conversion result is stored in the ADCR0m or ADCR1n register (m = 0 to 5, n = 0 to 7).

Using the ADTRG0 or ADTRG1 signal as a trigger, one analog input is A/D converted at a time and the result is stored in the ADCR0m or ADCR1n register. Analog inputs correspond one-to-one with A/D conversion result registers. For each A/D conversion, an A/D conversion end interrupt (INTAD0 or INTAD1) is generated, which terminates A/D conversion (ADCS0, ADCS1 bit = 0).

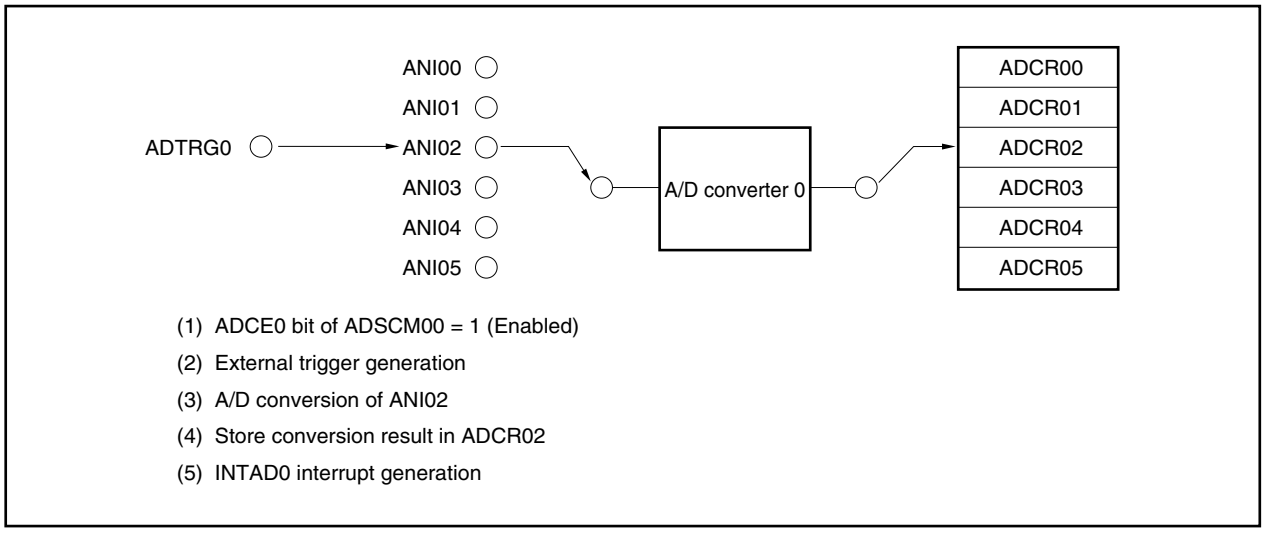
Trigger	Analog Input	A/D Conversion Result Register
ADTRGm signal	ANImn	ADCRmn

Remark m = 0, 1
n: 0 to 5 when m = 0, or 0 to 7 when m = 1

To restart A/D conversion, a trigger must be input again from the ADTRGn pin (n = 0, 1).

This is optimal for applications that read results each time there is an A/D conversion in synchronization with an external trigger.

Figure 11-12. Example of Select Mode (External Trigger Select) Operation (ANI02): For A/D Converter 0



11.10.2 Operation in scan mode

Using the ADTRG0 or ADTRG1 signal as a trigger, pins from the conversion start analog input pin to the conversion termination analog input pin specified by the ADSCM00 or ADSCM10 register are sequentially selected and A/D converted. A/D conversion results are stored in the ADCR0m or ADCRN1n register corresponding to the analog input (m = 0 to 5, n = 0 to 7). When conversion ends for all of the specified analog inputs, an INTAD0 or INTAD1 interrupt is generated, which terminates A/D conversion (ADCS0, ADCS1 = 0).

Trigger	Analog Input	A/D Conversion Result Register
ADTRGn signal	ANIn0	ADCRn0
	ANIn1	ADCRn1
	ANIn2	ADCRn2
	ANIn3	ADCRn3
	ANIn4	ADCRn4
	ANIn5	ADCRn5
	ANI16	ADCR16
	ANI17	ADCR17

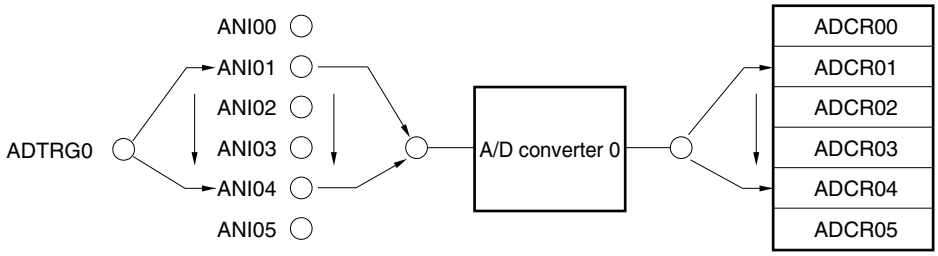
Remark n = 0, 1

After all specified A/D conversions have ended, A/D conversion is restarted when an external trigger signal occurs.

This is optimal for applications that regularly monitor multiple analog inputs in synchronization with an external trigger.

Figure 11-13. Example of Scan Mode (External Trigger Scan) Operation: For A/D Converter 0

(a) When setting to scan ANI01 to ANI04



(1) ADCE0 bit of ADSCM00 = 1 (Enabled)
(2) External trigger generation
(3) A/D conversion of ANI01
(4) Store conversion result in ADCR01
(5) A/D conversion of ANI02
(6) Store conversion result in ADCR02
(7) A/D conversion of ANI03
(8) Store conversion result in ADCR03
(9) A/D conversion of ANI04
(10) Store conversion result in ADCR04
(11) INTAD0 interrupt generation

11.11 Operation Cautions

11.11.1 Stopping A/D conversion operation

If 0 is written in the ADCE0 or ADCE1 bit of the ADSCM00 or ADSCM10 register during A/D conversion, it stops the A/D conversion operation and an A/D conversion result is not stored in the ADCR0m or ADCR1n register (m = 0 to 5, n = 0 to 7).

11.11.2 Trigger input during A/D conversion operation

If a trigger is input during A/D conversion, that trigger input is ignored.

11.11.3 External or timer trigger interval

Make the trigger interval (input time interval) in external or timer trigger mode longer than the conversion time specified by the FR2 to FR0 bits of the ADSCM01 or ADSCM11 register.

(1) When interval = 0

If multiple triggers are input simultaneously, processing is performed assuming that they are one trigger signal.

(2) When 0 < interval < conversion time

If an external or timer trigger is input during A/D conversion, that trigger input is ignored.

(3) When interval = conversion time

If an external or timer trigger is input at the same time as the end of A/D conversion (conflict of compare termination signal and trigger), interrupt generation and storage of the value at which conversion ended in the ADCR0m or ADCR1n register is performed correctly (m = 0 to 5, n = 0 to 7).

11.11.4 Operation in standby modes

(1) HALT mode

A/D conversion is suspended. If released by NMI or maskable interrupt input, the ADSCM00, ADSCM10, ADSCM01, or ADSCM11 register and ADCR0m or ADCR1n register maintain their values (m = 0 to 5, n = 0 to 7).

If released by $\overline{\text{RESET}}$ input, the ADCR0m and ADCR1n registers are initialized.

(2) IDLE mode, software STOP mode

Since clock provision to A/D converter 0 or 1 stops, A/D conversion is not performed.

If released by NMI or maskable interrupt input, the ADSCM00, ADSCM10, ADSCM01, or ADSCM11 register and ADCR0m or ADCR1n register maintain their values (m = 0 to 5, n = 0 to 7). However, if IDLE mode or software STOP mode is set during an A/D conversion operation, the A/D conversion operation stops. If released by $\overline{\text{RESET}}$ input, the ADCR0m and ADCR1n registers are initialized.

11.11.5 Compare match interrupt in timer trigger mode

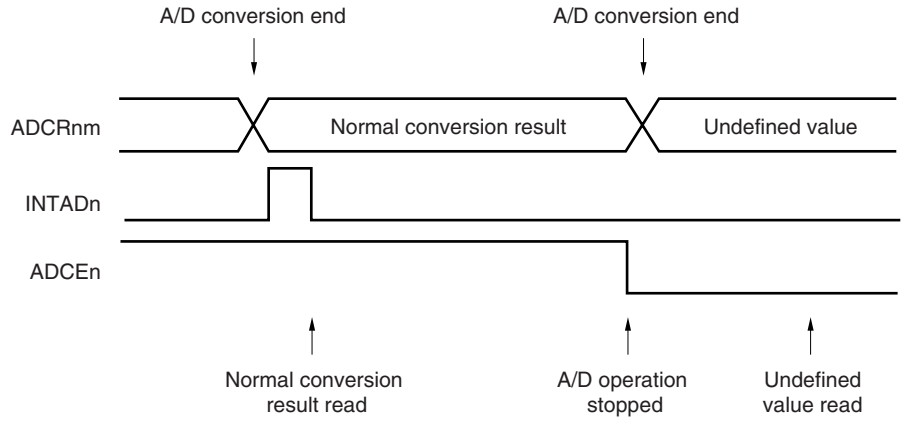
The TM0n timer 0 register underflow interrupt (INTTM00 or INTTM01) and CM003 to CM005 or CM013 to CM015 match interrupt (INTCM003 to INTCM005 or INTCM013 to INTCM015) are A/D conversion start triggers that start a conversion operation (n = 0,1). At this time, the CM003 to CM005 or CM013 to CM015 match interrupt (INTCM003 to INTCM005 or INTCM013 to INTCM015) also functions as a compare register match interrupt for the CPU. In order not to generate these match interrupts for the CPU, disable interrupts using the mask bits (TM0MK0, TM0MK1, CM03MK0 to CM05MK0, CM03MK1 to CM05MK1) of the interrupt control registers (TM0IC0, TM0IC1, CM03IC0 to CM05IC0, CM03IC1 to CM05IC1).

11.11.6 Timing that makes the A/D conversion result undefined

If the timing of the end of A/D conversion and the timing of the stop of operation of the A/D converter conflict, the A/D conversion value may be undefined. Because of this, be sure to read the A/D conversion result while the A/D converter is in operation. Furthermore, when reading an A/D conversion result after the A/D converter operation has stopped, be sure to have done so by the time the next conversion result is complete.

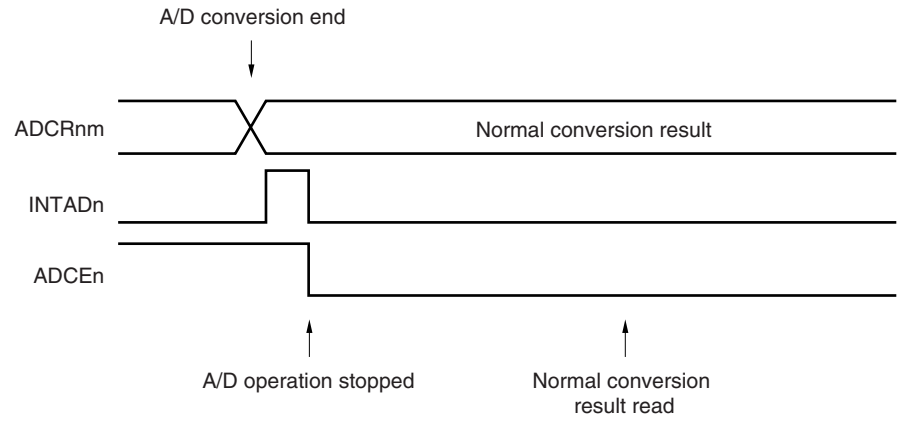
The conversion result read timing is shown in Figures 11-14 and 11-15 below.

Figure 11-14. Conversion Result Read Timing (When Conversion Result Is Undefined)



Remark n = 0, 1
When n = 0: m = 0 to 5
When n = 1: m = 0 to 7

Figure 11-15. Conversion Result Read Timing (When Conversion Result Is Normal)



Remark n = 0, 1
When n = 0: m = 0 to 5
When n = 1: m = 0 to 7

11.12 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range). %FSR indicates the ratio of analog input voltage that can be converted as a percentage, and is always represented by the following formula regardless of the resolution.

1%FSR = (Max. value of analog input voltage that can be converted − Min. value of analog input voltage that can be converted)/100
= (AV_{DDn} − 0)/100
= AV_{DDn}/100

Remark n = 0, 1

1LSB is as follows when the resolution is 10 bits.

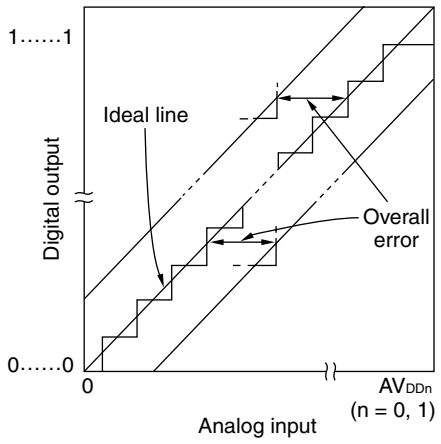
$1LSB = 1/2^{10} = 1/1024$
= 0.098%FSR

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.
Zero-scale error, full-scale error, linearity error, and errors that are combinations of these express the overall error.
Note that the quantization error is not included in the overall error in the characteristics table.

Figure 11-16. Overall Error

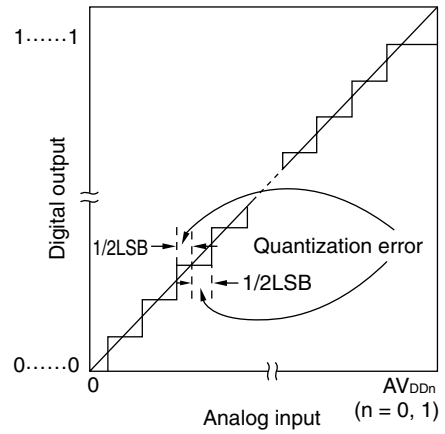


(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.

Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

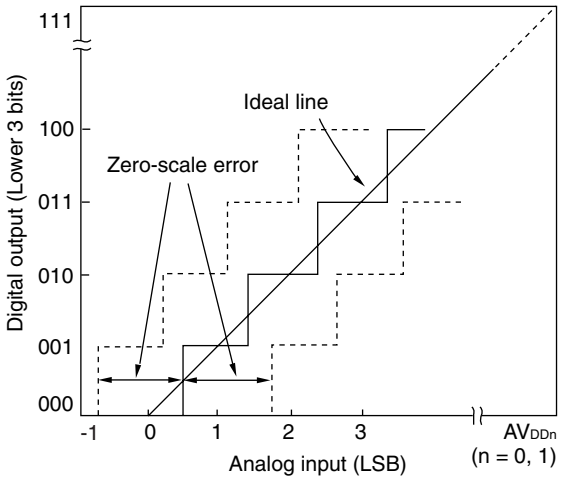
Figure 11-17. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2 LSB) when the digital output changes from 0……000 to 0……001.

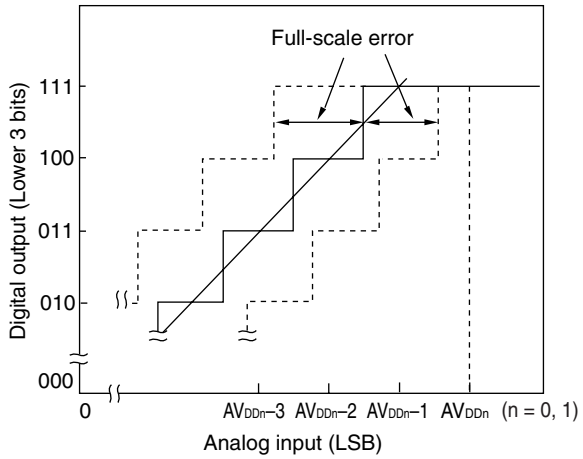
Figure 11-18. Zero-Scale Error



(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 1……110 to 1……111.

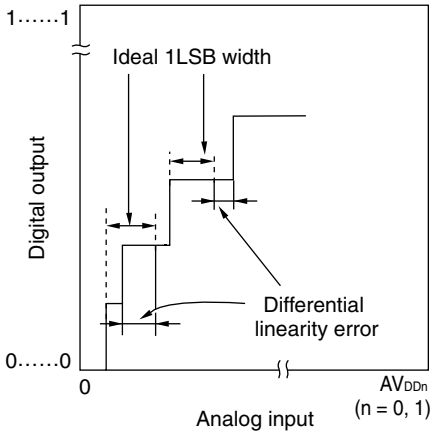
Figure 11-19. Full-Scale Error



(6) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

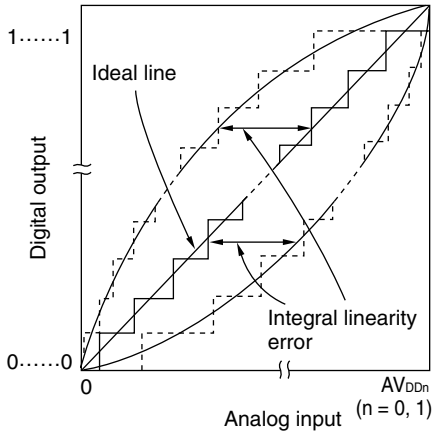
Figure 11-20. Differential Linearity Error



(7) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

Figure 11-21. Integral Linearity Error



(8) Conversion time

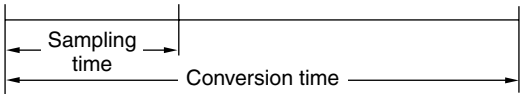
This expresses the time from when each trigger was generated to the time when the digital output was obtained.

The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.

Figure 11-22. Sampling Time



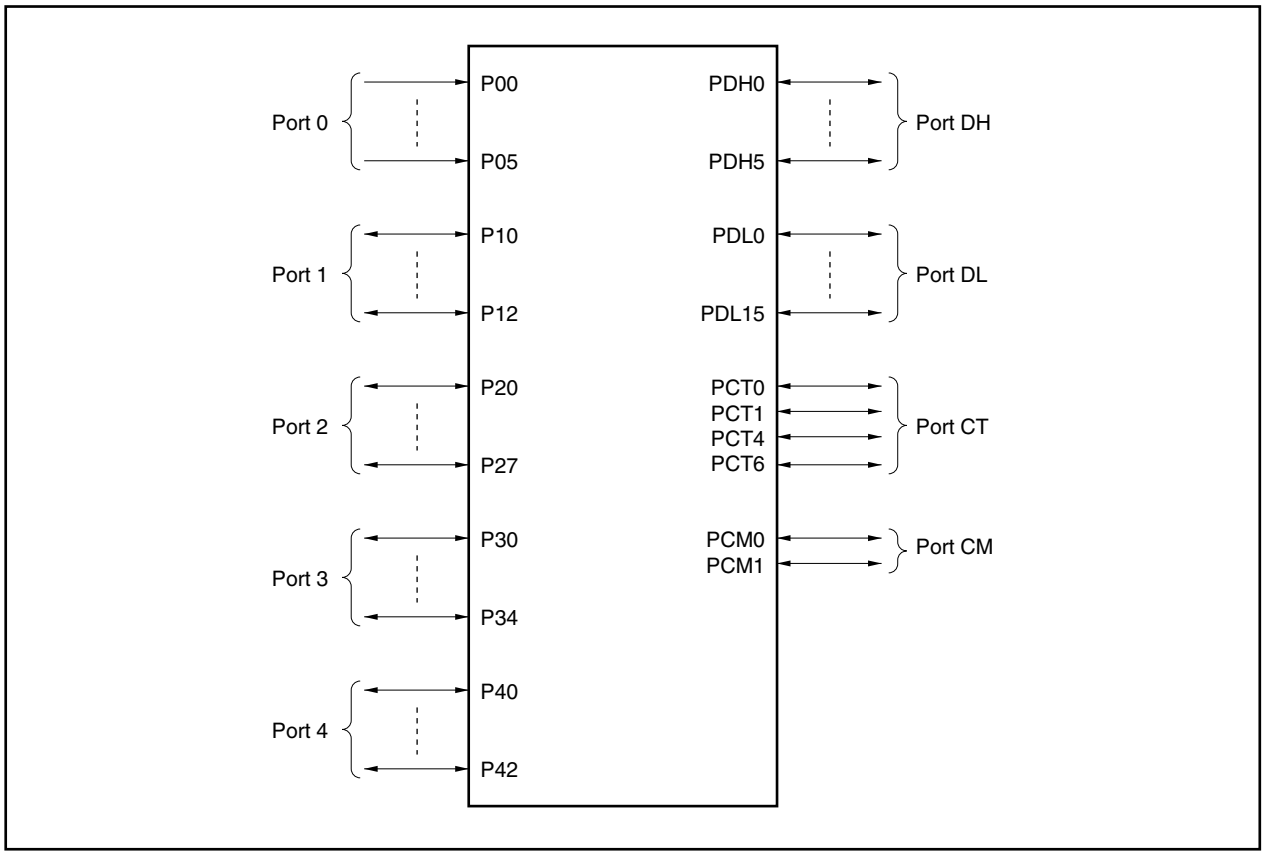
CHAPTER 12 PORT FUNCTIONS

12.1 Features

- Input-only ports: 6
  I/O ports: 47
- Ports function alternately as I/O pins of other peripheral functions
- Input or output can be specified in bit units

12.2 Basic Configuration of Ports

The V850E/IA2 has a total of 53 on-chip I/O ports (ports 0 to 4, DH, DL, CT, CM), of which 6 are input-only ports. The port configuration is shown below.



(1) Functions of each port

The V850E/IA2 has the ports shown below.

Any port can operate in 8-bit or 1-bit units and can provide a variety of controls.

Moreover, besides its function as a port, each has functions as the I/O pins of on-chip peripheral I/O in control mode.

Refer to **(3) Port block diagrams** for a block diagram of the block type of each port.

Port Name	Pin Name	Port Function	Function in Control Mode	Block Type
Port 0	P00 to P05	6-bit input	NMI input Timer/counter output stop signal input External interrupt input A/D converter (ADC) external trigger input Timer 3 output stop signal input	E
Port 1	P10 to P12	3-bit I/O	Timer/counter I/O External interrupt input	B, K
Port 2	P20 to P27	8-bit I/O	Timer/counter I/O External interrupt input	B, K, L
Port 3	P30 to P34	5-bit I/O	Serial interface I/O (UART0, UART1/CSI1)	A, C, F, G, H
Port 4	P40 to P42	3-bit I/O	Serial interface I/O (CSI0)	A, C, J
Port DH	PDH0 to PDH5	6-bit I/O	External address bus (A16 to A21)	N
Port DL	PDL0 to PDL15	16-bit I/O	External address/data bus (AD0 to AD15)	M
Port CT	PCT0 PCT1, PCT4, PCT6	4-bit I/O	External bus interface control signal output	I
Port CM	PCM0, PCM1	2-bit I/O	Wait insertion signal input Internal system clock output	D, I

Cautions 1. When switching to the control mode, be sure to set ports that operate as output pins or I/O pins in the control mode using the following procedure.

<1> Set the inactive level for the signal output in the control mode in the corresponding bits of port n (n = 0 to 4, CM, CS, CT, DH, and DL).

<2> Switch to the control mode using the port n mode control register (PMCn).

If <1> above is not performed, the contents of port n may be output for a moment when switching from the port mode to the control mode.

2. When port manipulation is performed by a bit manipulation instruction (SET1, CLR1, or NOT1), perform byte data read for the port and process the data of only the bits to be manipulated, and write the byte data after conversion back to the port.
For example, in ports in which input and output are mixed, because the contents of the output latch are overwritten to bits other than the bits for manipulation, the output latch of the input pin becomes undefined (in the input mode, however, the pin status does not change because the output buffer is off).
Therefore, when switching the port from input to output, set the output expected value to the corresponding bit, and then switch to the output port. This is the same as when the control mode and output port are mixed.

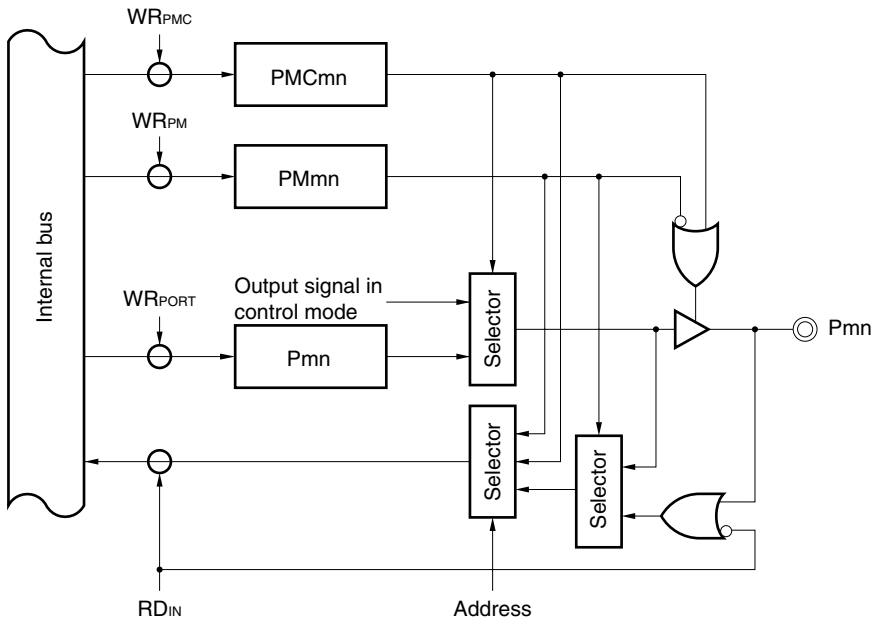
3. The state of the port pin can be read by setting the port n mode register (PMn) to the input mode regardless of the settings of the PMCn register. When the PMn register is set to the output mode, the value of the port n register (Pn) can be read in the port mode while the output state of the alternate function can be read in the control mode.

(2) Functions of each port pin after reset and registers that set port or control mode

Port Name	Pin Name	Pin Function After Reset		Mode-Setting Register
		Single-Chip Mode	ROMless Mode	
Port 0	P00/NMI	P00 (input mode)		–
	P01/ESO0/INTP0	P01 (input mode)		
	P02/ESO1/INTP1	P02 (input mode)		
	P03/ADTRG0/INTP2	P03 (input mode)		
	P04/ADTRG1/INTP3	P04 (input mode)		
	P05/INTP4/TO3OFF	P05 (input mode)		
Port 1	P10/TIUD10/TO10	P10 (input mode)		PMC1, PFC1
	P11/TCUD10/INTP100	P11 (input mode)		PMC1
	P12/TCLR10/INTP101	P12 (input mode)		
Port 2	P20/TI2/INTP20	P20 (input mode)		PMC2
	P21/TO21/INTP21	P21 (input mode)		PMC2, PFC2
	P22/TO22/INTP22	P22 (input mode)		
	P23/TO23/INTP23	P23 (input mode)		
	P24/TO24/INTP24	P24 (input mode)		
	P25/TCLR2/INTP25	P25 (input mode)		PMC2
	P26/TI3/TCLR3/INTP30	P26 (input mode)		
	P27/TO3/INTP31	P27 (input mode)		PMC2, PFC2
Port 3	P30/RXD0	P30 (input mode)		PMC3
	P31/TXD0	P31 (input mode)		
	P32/RXD1/SI1	P32 (input mode)		
	P33/TXD1/SO1	P33 (input mode)		
	P34/$\overline{\text{ASCK1}}$/$\overline{\text{SCK1}}$	P34 (input mode)		
Port 4	P40/SI0	P40 (input mode)		PMC4
	P41/SO0	P41 (input mode)		
	P42/$\overline{\text{SCK0}}$	P42 (input mode)		
Port CM	PCM0/$\overline{\text{WAIT}}$	PCM0 (input mode)	$\overline{\text{WAIT}}$	PMCCM
	PCM1/CLKOUT	PCM1 (input mode)	CLKOUT	
Port CT	PCT0/$\overline{\text{LWR}}$	PCT0 (input mode)	$\overline{\text{LWR}}$	PMCCT
	PCT1/$\overline{\text{UWR}}$	PCT1 (input mode)	$\overline{\text{LWR}}$	
	PCT4/$\overline{\text{RD}}$	PCT4 (input mode)	$\overline{\text{RD}}$	PMCCT
	PCT6/ASTB	PCT6 (input mode)	ASTB	PMCCT
Port DH	PDH0/A16 to PDH5/A21	PDH0 to PDH5 (input mode)	A16 to A21	PMCDH
Port DL	PDL0/AD0 to PDL15/AD15	PDL0 to PDL7 (input mode)	AD0 to AD15	PMCDL

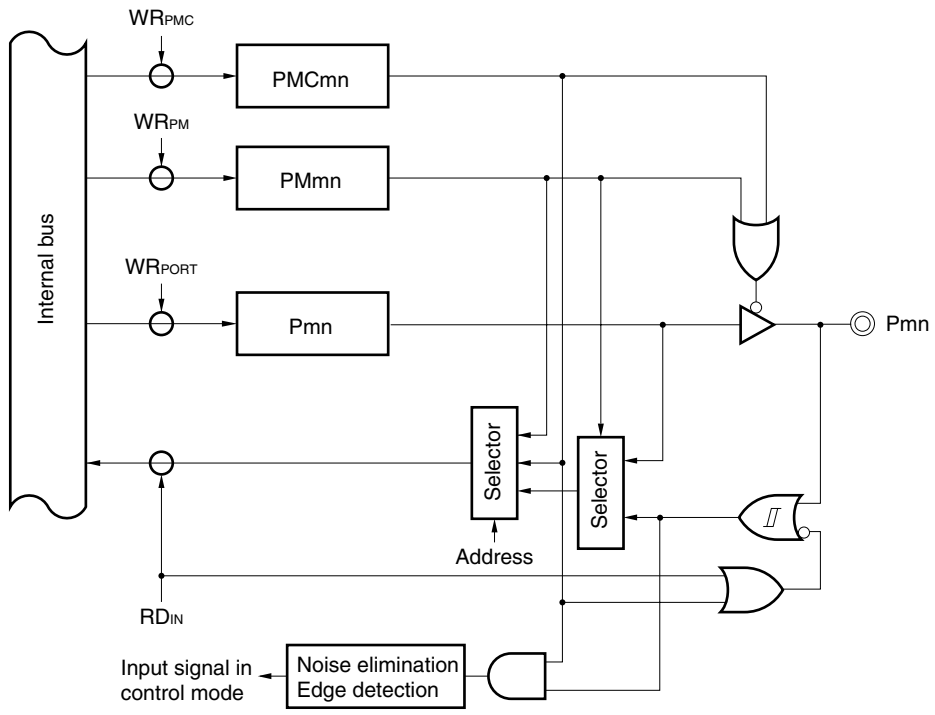
(3) Port block diagrams

Figure 12-1. Type A Block Diagram



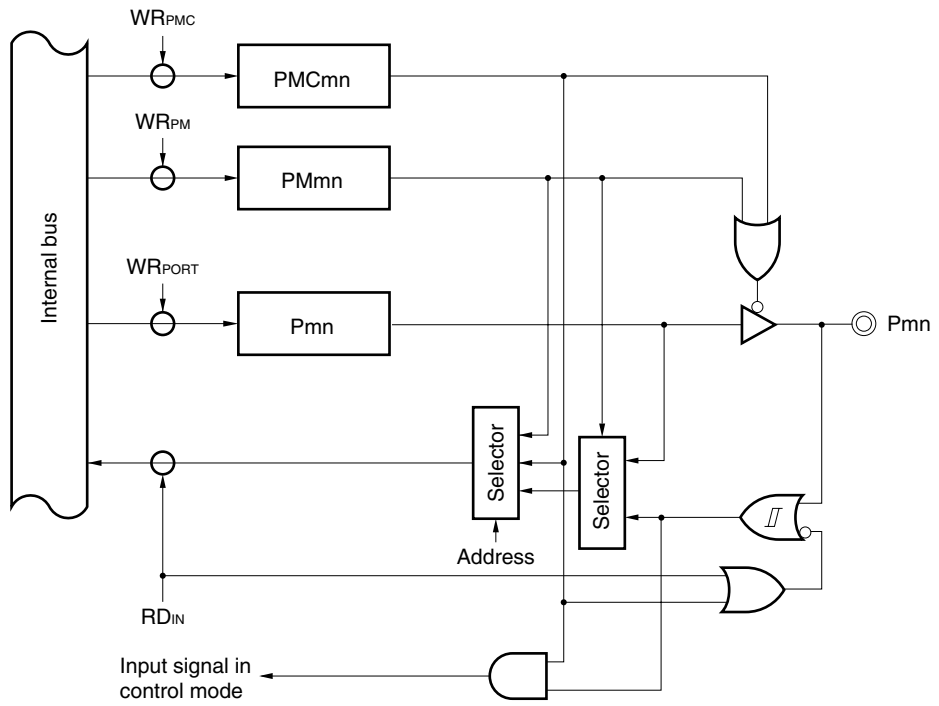
Remark m: Port number
n: Bit number

Figure 12-2. Type B Block Diagram



Remark m: Port number
n: Bit number

Figure 12-3. Type C Block Diagram



Remark m: Port number
n: Bit number

Figure 12-4. Type D Block Diagram

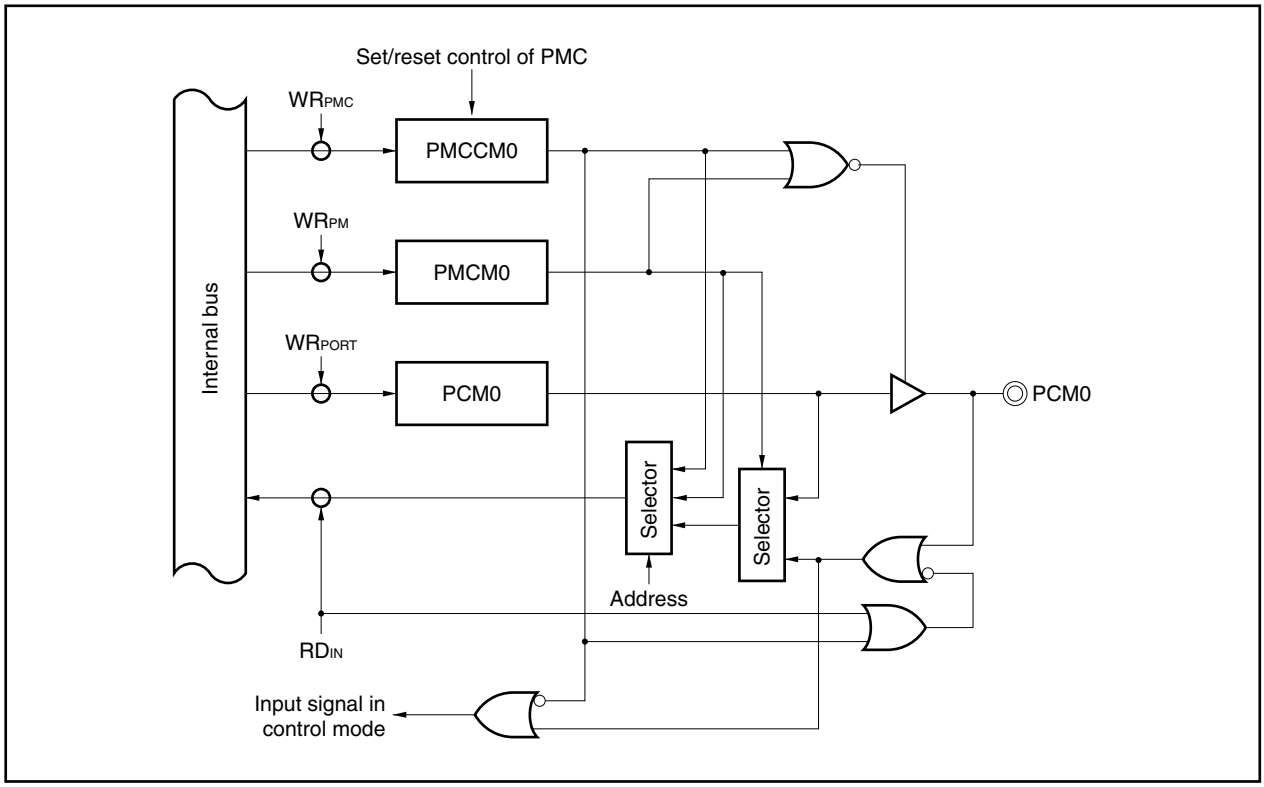
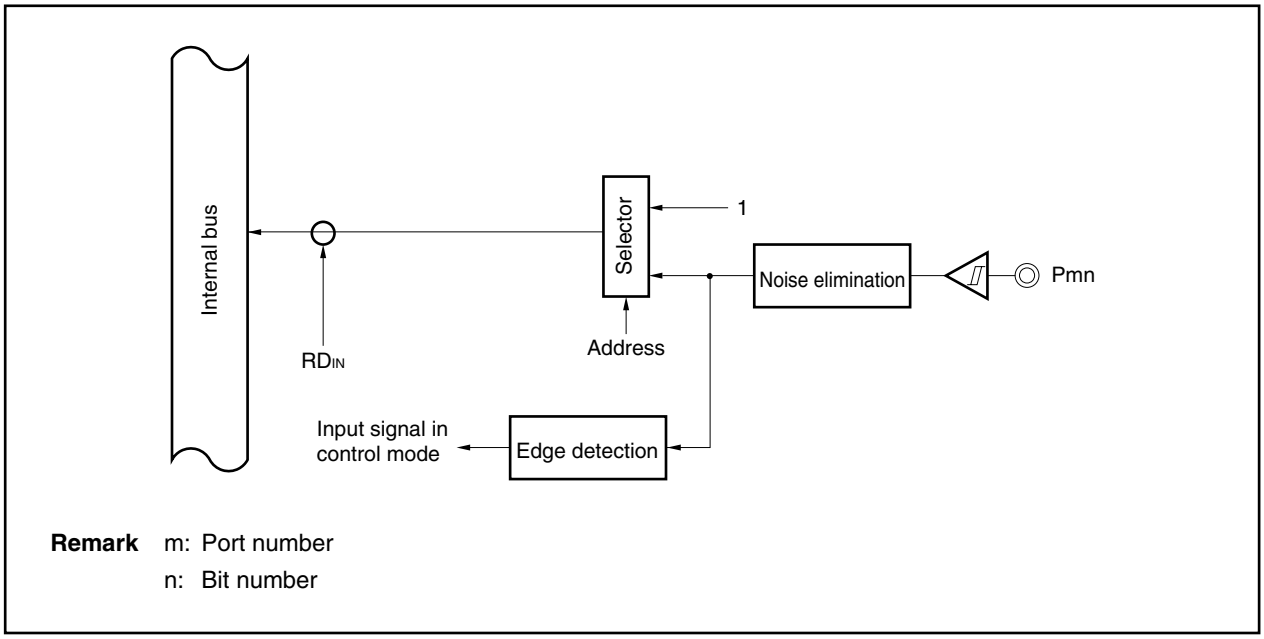


Figure 12-5. Type E Block Diagram



Remark m: Port number
n: Bit number

Figure 12-6. Type F Block Diagram

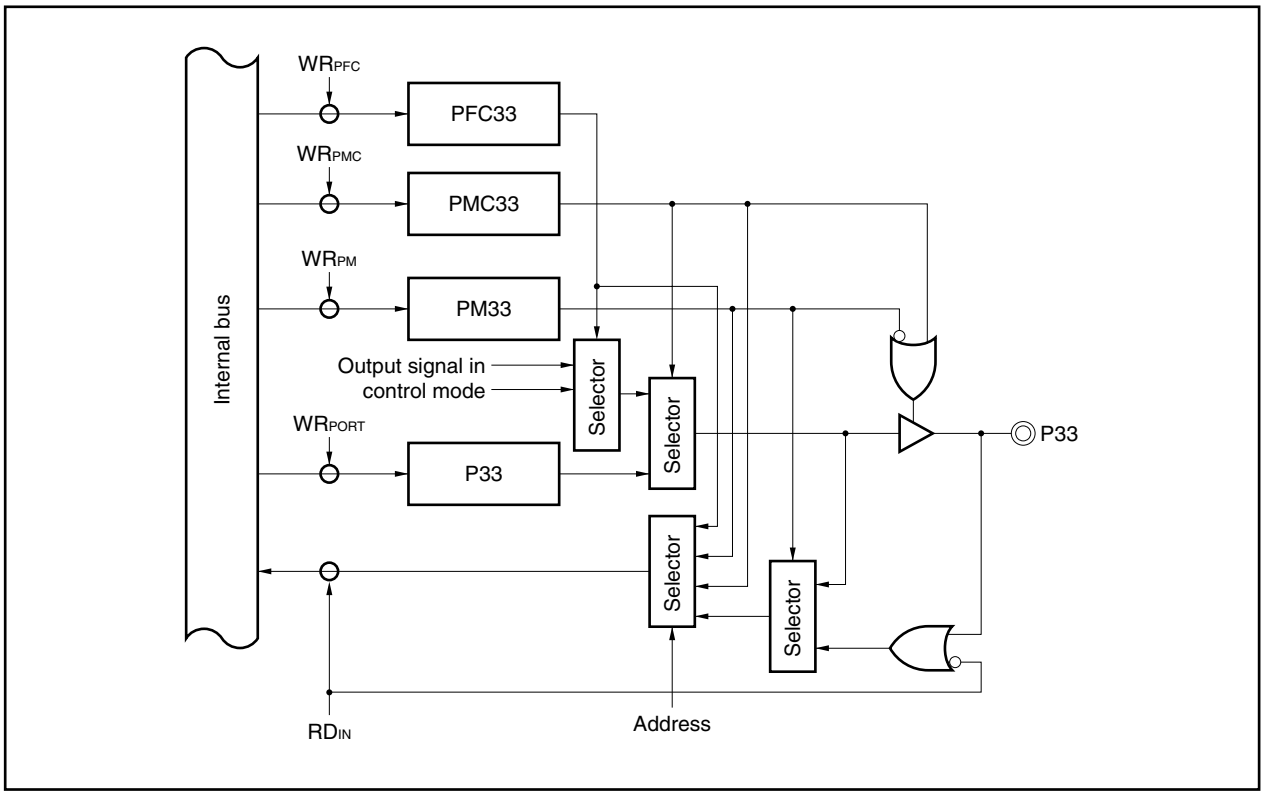
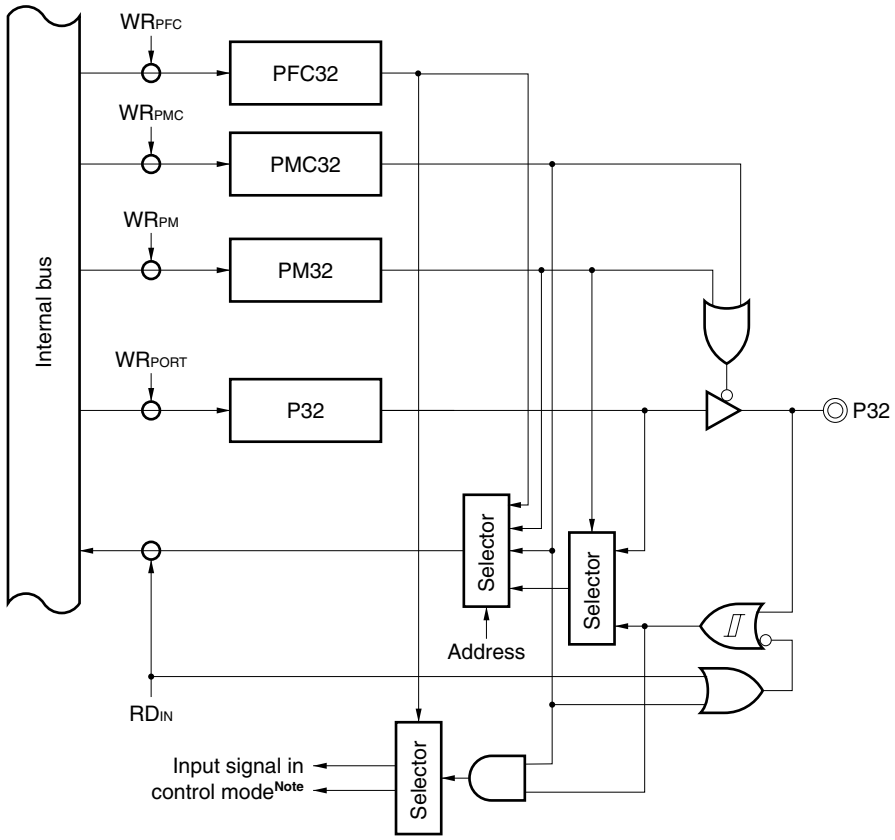


Figure 12-7. Type G Block Diagram

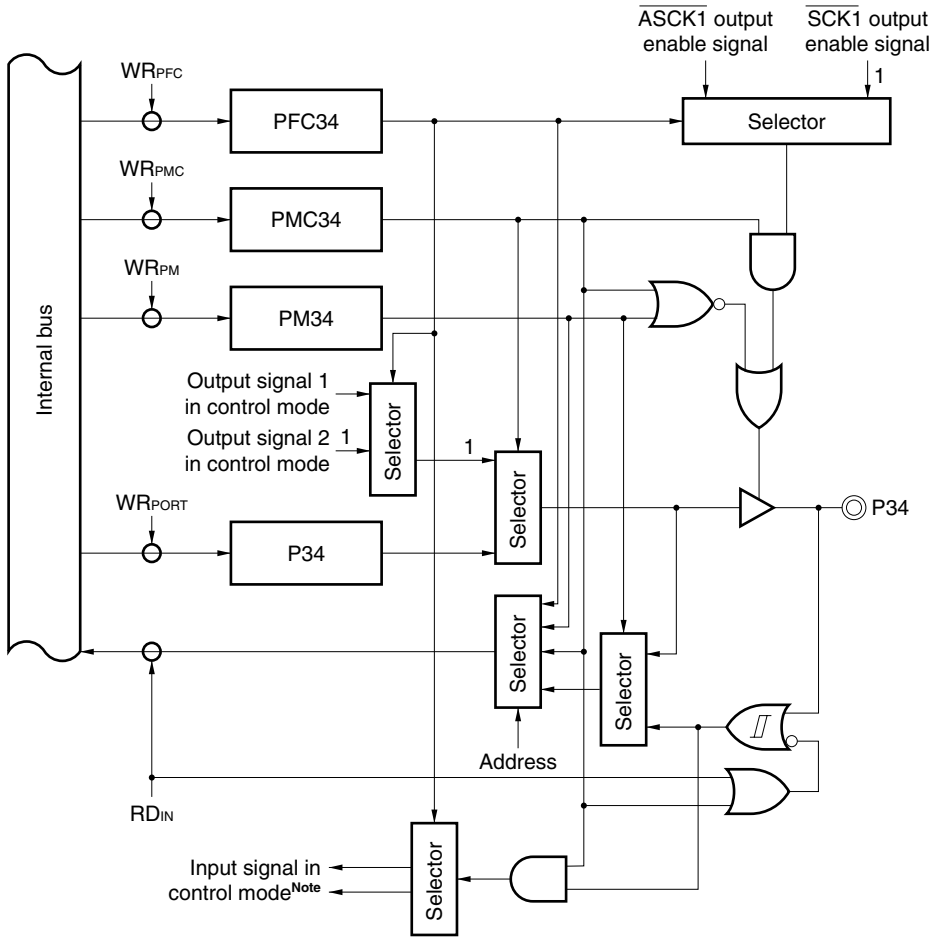


Note The signal level of the input signal is as follows in control mode.

PMC32 bit (PMC3 register)	PFC32 bit (PFC3 register)	Input signal in control mode	
		RXD1	SI1
0	×	H	L
1	0	Pin level	L
1	1	H	Pin level

H: High level
L: Low level
×: Don't care

Figure 12-8. Type H Block Diagram

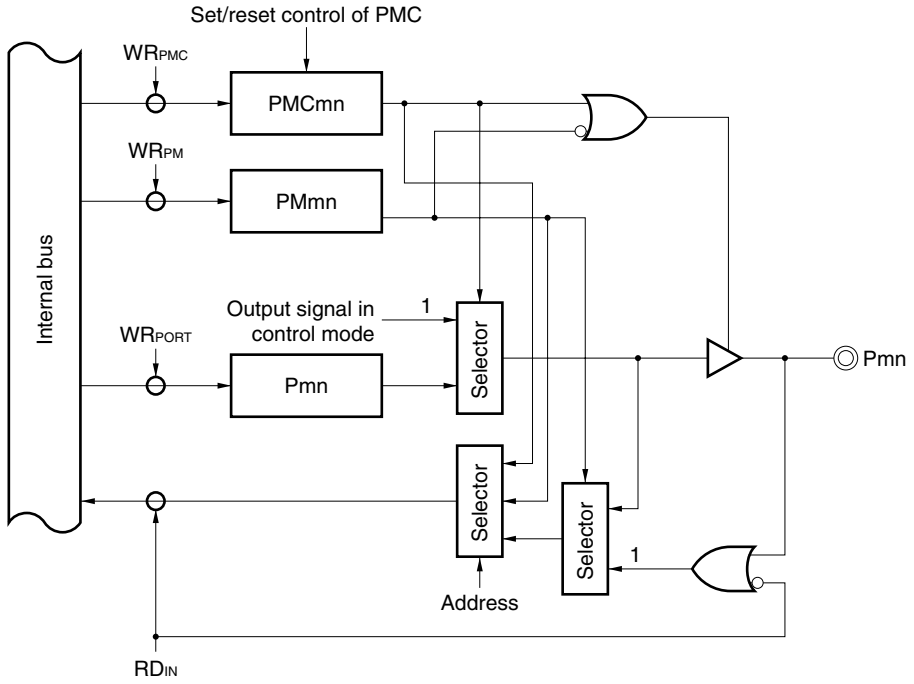


Note The signal level of the input signal is as follows in control mode.

PMC34 bit (PMC3 register)	PFC34 bit (PFC3 register)	Input signal in control mode	
		ASCK1	SCK1
0	×	L	L
1	0	Pin level	L
1	1	L	Pin level

H: High level
L: Low level
×: Don't care

Figure 12-9. Type I Block Diagram



Remark m: Port number
n: Bit number

Figure 12-10. Type J Block Diagram

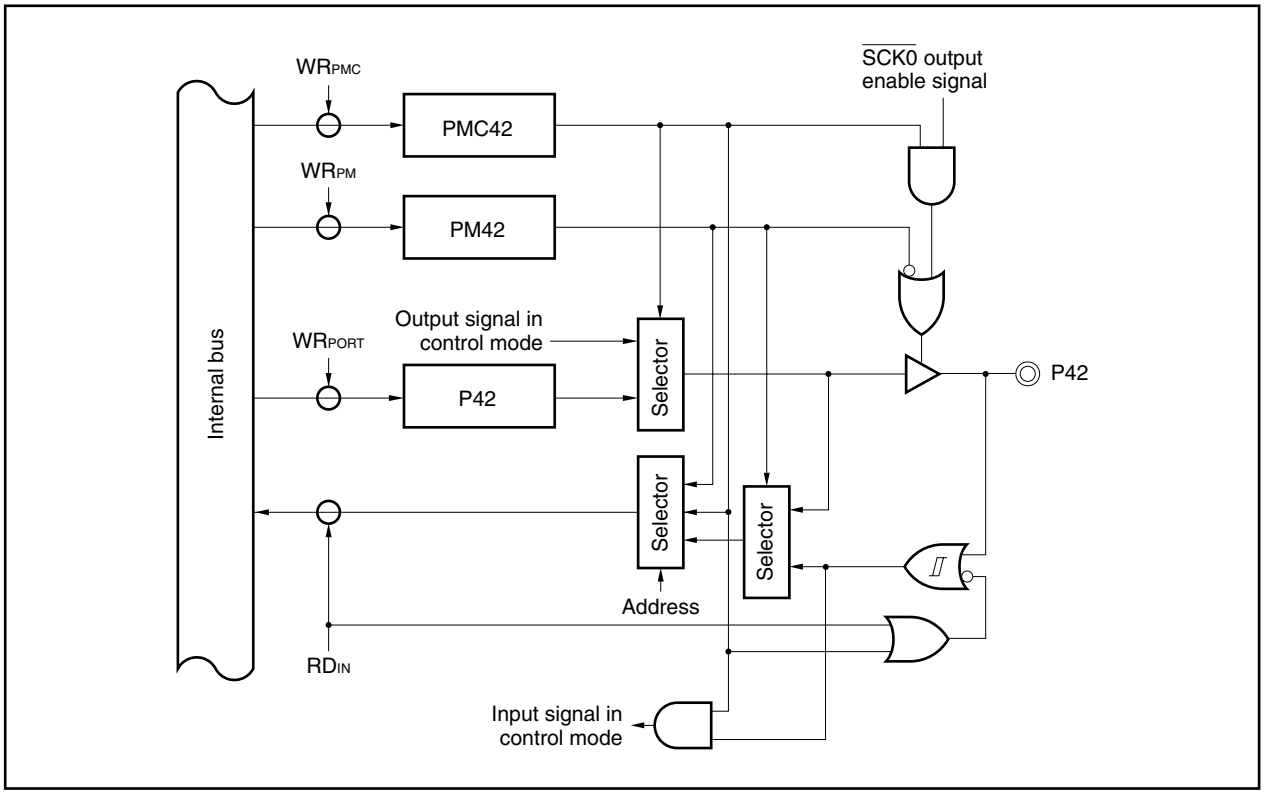
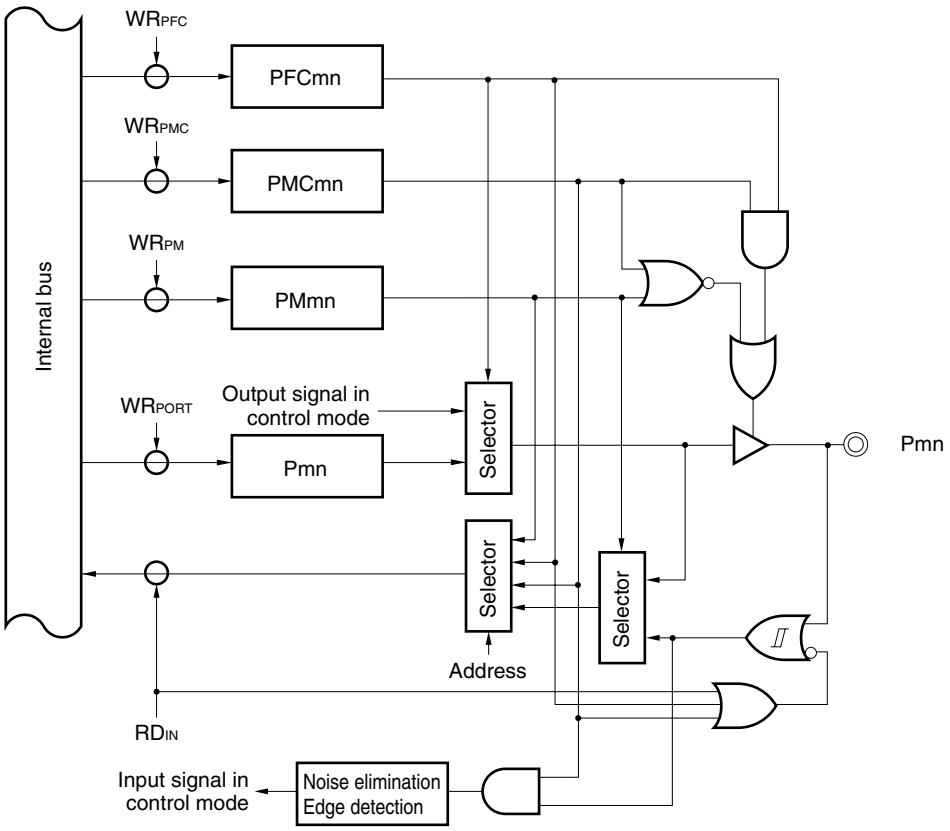
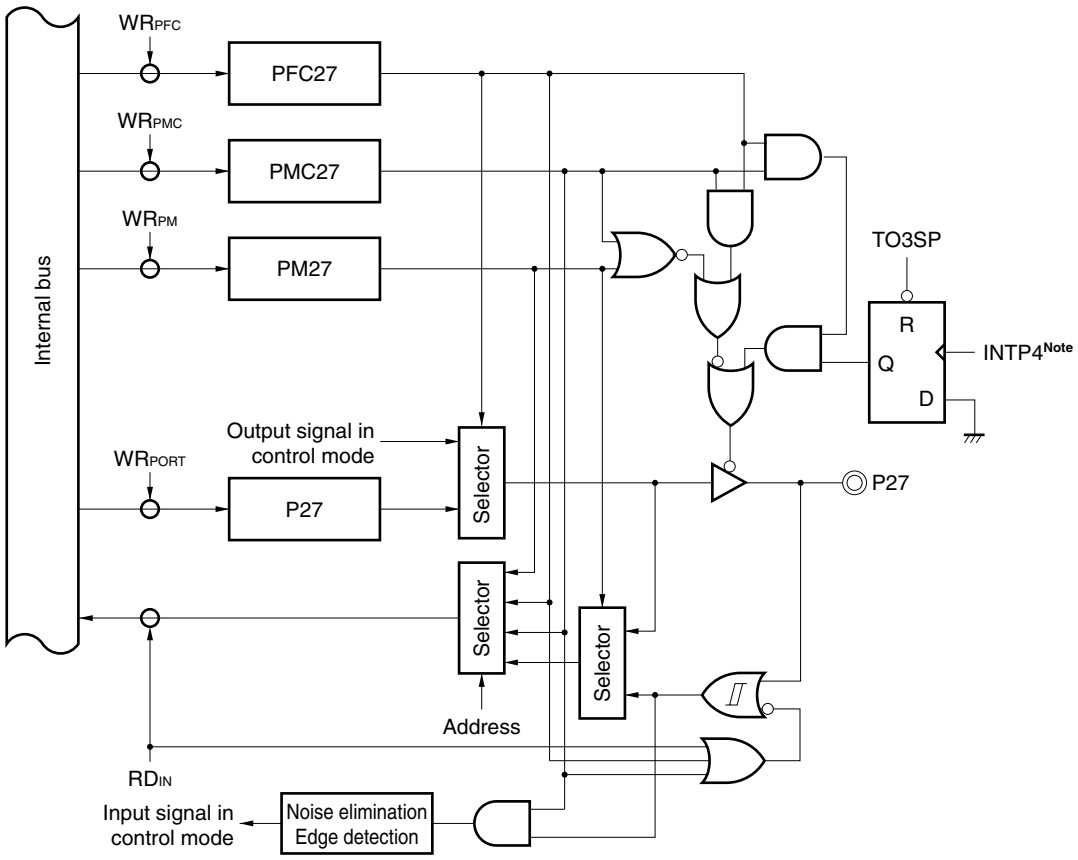


Figure 12-11. Type K Block Diagram



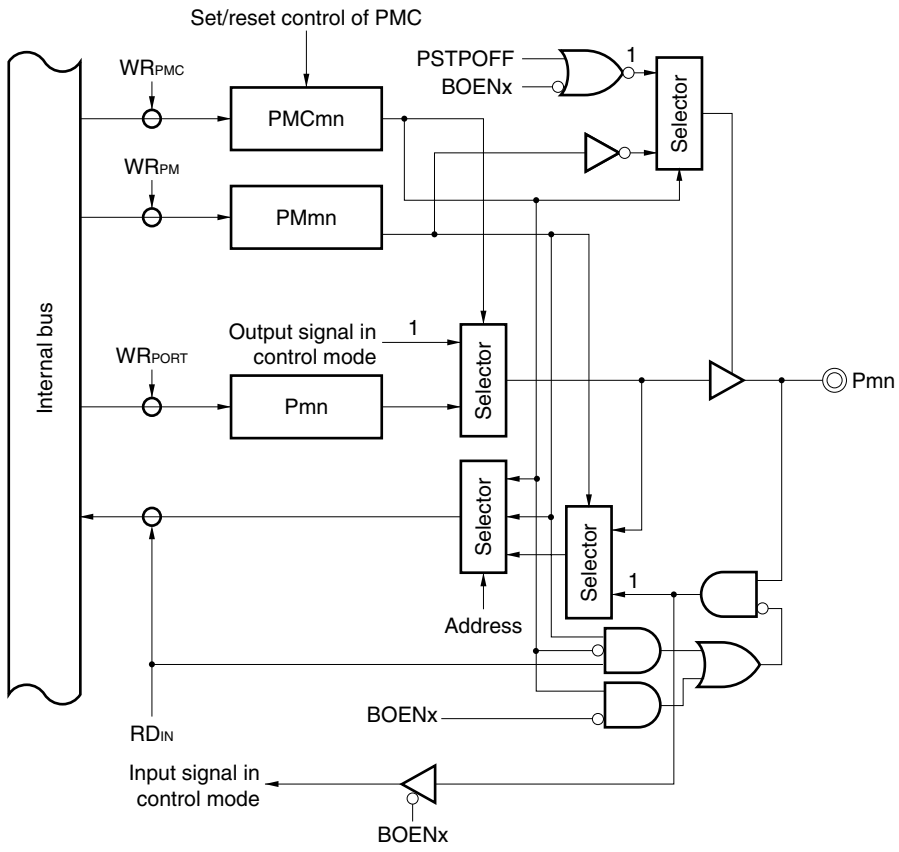
Remark m: Port number
n: Bit number

Figure 12-12. Type L Block Diagram



Note Output signal after an edge on the INTP4 pin has been detected.

Figure 12-13. Type M Block Diagram

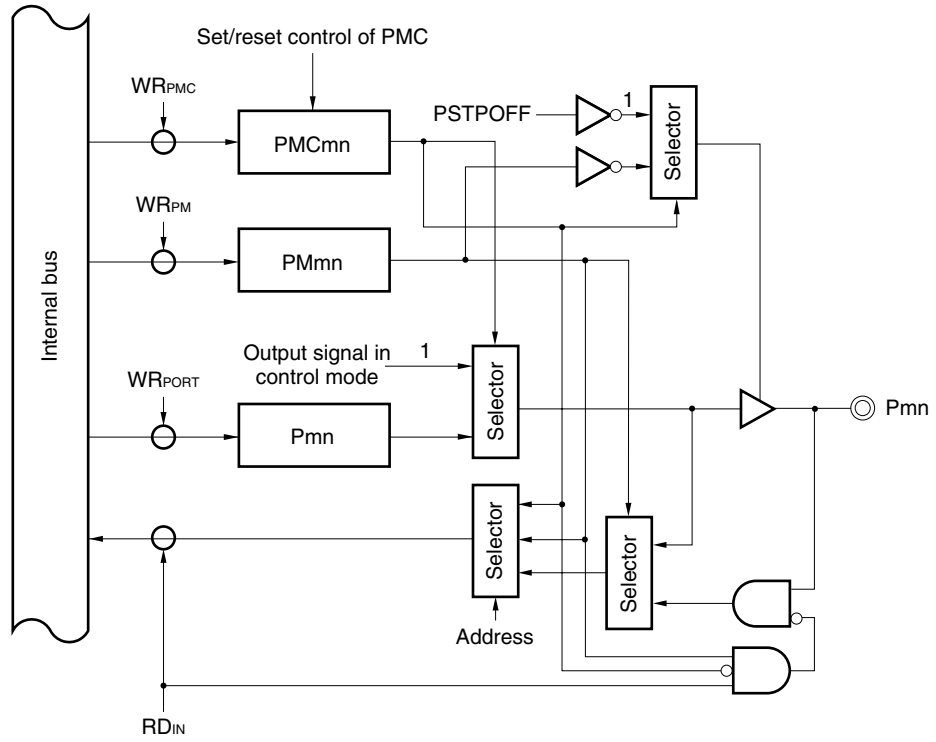


Remarks 1. m: Port number
n: Bit number

2. x = 0, 1

3. PSTPOFF: Signal in IDLE/software STOP mode
BOENx: A/D output signal

Figure 12-14. Type N Block Diagram



Remarks 1. m: Port number
n: Bit number
2. PSTPOFF: Signal in IDLE/software STOP mode

12.3 Pin Functions of Each Port

12.3.1 Port 0

Port 0 is a 6-bit input-only port in which all pins are fixed to input.

	7	6	5	4	3	2	1	0	Address	After reset
P0	–	–	P05	P04	P03	P02	P01	P00	FFFFF400H	Undefined

Besides functioning as an input port, in control mode, it can also operate as the timer/counter output stop signal input, external interrupt request input, A/D converter (ADC) external trigger input, and timer 3 output stop signal input.

Although this port is also used as NMI, ESO0/INTP0, ESO1/INTP1, ADTRG0/INTP2, ADTRG1/INTP3, and INTP4/TO3OFF, these functions cannot be switched with input port functions. The status of each pin is read by reading the port.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port 0	P00	NMI	Non-maskable interrupt request input	E
	P01	ESO0/INTP0	Timer/counter output stop signal input or external interrupt request input	
	P02	ESO1/INTP1		
	P03	ADTRG0/INTP2	A/D converter (ADC) external trigger input or external interrupt request input	
	P04	ADTRG1/INTP3		
	P05	INTP4/TO3OFF	External interrupt request input/timer 3 output stop signal input	

12.3.2 Port 1

Port 1 is a 3-bit I/O port in which input or output can be specified in 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
P1	–	–	–	–	–	P12	P11	P10	FFFFF402H	Undefined

Bit position	Bit name	Function
2 to 0	P1n (n = 2 to 0)	I/O port

Besides functioning as a port, in control mode, it can also operate as the timer/counter I/O and external interrupt request input.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port 1	P10	TIUD10/TO10	Timer/counter I/O	K
	P11	TCUD10/INTP100	Timer/counter input or external interrupt request input	B
	P12	TCLR10/INTP101		

Caution P10 to P12 have hysteresis characteristics when the alternate functions are input, but not in the port mode.

(2) Setting of I/O mode and control mode

The port 1 mode register (PM1) is used to set the I/O mode of port 1 and the port 1 mode control register (PMC1) and port function control register 1 (PFC1) are used to set the operation in control mode.

(a) Port 1 mode register (PM1)

This register can be read or written in 8-bit or 1-bit units. Write 1 in bits 3 to 7.

	7	6	5	4	3	2	1	0	Address	After reset
PM1	1	1	1	1	1	PM12	PM11	PM10	FFFFF422H	FFH

Bit position	Bit name	Function
2 to 0	PM1n (n = 2 to 0)	Specifies input/output mode of P1n pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port 1 mode control register (PMC1)

This register can be read or written in 8-bit or 1-bit units. Write 0 in bits 3 to 7.

Caution The PMC11 and PMC12 bits are also used as external interrupts (INTP100 and INTP101). When not using them as external interrupts, mask interrupt requests (refer to 7.3.4 Interrupt control registers (xxICn)).

	7	6	5	4	3	2	1	0	Address	After reset
PMC1	0	0	0	0	0	PMC12	PMC11	PMC10	FFFFF442H	00H

Bit position	Bit name	Function
2	PMC12	Specifies operation mode of P12 pin. 0: I/O port mode 1: TCLR10 input mode or external interrupt request (INTP101) input mode
1	PMC11	Specifies operation mode of P11 pin. 0: I/O port mode 1: TCUD10 input mode or external interrupt request (INTP100) input mode
0	PMC10	Specifies operation mode of P10 pin. 0: I/O port mode 1: TIUD10 input mode or TO10 output mode

(c) Port 1 function control register (PFC1)

This register can be read or written in 8-bit or 1-bit units. Write 0 in bits other than 0.

Caution When port mode is specified by the port 1 mode control register (PMC1), the setting of this register is invalid.

	7	6	5	4	3	2	1	0	Address	After reset
PFC1	0	0	0	0	0	0	0	PFC10	FFFFF462H	00H

Bit position	Bit name	Function
0	PFC10	Specifies operation mode of P10 pin in control mode. 0: TIUD10 input mode 1: TO10 output mode

12.3.3 Port 2

Port 2 is an 8-bit I/O port in which input or output can be specified in 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFFFF404H	Undefined

Bit position	Bit name	Function
7 to 0	P2n (n = 7 to 0)	I/O port

Besides functioning as a port, in control mode, it also can operate as the timer/counter I/O and external interrupt request input.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port 2	P20	TI2/INTP20	Timer/counter input or external interrupt request input	B
	P21 to 24	TO21/INTP21 to TO24/INTP24	Timer/counter output or external interrupt request input	K
	P25	TCLR2/INTP25	Timer/counter input or external interrupt request input	B
	P26	TI3/TCLR3/INTP30		
	P27	TO3/INTP31	Timer/counter output or external interrupt request input	L

Caution P20, P21, and P25 to P27 have hysteresis characteristics when the alternate functions are input, but not in the port mode.

(2) Setting of I/O mode and control mode

The port 2 mode register (PM2) is used to set the I/O mode of port 2 and the port 2 mode control register (PMC2) and port 2 function control register (PFC2) are used to set the operation in control mode.

(a) Port 2 mode register (PM2)

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

	7	6	5	4	3	2	1	0	Address	After reset
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFFFF424H	FFH

Bit position	Bit name	Function
7 to 0	PM2n (n = 7 to 0)	Specifies input/output mode of P2n pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port 2 mode control register (PMC2)

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

Caution The PMC20, PMC25, and PMC26 bits also serve as external interrupts (INTP20, INTP25, and INTP30). When not using them as external interrupts, mask interrupt requests (refer to 7.3.4 Interrupt control registers (xxICn)).

	7	6	5	4	3	2	1	0	Address	After reset
PMC2	PMC27	PMC26	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20	FFFFF444H	00H

Bit position	Bit name	Function
7	PMC27	Specifies operation mode of P27 pin 0: I/O port mode 1: TO3 output mode or external interrupt request (INTP31) input mode
6	PMC26	Specifies operation mode of P26 pin 0: I/O port mode 1: TI3 and TCLR3 input mode or external interrupt request (INTP30) input mode
5	PMC25	Specifies operation mode of P25 pin 0: I/O port mode 1: TCLR2 input mode or external interrupt request (INTP25) input mode
4 to 1	PMC24 to PMC21	Specify operation mode of P24 to P21 pins 0: I/O port mode 1: TO24 to TO21 output mode or external interrupt request (INTP24 to INTP21) input mode
0	PMC20	Specifies operation mode of P20 pin 0: I/O port mode 1: TI2 input mode or external interrupt request (INTP20) input mode

(c) Port 2 function control register (PFC2)

This register can be read or written in 8-bit or 1-bit units. Write 0 in bits 0, 5, and 6.

Caution When port mode is specified by the port 2 mode control register (PMC2), the setting of this register is invalid.

	7	6	5	4	3	2	1	0	Address	After reset
PFC2	PFC27	0	0	PFC24	PFC23	PFC22	PFC21	0	FFFFF464H	00H

Bit position	Bit name	Function
7	PFC27	Specifies operation mode of P27 pin in control mode 0: External interrupt request (INTP31) input mode 1: TO3 output mode
4 to 1	PFC24 to PFC21	Specify operation mode of P24 to P21 pins in control mode 0: External interrupt request (INTP24 to INTP21) input mode 1: TO24 to TO21 output mode

12.3.4 Port 3

Port 3 is a 5-bit I/O port in which input or output can be specified in 1-bit units

	7	6	5	4	3	2	1	0	Address	After reset
P3	–	–	–	P34	P33	P32	P31	P30	FFFFF406H	Undefined

Bit position	Bit name	Function
4 to 0	P3n (n = 4 to 0)	I/O port

Besides functioning as a port, in control mode, it also can operate as the serial interface (UART0, UART1/CSI1) I/O.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port 3	P30	RXD0	Serial interface (UART0, UART1/CSI1) I/O	C
	P31	TXD0		A
	P32	RXD1/SI1		G
	P33	TXD1/SO1		F
	P34	$\overline{\text{ASCK1}}$/$\overline{\text{SCK1}}$		H

Caution P30, P32, and P34 have hysteresis characteristics when the alternate functions are input, but not in the port mode.

(2) Setting of I/O mode and control mode

The port 3 mode register (PM3) is used to set the I/O mode of port 3 and the port 3 mode control register (PMC3) and the port 3 function control register (PFC3) are used to set the operation in control mode.

(a) Port 3 mode register (PM3)

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

	7	6	5	4	3	2	1	0	Address	After reset
PM3	1	1	1	PM34	PM33	PM32	PM31	PM30	FFFFF426H	FFH

Bit position	Bit name	Function
4 to 0	PM3n (n = 4 to 0)	Specifies input/output mode of P3n pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port 3 mode control register (PMC3)

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

	7	6	5	4	3	2	1	0	Address	After reset
PMC3	0	0	0	PMC34	PMC33	PMC32	PMC31	PMC30	FFFFF446H	00H

Bit position	Bit name	Function
4	PMC34	Specifies operation mode of P34 pin 0: I/O port mode 1: $\overline{\text{ASCK1}}$/$\overline{\text{SCK1}}$ I/O mode
3	PMC33	Specifies operation mode of P33 pin 0: I/O port mode 1: TXD1/SO1 output mode
2	PMC32	Specifies operation mode of P32 pin 0: I/O port mode 1: RXD1/SI1 input mode
1	PMC31	Specifies operation mode of P31 pin 0: I/O port mode 1: TXD0 output mode
0	PMC30	Specifies operation mode of P30 pin 0: I/O port mode 1: RXD0 input mode

(c) Port 3 function control register (PFC3)

This register can be read or written in 8-bit or 1-bit units. Write 0 in bits other than 2 to 4.

Caution When port mode is specified by the port 3 mode control register (PMC3), the setting of this register is invalid.

	7	6	5	4	3	2	1	0	Address	After reset
PFC3	0	0	0	PFC34	PFC33	PFC32	0	0	FFFFF466H	00H

Bit position	Bit name	Function
4	PFC34	Specifies operation mode of P34 pin in control mode 0: $\overline{\text{ASCK1}}$ I/O mode 1: $\overline{\text{SCK1}}$ I/O mode
3	PFC33	Specifies operation mode of P33 pin in control mode 0: TXD1 output mode 1: SO1 output mode
2	PFC32	Specifies operation mode of P32 pin in control mode 0: RXD1 input mode 1: SI1 input mode

12.3.5 Port 4

Port 4 is a 3-bit I/O port in which input or output can be specified in 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
P4	–	–	–	–	–	P42	P41	P40	FFFFF408H	Undefined

Bit position	Bit name	Function
2 to 0	P4n (n = 2 to 0)	I/O port

Besides functioning as a port, in control mode, it also can operate as the serial interface (CSI0) I/O.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port 4	P40	SI0	Serial interface (CSI0) I/O	C
	P41	SO0		A
	P42	$\overline{\text{SCK0}}$		J

Caution P40 and P42 have hysteresis characteristics when the alternate functions are input, but not in the port mode.

(2) Setting of I/O mode and control mode

The port 4 mode register (PM4) is used to set the I/O mode of port 4 and the port 4 mode control register (PMC4) is used to set the operation in control mode.

(a) Port 4 mode register (PM4)

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

	7	6	5	4	3	2	1	0	Address	After reset
PM4	1	1	1	1	1	PM42	PM41	PM40	FFFFF428H	FFH

Bit position	Bit name	Function
2 to 0	PM4n (n = 2 to 0)	Specifies input/output mode of P4n pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port 4 mode control register (PMC4)

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

	7	6	5	4	3	2	1	0	Address	After reset
PMC4	0	0	0	0	0	PMC42	PMC41	PMC40	FFFFF448H	00H

Bit position	Bit name	Function
2	PMC42	Specifies operation mode of P42 pin 0: I/O port mode 1: $\overline{\text{SCK0}}$ I/O mode
1	PMC41	Specifies operation mode of P41 pin 0: I/O port mode 1: SO0 output mode
0	PMC40	Specifies operation mode of P40 pin 0: I/O port mode 1: SI0 input mode

12.3.6 Port DH

Port DH is a 6-bit I/O port in which input or output can be specified in 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
PDH	–	–	PDH5	PDH4	PDH3	PDH2	PDH1	PDH0	FFFFF006H	Undefined

Bit position	Bit name	Function
5 to 0	PDHn (n = 5 to 0)	I/O port

Besides functioning as a port, in control mode, this can operate as an address bus when memory is expanded externally.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port DH	PDH5 to PDH0	A21 to A16	Memory expansion address bus	N

(2) Setting of I/O mode and control mode

The port DH mode register (PMDH) is used to set the I/O mode of port DH and the port DH mode control register (PMCDH) is used to set the operation in control mode.

(a) Port DH mode register (PMDH)

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

	7	6	5	4	3	2	1	0	Address	After reset
PMDH	1	1	PMDH5	PMDH4	PMDH3	PMDH2	PMDH1	PMDH0	FFFFF026H	FFH

Bit position	Bit name	Function
5 to 0	PMDHn (n = 5 to 0)	Specifies input/output mode of PDHn pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port DH mode control register (PMCDH)

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

Caution Set bits 7 and 6 as follows.

Operation Mode	Bit 7	Bit 6
Single-chip mode	0	0
ROMless mode	1	1

	7	6	5	4	3	2	1	0	Address	After reset[Note]
PMCDH	PMCDH7	PMCDH6	PMCDH5	PMCDH4	PMCDH3	PMCDH2	PMCDH1	PMCDH0	FFFFF046H	00H/FFH

Note 00H: Single-chip mode
FFH: ROMless mode

Bit position	Bit name	Function
5 to 0	PMCDHn (n = 5 to 0)	Specifies operation mode of PDHn pin 0: I/O port mode 1: A21 to A16 output mode

12.3.7 Port DL

Port DL is a 16-bit I/O port in which input or output can be specified in 1-bit units.

When using the higher 8 bits of PDL as PDLH and the lower 8 bits as PDLL, it can be used as an 8-bit I/O port that can specify input/output in 1-bit units.

	15	14	13	12	11	10	9	8	Address	After reset
PDL	PDL15	PDL14	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8	FFFFF005H	Undefined

	7	6	5	4	3	2	1	0	Address
	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0	FFFFF004H

Bit position	Bit name	Function
15 to 0	PDLn (n = 15 to 0)	I/O port

Besides functioning as a port, in control mode, this can operate as an address/data bus when memory is expanded externally.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port DL	PDL15 to PDL0	AD15 to AD0	Memory expansion address/data bus	M

(2) Setting of I/O mode and control mode

The port DL mode register (PMDL) is used to set the I/O mode of port DL and the port DL mode control register (PMCDL) is used to set the operation in control mode.

(a) Port DL mode register (PMDL)

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

When using the higher 8 bits of the PMDL register as the PMDLH register and the lower 8 bits as the PMDLL register, it can be read or written in 8-bit or 1-bit units.

	15	14	13	12	11	10	9	8	Address	After reset
PMDL	PMDL15	PMDL14	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8	FFFFF025H	FFFFH

7	6	5	4	3	2	1	0	Address
PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0	FFFFF024H

Bit position	Bit name	Function
15 to 0	PMDLn (n = 15 to 0)	Specifies input/output mode of PDLn pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port DL mode control register (PMCDL)

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

When using the higher 8 bits of the PMCDL register as the PMCDLH register and the lower 8 bits as the PMCDLL register, it can be read or written in 8-bit or 1-bit units.

	15	14	13	12	11	10	9	8	Address	After reset[Note]
PMCDL	PMCDL15	PMCDL14	PMCDL13	PMCDL12	PMCDL11	PMCDL10	PMCDL9	PMCDL8	FFFFF045H	0000H/FFFFH

7	6	5	4	3	2	1	0	Address
PMCDL7	PMCDL6	PMCDL5	PMCDL4	PMCDL3	PMCDL2	PMCDL1	PMCDL0	FFFFF044H

Note 0000H : Single-chip mode
FFFFH: ROMless mode

Bit position	Bit name	Function
15 to 0	PMCDLn (n = 15 to 0)	Specifies operation mode of PDLn pin. 0: I/O port mode 1: AD15 to AD0 I/O mode

12.3.8 Port CT

Port CT is a 4-bit I/O port in which input or output can be specified in 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
PCT	–	PCT6	–	PCT4	–	–	PCT1	PCT0	FFFFF00AH	Undefined

Bit position	Bit name	Function
6, 4, 1, 0	PCTn (n = 6, 4, 1, 0)	I/O port

Besides functioning as a port, in control mode, this can operate as control signal outputs when memory is expanded externally.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port CT	PCT0	$\overline{\text{LWR}}$	Write strobe signal output	I
	PCT1	$\overline{\text{UWR}}$		
	PCT4	$\overline{\text{RD}}$	Read strobe signal output	
	PCT6	ASTB	Address strobe signal output	

(2) Setting of I/O mode and control mode

The port CT mode register (PMCT) is used to set the I/O mode of port CT and the port CT mode control register (PMCCT) is used to set the operation in control mode.

(a) Port CT mode register (PMCT)

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

	7	6	5	4	3	2	1	0	Address	After reset
PMCT	1	PMCT6	1	PMCT4	1	1	PMCT1	PMCT0	FFFFF02AH	FFH

Bit position	Bit name	Function
6, 4, 1, 0	PMCTn (n = 6, 4, 1, 0)	Specifies input/output mode of PCTn pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port CT mode control register (PMCCT)

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

	7	6	5	4	3	2	1	0	Address	After reset[Note]
PMCCT	0	PMCCT6	0	PMCCT4	0	0	PMCCT1	PMCCT0	FFFFF04AH	00H/53H

Note 00H: Single-chip mode
53H: ROMless mode

Bit position	Bit name	Function
6	PMCCT6	Specifies operation mode of PCT6 pin 0: I/O port mode 1: ASTB output mode
4	PMCCT4	Specifies operation mode of PCT4 pin 0: I/O port mode 1: $\overline{RD}$ output mode
1	PMCCT1	Specifies operation mode of PCT1 pin 0: I/O port mode 1: $\overline{UWR}$ output mode
0	PMCCT0	Specifies operation mode of PCT0 pin 0: I/O port mode 1: $\overline{LWR}$ output mode

12.3.9 Port CM

Port CM is a 2-bit I/O port in which input or output can be specified in 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
PCM	–	–	–	–	–	–	PCM1	PCM0	FFFFF00CH	Undefined

Bit position	Bit name	Function
1, 0	PCMn (n = 1, 0)	I/O port

Besides functioning as a port, in control mode, this can operate as the wait insertion signal input and internal system clock output.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port CM	PCM0	$\overline{\text{WAIT}}$[Note]	Wait insertion signal input	D
	PCM1	CLKOUT	Internal system clock output	I

Note In the ROMless mode, the default operation mode of the PCM0 pin is the $\overline{\text{WAIT}}$ input mode. When unused, fix the pin to the inactive level. When used as a port, this pin functions in the control mode until the port mode is set using the port CM mode control register (PMCCM). Set this pin to the inactive level during this period.

(2) Setting of I/O mode and control mode

The port CM mode register (PMCM) is used to set the I/O mode of port CM and the CM mode control register (PMCCM) is used to set the operation in control mode.

(a) Port CM mode register (PMCM)

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

	7	6	5	4	3	2	1	0	Address	After reset
PMCM	1	1	1	1	1	1	PMCM1	PMCM0	FFFFF02CH	FFH

Bit position	Bit name	Function
1, 0	PMCMn (n = 1, 0)	Specifies input/output mode of PCMn pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port CM mode control register (PMCCM)

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

	7	6	5	4	3	2	1	0	Address	After reset[Note]
PMCCM	0	0	0	0	0	0	PMCCM1	PMCCM0	FFFFF04CH	00H/03H

Note 00H: Single-chip mode
03H: ROMless mode

Bit position	Bit name	Function
1	PMCCM1	Specifies operation mode of PCM1 pin 0: I/O port mode 1: CLKOUT output mode
0	PMCCM0	Specifies operation mode of PCM0 pin 0: I/O port mode 1: $\overline{\text{WAIT}}$ input mode

12.4 Operation of Port Function

The operation of a port differs depending on whether it is set in the input or output mode, as follows.

12.4.1 Writing to I/O port

(1) In output mode

A value can be written to the output latch (Pn) by writing it to the port n register (Pn). The contents of the output latch are output from the pin.

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

(2) In input mode

A value can be written to the output latch (Pn) by writing it to the port n register (Pn). However, the status of the pin does not change because the output buffer is off.

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

Caution A bit manipulation instruction (CLR1, SET1, NOT1) manipulates 1 bit but accesses a port in 8-bit units. If this instruction is executed to manipulate a port with a mixture of input and output bits, the contents of the output latch of a pin set in the input mode, in addition to the bit to be manipulated, are overwritten to the current input pin status and become undefined.

12.4.2 Reading from I/O port

(1) In output mode

The contents of the output latch (Pn) can be read by reading the port n register (Pn). The contents of the output latch do not change.

(2) In input mode

The status of the pin can be read by reading the port n register (Pn). The contents of the output latch (Pn) do not change.

12.4.3 Output status of alternate function in control mode

The status of a port pin can be read by setting the port n mode register (PMn) to the input mode regardless of the setting of the PMCn register. If the PMn register is set to the output mode, the value of the port n register (Pn) can be read in the port mode, and the output status of the alternate function can be read in the control mode.

12.5 Noise Eliminator

12.5.1 Interrupt pins

A timing controller to guarantee the noise elimination time shown below is added to the pins that operate as NMI and valid edge inputs in port control mode. A signal input that changes in less than this elimination time is not accepted internally.

Pin	Noise Elimination Time
P00/NMI P01/ESO0/INTP0, P02/ESO1/INTP1 P03/ADTRG0/INTP2, P04/ADTRG1/INTP3 P05/INTP4/TO3OFF	Analog delay (several 10 ns)

Cautions 1. The above non-maskable/maskable interrupt pins are used to release standby mode. A clock control timing circuit is not used since the internal system clock is stopped in standby mode.

2. The noise eliminator is valid only in control mode.

12.5.2 Timer 10, timer 3 input pins

Noise filtering using the clock sampling shown below is added to the pins that operate as valid edge inputs to timer 10 and timer 3. A signal input that changes in less than these elimination times is not accepted internally.

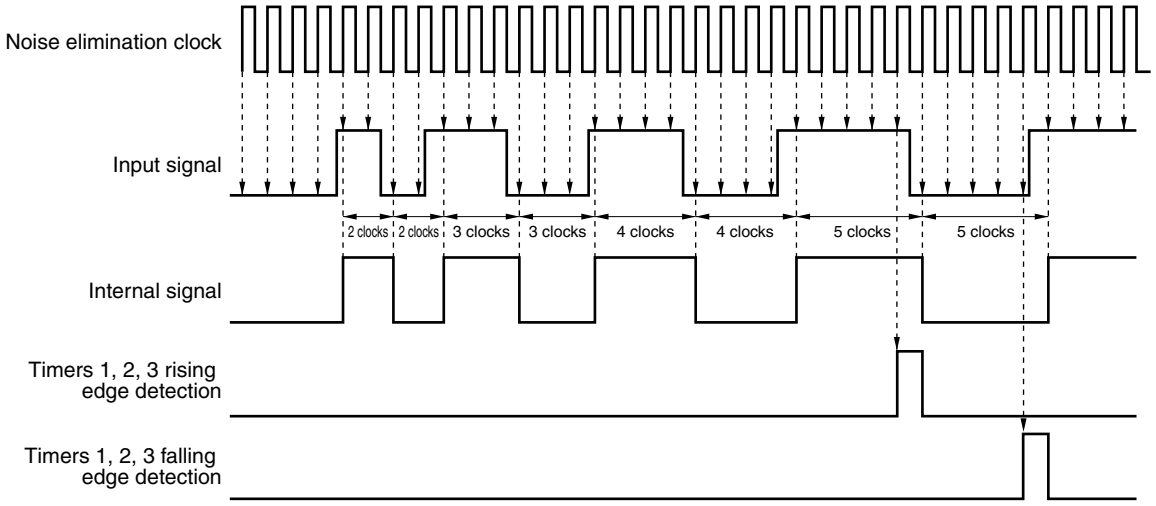
Pin		Noise Elimination Time	Sampling Clock
Timer 10	P10/TIUD10/TO10 P11/TCUD10/INTP100 P12/TCLR10/INTP101	4 to 5 clocks	Select from f_{XXTM10} $f_{XXTM10}/2$ $f_{XXTM10}/4$ $f_{XXTM10}/8$
Timer 3	P26/TI3/INTP30/TCLR3		Select from $f_{XXTM3}/2$ $f_{XXTM3}/4$ $f_{XXTM3}/8$ $f_{XXTM3}/16$
	P27/TO3/INTP31		Select from $f_{XXTM3}/32$ $f_{XXTM3}/64$ $f_{XXTM3}/128$ $f_{XXTM3}/256$

Cautions 1. Since the above pin noise filtering uses clock sampling, input signals are not received when the CPU clock is stopped.

2. The noise eliminator is valid only in control mode.

Remark f_{XXTM10}: Clock of TM10 selected in PRM02 register (be sure to set PRM02 = 01H)
f_{XXTM3}: Clock of TM3 selected in PRM03 register

Figure 12-15. Example of Noise Elimination Timing



Caution If there are three or less noise elimination clocks while the timer 1 or 3 input signal is high level (or low level), the input pulse is eliminated as noise. If it is sampled at least four times, the edge is detected as valid input.

(1) Timer 10 noise elimination time selection register (NRC10)

The NRC10 register is used to set the clock source of timer 10 input pin noise elimination time.

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

★ **Caution The noise elimination function starts operating by setting the TM1CE0 bit of the TMC10 register to 1 (enabling count operations).**

	7	6	5	4	3	2	1	0	Address	After reset
NRC10	0	0	0	0	0	0	NRC101	NRC100	FFFFF5F8H	00H

Bit position	Bit name	Function
1, 0	NRC101, NRC100	Selects the TIUD10/TO10, TCUD10/INTP100, and TCLR10/INTP101 pin noise elimination clocks.

NRC101	NRC100	Noise elimination clocks
0	0	$f_{XXTM10}/8$
0	1	$f_{XXTM10}/4$
1	0	$f_{XXTM10}/2$
1	1	f_{XXTM10}

Remark f_{XXTM10}: Clock of TM10 selected by PRM02 register (be sure to set PRM02 = 01H)

(2) Timer 3 noise elimination time selection register (NRC3)

The NRC3 register is used to set the clock source of the timer 3 input pin noise elimination time.

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

★ **Caution The noise elimination function starts operating by setting the TM3CE0 bit of the TMC30 register to 1 (enabling count operations).**

	7	6	5	4	3	2	1	0	Address	After reset
NRC3	0	0	0	0	NRC33	NRC32	NRC31	NRC30	FFFFF698H	00H

Bit position	Bit name	Function
3, 2	NRC33, NRC32	Selects the TO3/INTP31 pin noise elimination clock.

NRC33	NRC32	Noise elimination clock
0	0	f_{XXTM3}/256
0	1	f_{XXTM3}/128
1	0	f_{XXTM3}/64
1	1	f_{XXTM3}/32

Remark f_{XXTM3}: Clock of TM3 selected by PRM03 register

Bit position	Bit name	Function
1, 0	NRC31, NRC30	Selects the TI3/INTP30/TCLR3 pin noise elimination clock.

NRC31	NRC30	Noise elimination clocks
0	0	f_{XXTM3}/16
0	1	f_{XXTM3}/8
1	0	f_{XXTM3}/4
1	1	f_{XXTM3}/2

Remark f_{XXTM3}: Clock of TM3 selected by PRM03 register

12.5.3 Timer 2 input pins

A noise eliminator using analog filtering and digital filtering using clock sampling are added to the timer 2 input pins. A signal input that changes in less than this elimination time is not accepted internally.

Pin	Analog Filter Noise Elimination Time	Digital Filter	
		Noise Elimination Time	Sampling Clock
P20/TI2/INTP20 P21/TO21/INTP21 to P24/TO24/INTP24 P25/TCLR2/INTP25	10 to 100 ns	4 to 5 clocks	f_{XXTM2}

Cautions 1. Since digital filtering uses clock sampling, if it is selected, input signals are not received when the CPU clock is stopped.

2. The noise eliminator is valid only in control mode.

3. Refer to Figure 12-15 for an example of a noise eliminator.

Remark f_{XXTM2}: Clock of TM20 and TM21 selected in PRM02 register (be sure to set PRM02 = 01H)

(1) Timer 2 input filter mode registers 0 to 5 (FEM0 to FEM5)

The FEMn registers are used to specify timer 2 input pin filtering and to set the clock source of noise elimination time and the input valid edge.

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

Cautions 1. Be sure to clear (0) the STFTE bit of timer 2 clock stop register 0 (STOPTE0) even when using the TI2/INTP20, TO21/INTP21, TO22/INTP22, TO23/INTP23, TO24/INTP24, and TCLR2/INTP25 pins as INTP20, INTP21, INTP22, INTP23, INTP24, and INTP25, respectively, and not using timer 2.

2. Setting the trigger mode of the INTP2n pin should be performed after setting the PMC2 register.

If the PMC2 register is set after setting the FEMn register, an invalid interrupt may occur when the PMC2 register is set (n = 0 to 5).

★ **3. The noise elimination function starts operating by setting the CEEn bit of the TCRE0 register to 1 (enabling count operations).**

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
FEM0	DFEN00	0	0	0	EDGE010	EDGE000	TMS010	TMS000	FFFFF630H	00H
					INTP20					
FEM1	DFEN01	0	0	0	EDGE011	EDGE001	TMS011	TMS001	FFFFF631H	00H
					INTP21					
FEM2	DFEN02	0	0	0	EDGE012	EDGE002	TMS012	TMS002	FFFFF632H	00H
					INTP22					
FEM3	DFEN03	0	0	0	EDGE013	EDGE003	TMS013	TMS003	FFFFF633H	00H
					INTP23					
FEM4	DFEN04	0	0	0	EDGE014	EDGE004	TMS014	TMS004	FFFFF634H	00H
					INTP24					
FEM5	DFEN05	0	0	0	EDGE015	EDGE005	TMS015	TMS005	FFFFF635H	00H
					INTP25					

Bit position	Bit name	Function
7	DFEN0n	Specifies the INTP2n pin filter. 0: Analog filter 1: Digital filter **Caution When the DFEN0n bit = 1, the sampling clock of the digital filter is f_{XXTM2} (clock selected by the PRM02 register).**

Remark n = 0 to 5

(2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3, 2</td><td>EDGE01n,
EDGE00n</td><td>Specifies the INTP2n pin valid edge.

<table>
<tr><th>EDGE01n</th><th>EGE00n</th><th>Operation</th></tr>
<tr><td>0</td><td>0</td><td>Interrupt due to INTCC2n^{Note}</td></tr>
<tr><td>0</td><td>1</td><td>Rising edge</td></tr>
<tr><td>1</td><td>0</td><td>Falling edge</td></tr>
<tr><td>1</td><td>1</td><td>Both rising and falling edges</td></tr>
</table>

Note Specify when selecting INTCC2n according to a match of TM20, TM21 and the subchannel compare registers (TMS01n, TMS00n bit settings) (n = 0 to 5).</td></tr>
<tr><td>1, 0</td><td>TMS01n,
TMS00n</td><td>Selects capture input^{Note}.

<table>
<tr><th>TMS01n</th><th>TMS00n</th><th>Operation</th></tr>
<tr><td>0</td><td>0</td><td>Used as pin</td></tr>
<tr><td>0</td><td>1</td><td>Digital filter (noise eliminator specification)</td></tr>
<tr><td>1</td><td>0</td><td>Capture to subchannel 1 according to timer</td></tr>
<tr><td>1</td><td>1</td><td>Capture to subchannel 2 according to timer</td></tr>
</table>

Note Capture input according to INTCM100 and INTCM101 can be selected only for the FEM1 and FEM2 registers. Set the values of the TMS01m and TMS00m bits in the FEMm register to 00B or 01B. Settings other than these are prohibited (m = 1, 3 to 5).
Capture according to INTP21, INTP22 and INTCM100, INTCM101 is possible for subchannel 1 and subchannel 2 of timer 2.
Examples are shown below.
(a) Capture subchannel 1 on INTCM101
FEM1 register = xxxxxx10B
TMIC0 register = 00000010B
(b) Capture subchannel 2 on INTCM101
FEM2 register = xxxxxx11B
TMIC0 register = 00001000B</td></tr>
</table>

Remark n = 0 to 5

12.6 Cautions

12.6.1 Hysteresis characteristics

The following ports do not have hysteresis characteristics in the port mode.

P10 to P12
P20, P21, P25 to P27
P30, P32, P34
P40, P42

CHAPTER 13 RESET FUNCTION

When a low level is input to the $\overline{\text{RESET}}$ pin, the system is reset and each hardware item of the V850E/IA2 is initialized to its initial status.

When the $\overline{\text{RESET}}$ pin changes from low level to high level, the reset status is released and the CPU starts program execution. Initialize the contents of various registers as needed within the program.

13.1 Features

- Noise elimination using analog delay (approx. 60 ns) at reset pin ($\overline{\text{RESET}}$)

13.2 Pin Functions

During a system reset period, most pin output is high impedance (all pins except CLKOUT[Note], $\overline{\text{RESET}}$, X2, V_{DD}, V_{SS}, V_{SS3}, CV_{SS}, RV_{DD}, REGOUT, REGIN, AV_{DD0}, AV_{DD1}, AV_{SS0}, and AV_{SS1} pins).

Thus, if memory is extended externally, a pull-up (or pull-down) resistor must be attached to each pin of ports DH, DL, CT, and CM. If there are no resistors, the external memory that is connected may be destroyed when these pins become high impedance.

Similarly, perform pin processing so that on-chip peripheral I/O function signal outputs and output ports are not affected.

Note In ROMless mode, CLKOUT signals are also output during a reset period. In single-chip mode, CLKOUT signals are not output until the PMCCM register is set.

Table 13-1 shows the operation status of each pin during a reset period.

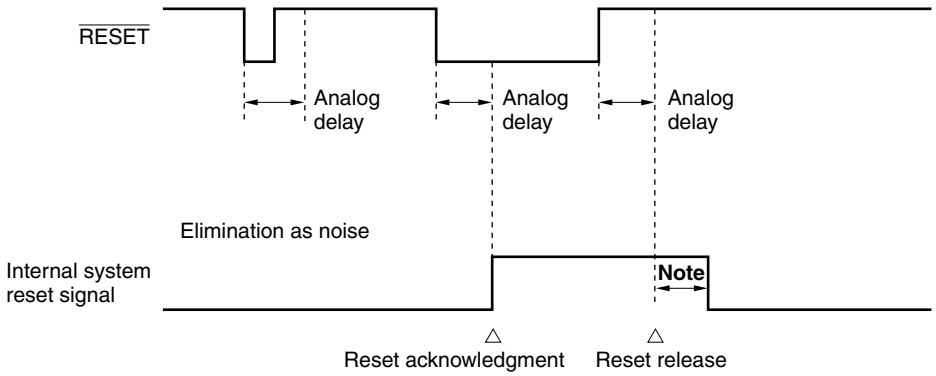
Table 13-1. Operation Status of Each Pin During Reset Period

Pin Name		Pin Status	
		In Single-Chip Mode	In ROMless Mode
External access pin	A16 to A21, AD0 to AD15, $\overline{\text{LWR}}$, $\overline{\text{UWR}}$, $\overline{\text{RD}}$, ASTB, $\overline{\text{WAIT}}$	High impedance (Input port mode)	High impedance
	CLKOUT	High impedance (Input port mode)	Operation
Port pin[Note]	Port 0 to 4	High impedance (Input port mode)	
	Ports CM, CT, DH, DL	High impedance (Input port mode)	Refer to the description of the external access pin. (control mode)
Dedicated function pin	TO0n0 to TO0n5 (Pins dedicated to timer 0 output)	High impedance	
	ANI00 to ANI05, ANI10 to ANI17 (Pins dedicated to A/D converter input)	High impedance (A/D converter input)	

Note The names of the control pins that function alternately as port pins are omitted.

Remark n = 0, 1

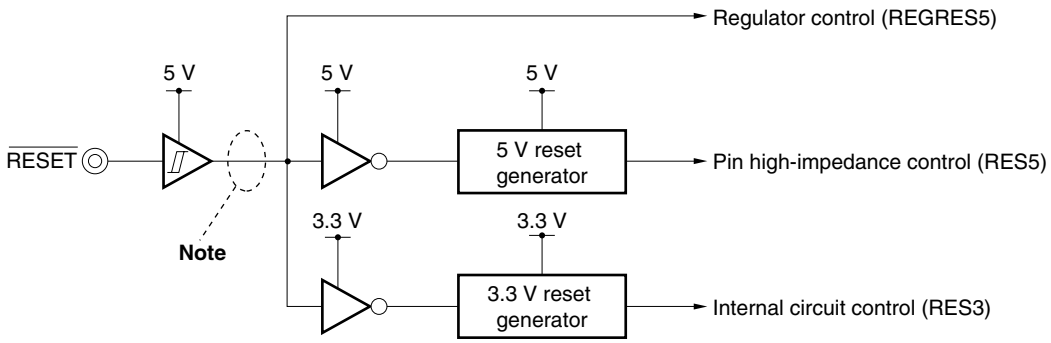
(1) Reset signal acknowledgment



Note The internal system reset signal remains active for a period of at least 4 system clocks after the timing of a reset release by the $\overline{\text{RESET}}$ pin.

(2) Reset at power-on

<1> Reset circuit



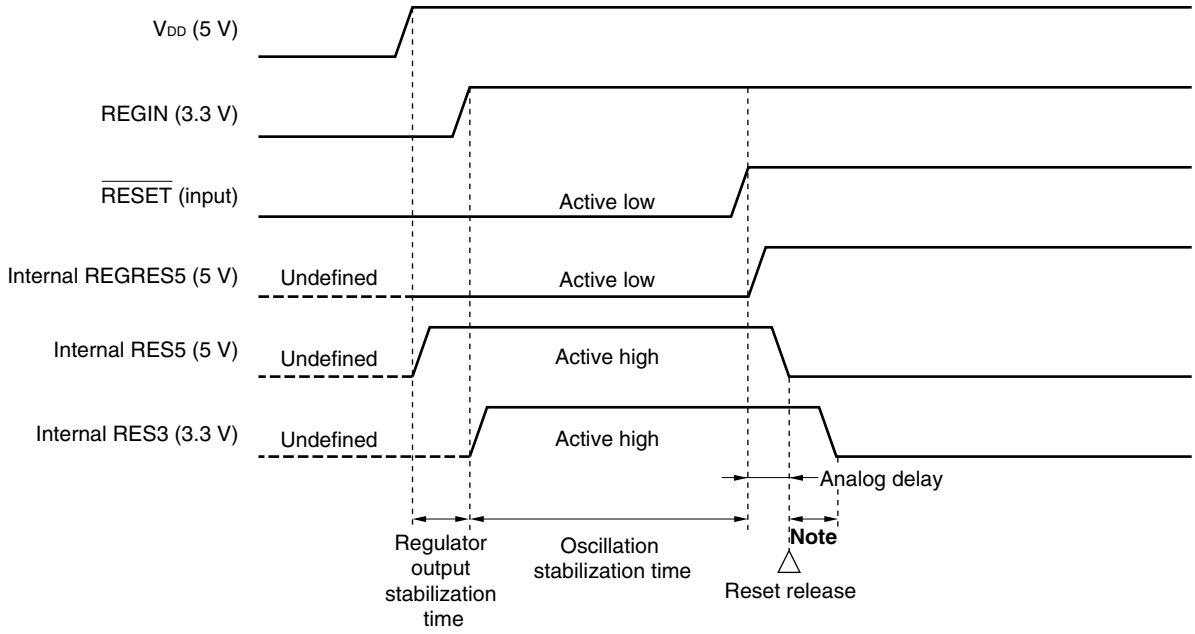
Note Apply 5 V initially. If 5 V is not applied initially, this level cannot be determined, and a reset will not occur.

Caution Apply power in the following sequence.

<1> 5 V power supply

<2> 3.3 V power supply

<2> Reset timing



Note The internal system reset signal stays active for at least 4 system clocks after the reset status caused by the $\overline{\text{RESET}}$ pin is released.

<3> Description

A reset operation at power-on (power supply application) must guarantee "regulator output stabilization time + oscillation stabilization time" from power-on until reset acknowledgment due to the low level width of the $\overline{\text{RESET}}$ signal.

Cautions 1. The V850E/IA2 has an internal regulator that generates 3.3 V from a 5 V system power supply. Therefore, 3.3 V system power is supplied after the lapse of the regulator output stabilization time after 5 V power was supplied. When supplying the two power supplies from external supplies with the regulator turned off, be sure to supply 5 V system power first.

2. The V850E/IA2 is internally reset after 3.3 V system power has been supplied. During the regulator output stabilization time, the internal circuits may not be reset when only 5 V system power is being supplied. Consequently, the pins may output undefined levels. For this reason, the V850E/IA2 makes the pins listed in (a) below that may affect the application system (mainly the I/O pins of the internal timers) go into a high-impedance state (refer to (b) and (c) below).

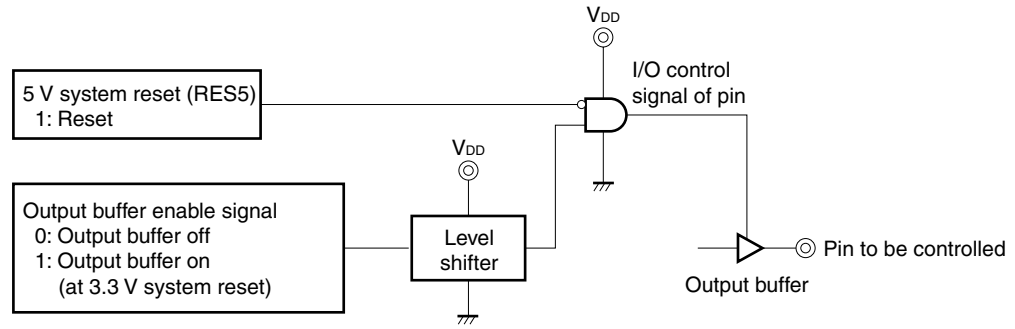
Note that pins other than those to be controlled do not go into a high-impedance state unless supplied with 3.3 V system power.

The pins listed in (a) may also output undefined levels until a 5 V reset (internal RES5) occurs (after the power was supplied until VDD reaches approximately 1.8 V (reference value)) if the 5 V system power supply is gradually stabilized. The undefined level output time depends on how rapidly the power supply is stabilized. Attention must be paid when the system requires several tens of ms for the 5 V system power to stabilize.

(a) Pins to be controlled

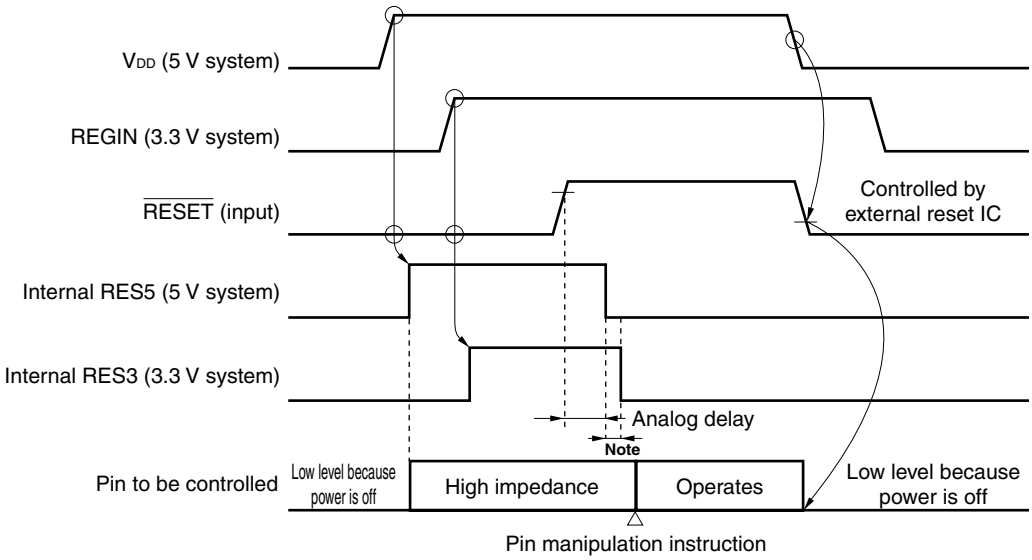
TO000 to TO005, TO010 to TO015, P10/TO10/TIUD10, P11/INTP100/TCUD10, P12/INTP101/TCLR10, P20/INTP20/TI2, P21/INTP21/TO21, P22/INTP22/TO22, P23/INTP23/TO23, P24/INTP24/TO24, P25/INTP25/TCLR2, P26/TCLR3/INTP30/TI3, P27/INTP31/TO3

(b) Circuit of above pins



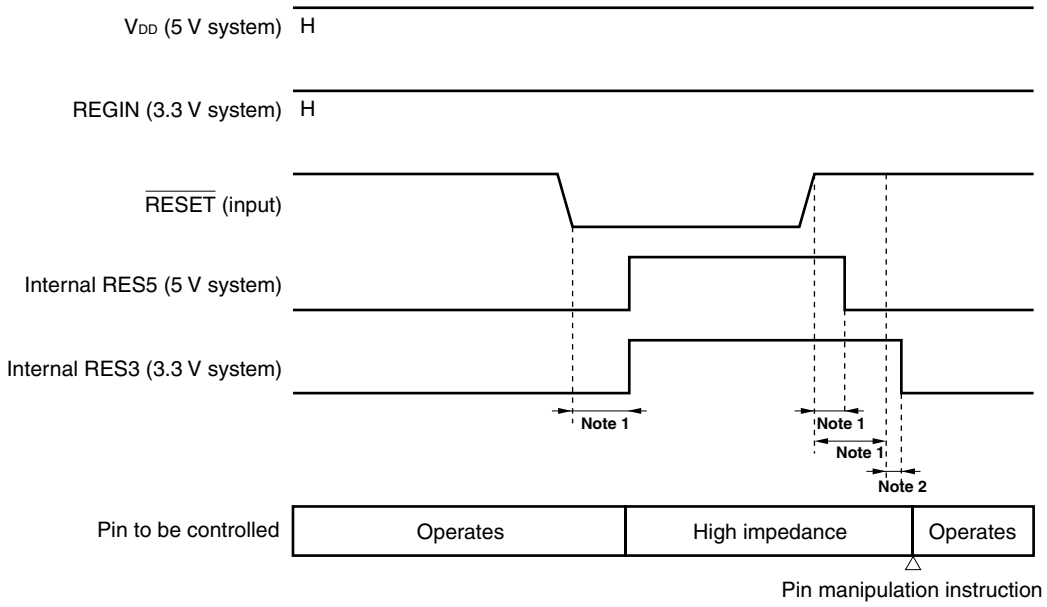
(c) Internal reset of 5 V system/3.3 V system power supply

(i) Operation on turning ON/OFF power



Note The internal system reset signal stays active for at least 4 system clocks after the reset status caused by the $\overline{\text{RESET}}$ pin is released.

(ii) Reset during normal operation



Notes 1. Analog delay

2. The internal system reset signal stays active for at least 4 system clocks after the reset status caused by the $\overline{\text{RESET}}$ pin is released.

13.3 Initialization

Initialize the contents of each register as needed within the program.

Table 13-2 shows the initial values of the CPU, internal RAM, and on-chip peripheral I/O after reset.

Table 13-2. Initial Values of CPU, Internal RAM, and On-Chip Peripheral I/O After Reset (1/5)

On-Chip Hardware		Register Name	Initial Value After Reset
CPU	Program registers	General-purpose register (r0)	00000000H
		General-purpose registers (r1 to r31)	Undefined
		Program counter (PC)	00000000H
	System registers	Status save registers during interrupt (EIPC, EIPSW)	Undefined
		Status save registers during NMI (FEPC, FEPSW)	Undefined
		Interrupt cause register (ECR)	00000000H
		Program status word (PSW)	00000020H
		Status save registers during CALLT execution (CTPC, CTPSW)	Undefined
		Status save registers during exception/debug trap (DBPC, DBPSW)	Undefined
		CALLT base pointer (CTBP)	Undefined
Internal RAM		–	Undefined
On-chip peripheral I/O	Bus control function	Chip area selection control register n (CSCn) (n = 0, 1)	2C11H
		Bus size configuration register (BSC)	5555H
		System wait control register (VSWC)	77H
	Memory control function	Bus cycle type configuration register n (BCTn) (n = 0,1)	CCCCH
		Data wait control register n (DWCn) (n = 0,1)	3333H
		Address wait control register (AWC)	0000H
		Bus cycle control register (BCC)	AAAAH
	DMA function	DMA source address register nL (DSAnL) (n = 0 to 3)	Undefined
		DMA source address register nH (DSAnH) (n = 0 to 3)	Undefined
		DMA destination address register nL (DDAnL) (n = 0 to 3)	Undefined
		DMA destination address register nH (DDAnH) (n = 0 to 3)	Undefined
		DMA transfer count register n (DBCn) (n = 0 to 3)	Undefined
		DMA addressing control register n (DADCn) (n = 0 to 3)	0000H
		DMA channel control register n (DCHCn) (n = 0 to 3)	00H
		DMA disable status register (DDIS)	00H
		DMA restart register (DRST)	00H
		DMA trigger source register n (DTFRn) (n = 0 to 3)	00H
	Interrupt/ exception control function	In service priority register (ISPR)	00H
		External interrupt mode register n (INTMn) (n = 0 to 2)	00H
		Interrupt mask register n (IMRn) (n = 0 to 3)	FFFFH
		Interrupt mask register nL (IMRnL) (n = 0 to 3)	FFH
		Interrupt mask register nH (IMRnH) (n = 0 to 3)	FFH
		Signal edge selection register 10 (SESA10)	00H

Table 13-2. Initial Values of CPU, Internal RAM, and On-Chip Peripheral I/O After Reset (2/5)

On-Chip Hardware		Register Name	Initial Value After Reset
On-chip peripheral I/O	Interrupt/ exception control function	Valid edge selection register (SESC)	00H
		Timer 2 input filter mode register n (FEMn) (n = 0 to 5)	00H
		Interrupt control registers (P0IC0 to P0IC4, DETIC0, DETIC1, TM0IC0, TM0IC1, TM2IC0, TM2IC1, TM3IC0, CC10IC0, CC10IC1, CC2IC0 to CC2IC5, CC3IC0, CC3IC1, CM00IC1, CM01IC1, CM02IC1, CM03IC0, CM03IC1, CM04IC0, CM04IC1, CM05IC0, CM05IC1, CM10IC0, CM10IC1, CM4IC0, DMAIC0 to DMAIC3, CSIIC0, CSIIC1, SEIC0, SRIC0, SRIC1, STIC0, STIC1, ADIC0, ADIC1)	47H
	Power save control function	Command register (PRCMD)	Undefined
		Power save control register (PSC)	00H
		Clock control register (CKC)	00H
		Power save mode register (PSMR)	00H
		Lock register (LOCKR)	0000000xB
	System control	Peripheral command register (PHCMD)	Undefined
		Peripheral status register (PHS)	00H
	Timer 0	Dead time timer reload register n (DTRRn) (n = 0,1)	0FFFH
		Buffer registers CM0n, CM1n (BFCM0n, BFCM1n) (n = 0 to 5)	FFFFH
		Timer control register 0n (TMC0n) (n = 0,1)	0508H
		Timer control register 0nL (TMC0nL) (n = 0, 1)	08H
		Timer control register 0nH (TMC0nH) (n = 0, 1)	05H
		Timer unit control register 0n (TUC0n) (n = 0,1)	01H
		Timer output mode register n (TOMRn) (n = 0,1)	00H
		PWM software timing output register n (PSTOn) (n = 0,1)	00H
		PWM output enable register n (POERn) (n = 0,1)	00H
		TOMR write enable register n (SPECn) (n = 0,1)	0000H
		Timer 0 clock selection register (PRM01)	00H
	Timer 1	Timer 10 (TM10)	0000H
		Compare register 1n (CM1n) (n = 00, 01)	0000H
		Capture/compare register 1n (CC1n) (n = 00, 01)	0000H
		Capture/compare control register 0 (CCR0)	00H
		Timer unit mode register 0 (TUM0)	00H
		Timer control register 10 (TMC10)	00H
		Signal edge selection register 10 (SESA10)	00H
		Prescaler mode register 10 (PRM10)	07H
		Status register 0 (STATUS0)	00H
		Timer connection selection register 0 (TMIC0)	00H
		Timer 1/timer 2 clock selection register (PRM02)	00H
		CC101 capture input selection register (CSL10)	00H
		Timer 10 noise elimination time selection register (NRC10)	00H

Table 13-2. Initial Values of CPU, Internal RAM, and On-Chip Peripheral I/O After Reset (3/5)

On-Chip Hardware		Register Name	Initial Value After Reset
On-chip peripheral I/O	Timer 2	Timer 2 clock stop register 0 (STOPTE0)	0000H
		Timer 2 clock stop register 0L (STOPTE0L)	00H
		Timer 2 clock stop register 0H (STOPTE0H)	00H
		Timer 2 count clock/control edge selection register 0 (CSE0)	0000H
		Timer 2 count clock/control edge selection register 0L (CSE0L)	00H
		Timer 2 count clock/control edge selection register 0H (CSE0H)	00H
		Timer 2 subchannel input event edge selection register 0 (SESE0)	0000H
		Timer 2 subchannel input event edge selection register 0L (SESE0L)	00H
		Timer 2 subchannel input event edge selection register 0H (SESE0H)	00H
		Timer 2 time base control register 0 (TCRE0)	0000H
		Timer 2 time base control register 0L (TCRE0L)	00H
		Timer 2 time base control register 0H (TCRE0H)	00H
		Timer 2 output control register 0 (OCTLE0)	0000H
		Timer 2 output control register 0L (OCTLE0L)	00H
		Timer 2 output control register 0H (OCTLE0H)	00H
		Timer 2 subchannels 0 and 5 capture/compare control register (CMSE050)	0000H
		Timer 2 subchannels 1 and 2 capture/compare control register (CMSE120)	0000H
		Timer 2 subchannels 3 and 4 capture/compare control register (CMSE340)	0000H
		Timer 2 subchannel n secondary capture/compare register (CVSEn0) (n = 0 to 4)	0000H
		Timer 2 subchannel n main capture/compare register (CVPEn0) (n = 0 to 4)	0000H
		Timer 2 subchannel n capture/compare register (CVSEn0) (n = 0, 5)	0000H
		Timer 2 time base status register 0 (TBSTATE0)	0101H
		Timer 2 time base status register 0L (TBSTATE0L)	01H
		Timer 2 time base status register 0H (TBSTATE0H)	01H
		Timer 2 capture/compare 1 to 4 status register 0 (CCSTATE0)	0000H
		Timer 2 capture/compare 1 to 4 status register 0L (CCSTATE0L)	00H
		Timer 2 capture/compare 1 to 4 status register 0H (CCSTATE0H)	00H
		Timer 2 output delay register 0 (ODELE0)	0000H
		Timer 2 output delay register 0L (ODELE0L)	00H
		Timer 2 output delay register 0H (ODELE0H)	00H
		Timer 2 software event capture register 0 (CSCE0)	0000H
	Timer 3	Timer 3 (TM3)	0000H
		Capture/compare register 3n (CC3n) (n = 0,1)	0000H
		Timer control register 30 (TMC30)	00H
		Timer control register 31 (TMC31)	20H

Table 13-2. Initial Values of CPU, Internal RAM, and On-Chip Peripheral I/O After Reset (4/5)

On-Chip Hardware		Register Name	Initial Value After Reset
On-chip peripheral I/O	Timer 3	Valid edge selection register (SESC)	00H
		Timer 3 clock selection register (PRM03)	00H
		Timer 3 noise elimination time selection register (NRC3)	00H
		Timer 3 output control register (TOC3)	00H
	Timer 4	Timer 4 (TM4)	0000H
		Compare register 4 (CM4)	0000H
		Timer control register 4 (TMC4)	00H
	Serial interface function (CSI0, CSI1)	Clocked serial interface mode register n (CSIMn) (n = 0,1)	00H
		Clocked serial interface clock selection register n (CSICn) (n = 0,1)	00H
		Clocked serial interface receive buffer register n (SIRBn) (n = 0,1)	0000H
		Clocked serial interface receive buffer register Ln (SIRBLn) (n = 0, 1)	00H
		Clocked serial interface transmit buffer register n (SOTBn) (n = 0,1)	0000H
		Clocked serial interface transmit buffer register Ln (SOTBLn) (n = 0, 1)	00H
		Clocked serial interface read-only receive buffer register n (SIRBEn) (n = 0,1)	0000H
		Clocked serial interface read-only receive buffer register Ln (SIRBELn) (n = 0, 1)	00H
		Clocked serial interface first stage transmit buffer register n (SOTBFn) (n = 0,1)	0000H
		Clocked serial interface first stage transmit buffer register Ln (SOTBFLn) (n = 0, 1)	00H
		Serial I/O shift register n (SIOn) (n = 0,1)	0000H
		Serial I/O shift register Ln (SIOLn) (n = 0, 1)	00H
		Prescaler mode register 3 (PRSM3)	00H
		Prescaler compare register 3 (PRSCM3)	00H
	Serial interface function (UART0)	Asynchronous serial interface mode register 0 (ASIM0)	01H
		Receive buffer register 0 (RXB0)	FFH
		Asynchronous serial interface status register 0 (ASIS0)	00H
		Transmit buffer register 0 (TXB0)	FFH
		Asynchronous serial interface transmit status register 0 (ASIF0)	00H
		Baud rate generator control register 0 (BRGC0)	FFH
		Clock selection register 0 (CKSR0)	00H
	Serial interface function (UART1)	Asynchronous serial interface mode register 10 (ASIM10)	81H
		Asynchronous serial interface mode register 11 (ASIM11)	00H
		Asynchronous serial interface status register 1 (ASIS1)	00H
		2-frame consecutive receive buffer register 1 (RXB1)	Undefined
		Receive buffer register L1 (RXBL1)	Undefined
		2-frame consecutive transmit shift register 1 (TXS1)	Undefined
		Transmit shift register L1 (TXSL1)	Undefined

Table 13-2. Initial Values of CPU, Internal RAM, and On-Chip Peripheral I/O After Reset (5/5)

On-Chip Hardware		Register Name	Initial Value After Reset
On-chip peripheral I/O	Serial interface function (UART1)	Prescaler mode register 1 (PRSM1)	00H
		Prescaler compare register 1 (PRSCM1)	00H
	A/D converter	A/D scan mode register n0 (ADSCMn0) (n = 0,1)	0000H
		A/D scan mode register n0L (ADSCMn0L) (n = 0, 1)	00H
		A/D scan mode register n0H (ADSCMn0H) (n = 0, 1)	00H
		A/D scan mode register n1 (ADSCMn1) (n = 0,1)	0000H
		A/D scan mode register n1L (ADSCMn1L) (n = 0, 1)	00H
		A/D scan mode register n1H (ADSCMn1H) (n = 0, 1)	00H
		A/D voltage detection mode register n (ADETMn) (n = 0,1)	0000H
		A/D voltage detection mode register nL (ADETMnL) (n = 0, 1)	00H
		A/D voltage detection mode register nH (ADETMnH) (n = 0, 1)	00H
		A/D conversion result register 0n (ADCR0n) (n = 0 to 5)	0000H
		A/D conversion result register 1n (ADCR1n) (n = 0 to 7)	0000H
		A/D internal trigger selection register n (ITRGn) (n = 0, 1)	00H
	Port function	Ports (P0 to P4, PDH, PCT, PCM)	Undefined
		Port (PDL)	Undefined
		Port (PDLL)	Undefined
		Port (PDLH)	Undefined
		Mode registers (PM1 to PM4, PMDH, PMCT, PMCM)	FFH
		Mode register (PMDL)	FFFFH
		Mode register (PMDLL)	FFH
		Mode register (PMDLH)	FFH
		Mode control registers (PMC1 to PMC4)	00H
		Mode control registers (PMCDH)	00H/FFH
		Mode control register (PMCDL)	0000H/FFFFH
		Mode control register (PMCDLL)	00H/FFH
		Mode control register (PMCDLH)	00H/FFH
		Mode control register (PMCCT)	00H/53H
		Mode control register (PMCCM)	00H/03H
		Function control registers (PFC1, PFC2, PFC3)	00H
	Regulator	Regulator control register (REGC)	00H
★	Flash memory	Flash programming mode control register (FLPMC)	08H/0CH/00H[Note]

★ **Note** μPD703114: 00H
μPD70F3114: 08H or 0CH (For details, refer to **15.7.12 Flash programming mode control register (FLPMC)**.)

Caution In the table above, "Undefined" means either undefined at the time of a power-on reset or undefined due to data destruction when $\overline{\text{RESET}}$ ↓ input and data write timing are synchronized. For a $\overline{\text{RESET}}$ ↓ other than this, data is maintained in its previous status.

CHAPTER 14 REGULATOR

14.1 Features

- Two power supplies, one for the internal CPU and one for the peripheral interface, are not necessary.
- A 5 V single power supply system can be configured by connecting an N-ch transistor (2SD1950 (VL standard product, surface mount type) or 2SD1581 (independent type) is recommended).
- If a 3.3 V power supply is available, it can be directly connected to the REGIN pin.

14.2 Functional Outline

The V850E/IA2 has an internal regulator that can be used to configure a 5 V single power supply system.

To use this regulator, connect an N-ch transistor (2SD1950 (VL standard product, surface mount type) or 2SD1581 (independent type) is recommended) to the REGOUT pin, and the REGIN pin to CV_{SS} via a capacitor for stabilizing the regulator output (refer to **14.3 Connection Example**). If two power supplies (5 V system for the peripheral interface and 3.3 V system for the internal CPU) are available on the system, the regulator can be stopped by the regulator control register (REGC).

The regulator always operates in each operation mode (normal operation, HALT, IDLE, and software STOP mode).

If the 3.3 V power supply is provided separately, setting REGC = 01H suppresses the current consumption (several 10 μA) of the on-chip regulator.

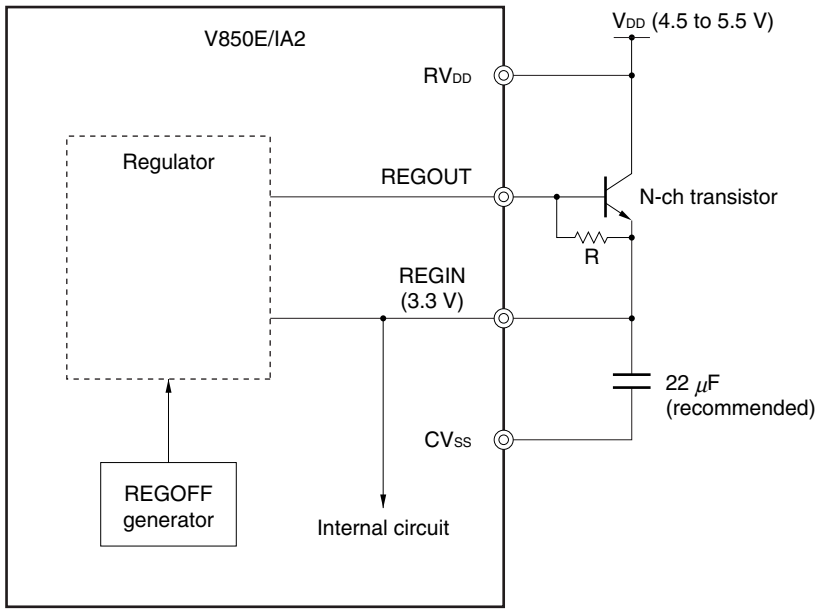
14.3 Connection Example

(1) When using an on-chip regulator

An on-chip regulator is used connected to an N-ch transistor.

An example of connection when using an N-ch transistor and the mount pad dimensions when mounted on the 2SD1950 (VL standard product) (when using a glass epoxy board) are shown below.

Figure 14-1. Example of Connection When Using N-ch Transistor

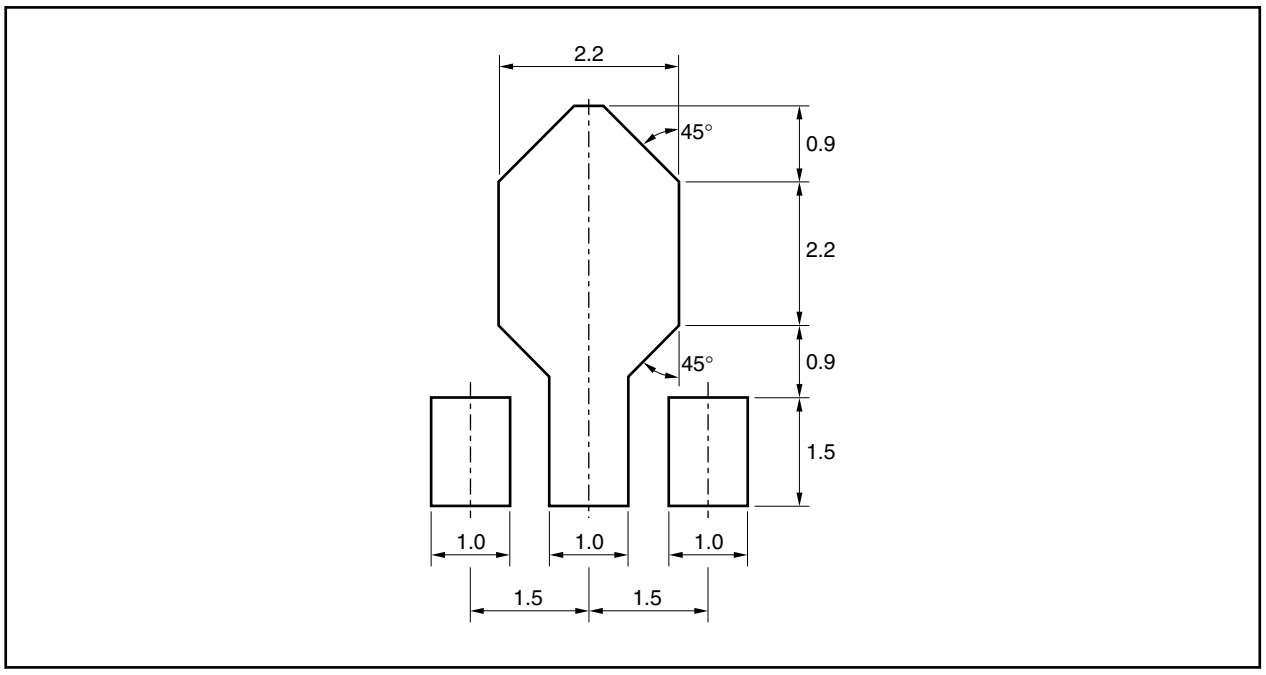


Remark The 2SD1950 (VL standard product, surface mount type) or 2SD1581 (independent type) is recommended as the N-ch transistor.

110 kΩ is recommended for R.

An electrolytic capacitor of 22 μF is recommended.

Figure 14-2. Mount Pad Dimensions When Mounted on 2SD1950 (VL Standard Product) (Glass Epoxy Board) (Unit: mm)

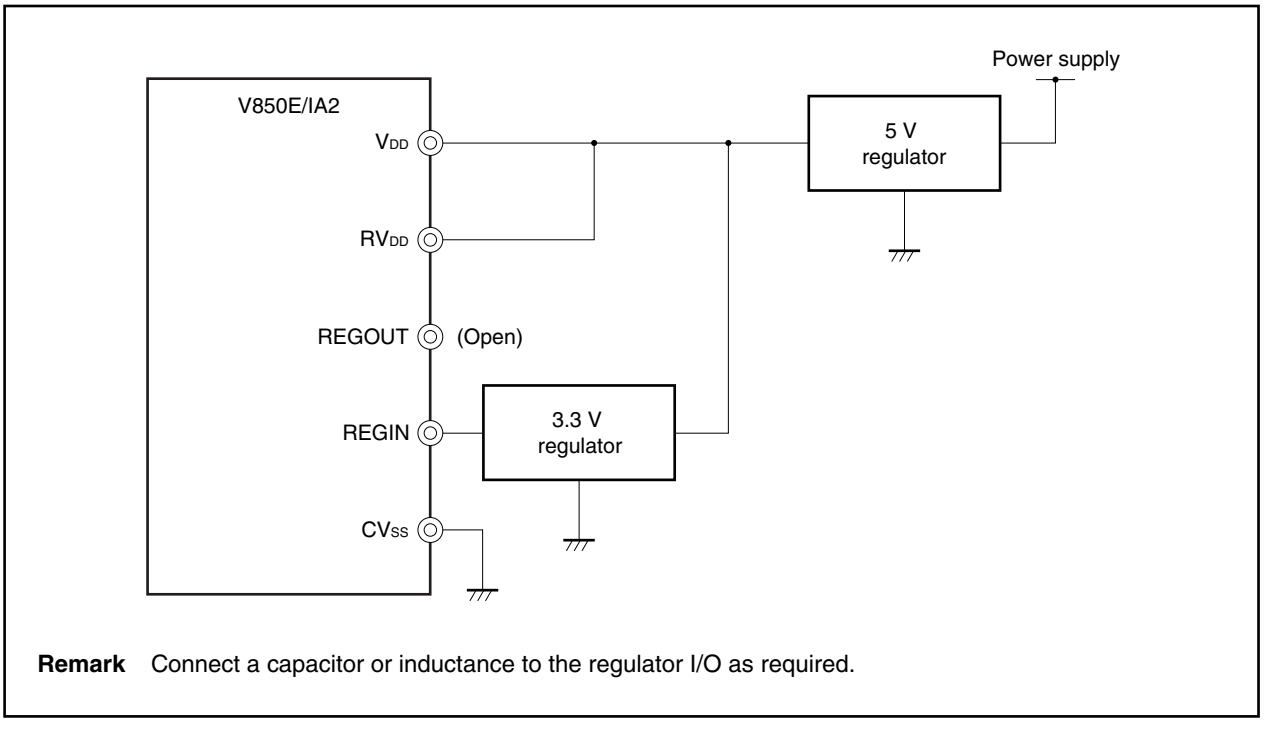


(2) When using an external regulator

When an on-chip regulator is not used, an external regulator can be used.

An example of connection when using an external regulator application is shown below.

Figure 14-3. Connection When Using External Regulator



Remark Connect a capacitor or inductance to the regulator I/O as required.

14.4 Control Register

(1) Regulator control register (REGC)

The REGC register controls the operation of the regulator.

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

Cautions 1. Change the value of the REGC register only once after the system has been reset for system stabilization.

2. Make sure that the pins are set as follows when the REGC0 bit = 1 (when the regulator is stopped).

- **REGOUT pin: Leave open**
- **REGIN pin: Supply 3.3 V (3.0 to 3.6 V) to this pin.**

3. Also make sure that the pins are set as follows when the REGC0 bit = 0 (regulator operating) (for details of the connection method, refer to 14.3 Connection Example).

- **REGOUT pin: Connect this pin to the base of the external transistor.**
- **REGIN pin: Connect this pin to the emitter of the external transistor and to an electrolytic capacitor.**
- **Connect a bias resistor between the base and emitter of the external transistor.**

	7	6	5	4	3	2	1	0	Address	After reset
REGC	0	0	0	0	0	0	0	REGC0	FFFFF300H	00H

Bit position	Bit name	Function
0	REGC0	Controls the operation of the regulator. 0: Regulator operates. 1: Regulator stops.

CHAPTER 15 FLASH MEMORY (μPD70F3114)

The μPD70F3114 is the flash memory version of the V850E/IA2 and has an on-chip 128 KB flash memory.

Caution There are differences in noise immunity and noise radiation between the flash memory and mask ROM versions. When pre-producing an application set with the flash memory version and then mass producing it with the mask ROM version, be sure to conduct sufficient evaluations on the commercial samples (CS) (not engineering samples (ES)) of the mask ROM versions.

Writing to flash memory can be performed with the memory mounted on the target system (on board). A dedicated flash programmer is connected to the target system to perform writing.

The following can be considered as the development environment and the applications of flash memory.

- Software can be changed after the V850E/IA2 is solder-mounted on the target system.
- Small scale production of various models is made easier by differentiating software.
- Data adjustment in starting mass production is made easier.

15.1 Features

- All area batch erase
- Communication via serial interface from the dedicated flash programmer
- Erase/write voltage: V_{PP} = 7.8 V
- On-board programming

15.2 Writing Using Flash Programmer

Writing can be performed either on-board or off-board using a dedicated flash programmer.

Caution When writing flash memory using the flash programmer, be sure to operate the V850E/IA2 at ×5 frequency in PLL mode.

(1) On-board programming

The contents of the flash memory are rewritten after the V850E/IA2 is mounted on the target system. Mount connectors, etc., on the target system to connect the dedicated flash programmer.

(2) Off-board programming

Writing to flash memory is performed by the dedicated program adapter (FA series), etc., before mounting the V850E/IA2 on the target system.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

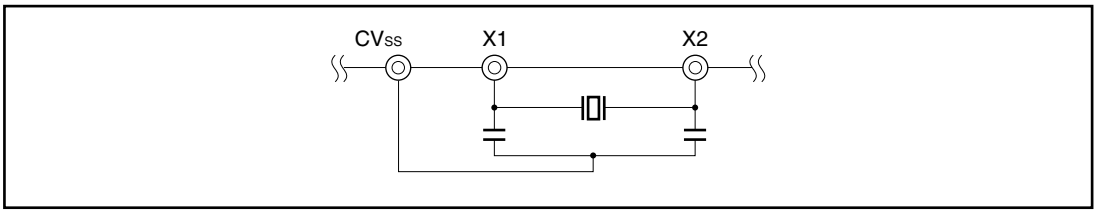
When the flash writing adapter (FA-100GC-8EU) and dual-power-supply adapter (FA-TVC) are used for writing to the μPD70F3114GC, connect the pins as follows.

Table 15-1. Connection of V850E/IA2 Flash Writing Adapter (FA-100GC-8EU)

Name Marked on FA-100GC-8EU PWB	V850E/IA2			
	When UART0 Used		When CSI0 Used	
	Pin Name	Pin No.	Pin Name	Pin No.
SI	TXD0/P31	26	SO0/P41	23
SO	RXD0/P30	25	SI0/P40	22
SCK	–		$\overline{SCK0}$/P42	24
X1	X1	17 Note 1	X1	17 Note 1
X2	X2	18 Note 1	X2	18 Note 1
/RESET	$\overline{RESET}$	19	$\overline{RESET}$	19
V_{PP}	MODE1/V_{PP}	62	MODE1/V_{PP}	62
RESERVE/HS	–		A16/PDH0 Note 2	56
VDD Note 3	V_{DD}	39, 64, 86	V_{DD}	39, 64, 86
	AV_{DD0}	94	AV_{DD0}	94
	AV_{DD1}	2	AV_{DD1}	2
	MODE0	12	MODE0	12
	RV_{DD}	14	RV_{DD}	14
GND Note 3	V_{SS3}	13, 63	V_{SS3}	13, 63
	V_{SS}	38, 87	V_{SS}	38, 87
	AV_{SS0}	95	AV_{SS0}	95
	CV_{SS}	20	CV_{SS}	20
	AV_{SS1}	3	AV_{SS1}	3
	NMI/P00	74	NMI/P00	74
Note 4	CKSEL	21	CKSEL	21

Notes 1. The clock amplitude of X1 and X2 is 3.3 V. Configure the oscillator on the FA-100GC-8EU board using a resonator and a capacitor. The following figure shows an example of the oscillator.

Example



2. Connection is not required for this pin when not using a handshake.

3. Use the dual-power-supply adapter (FA-TVC) for generating 3.3 V on the FA-100GC-8EU board. In this case, the 2SD1950 or 2SD1581 is not required.

4. In PLL mode: GND
In direct mode: V_{DD}

Remark –: Leave open

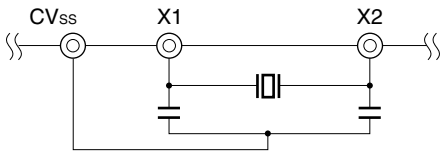
When the flash writing adapter (FA-100GF-3BA) and dual-power-supply adapter (FA-TVC) are used for writing to the μPD70F3114GF, connect the pins as follows.

Table 15-2. Connection of V850E/IA2 Flash Writing Adapter (FA-100GF-3BA)

Name Marked on FA-100GF-3BA PWB	V850E/IA2			
	When UART0 Used		When CSI0 Used	
	Pin Name	Pin No.	Pin Name	Pin No.
SI	TXD0/P31	28	SO0/P41	25
SO	RXD0/P30	27	SI0/P40	24
SCK	–		$\overline{SCK0}$/P42	26
X1	X1	19 Note 1	X1	19 Note 1
X2	X2	20 Note 1	X2	20 Note 1
/RESET	$\overline{RESET}$	21	$\overline{RESET}$	21
V_{PP}	MODE1/V_{PP}	64	MODE1/V_{PP}	64
RESERVE/HS	–		A16/PDH0 Note 2	58
VDD Note 3	V_{DD}	41, 66, 88	V_{DD}	41, 66, 88
	AV_{DD0}	96	AV_{DD0}	96
	AV_{DD1}	4	AV_{DD1}	4
	MODE0	14	MODE0	14
	RV_{DD}	16	RV_{DD}	16
GND Note 3	V_{SS3}	15, 65	V_{SS3}	15, 65
	V_{SS}	40, 89	V_{SS}	40, 89
	AV_{SS0}	97	AV_{SS0}	97
	CV_{SS}	22	CV_{SS}	22
	AV_{SS1}	5	AV_{SS1}	5
	NMI/P00	76	NMI/P00	76
Note 4	CKSEL	23	CKSEL	23

Notes 1. The clock amplitude of X1 and X2 is 3.3 V. Configure the oscillator on the FA-100GF-3BA board using a resonator and a capacitor. The following figure shows an example of the oscillator.

Example



2. Connection is not required for this pin when not using a handshake.

3. Use the dual-power-supply adapter (FA-TVC) for generating 3.3 V on the FA-100GF-3BA board. In this case, the 2SD1950 or 2SD1581 is not required.

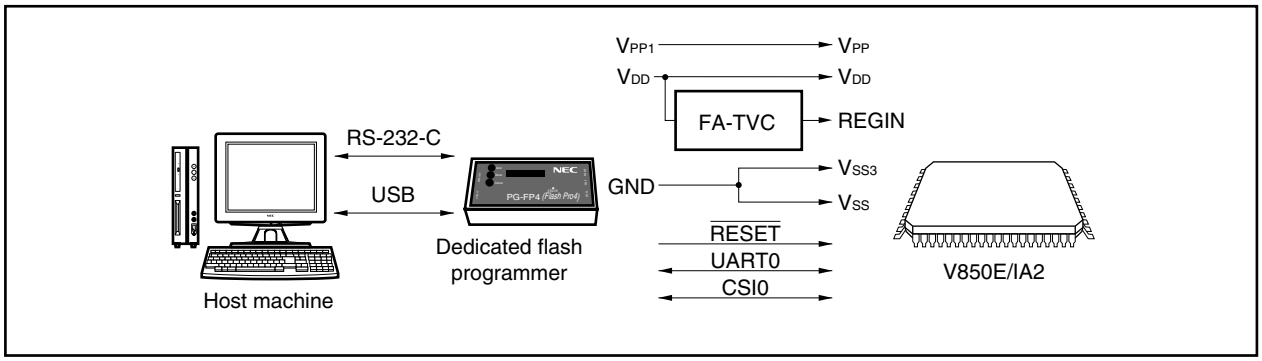
4. In PLL mode: GND
In direct mode: V_{DD}

Remark –: Leave open

15.3 Programming Environment

The following shows the environment required for writing programs to the flash memory of the V850E/IA2.

Figure 15-1. Environment for Writing a Program to Flash Memory



A host machine is required for controlling the dedicated flash programmer.

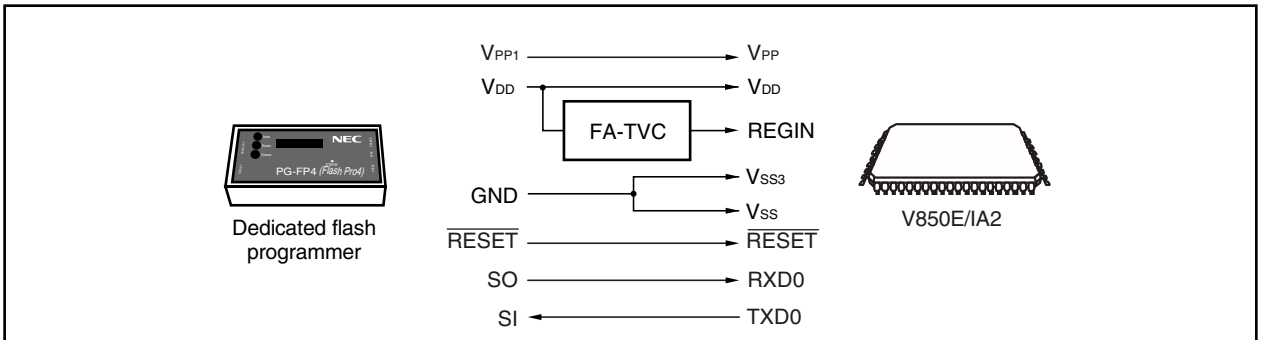
UART0 or CSI0 is used for the interface between the dedicated flash programmer and the V850E/IA2 to perform writing, erasing, etc. A dedicated program adapter (FA series) and dual-power-supply adapter (FA-TVC) are required for off-board writing.

15.4 Communication Mode

(1) UART0

Transfer rate: 4,800 bps to 76,800 bps (LSB first)

Figure 15-2. Communication with Dedicated Flash Programmer (UART0)

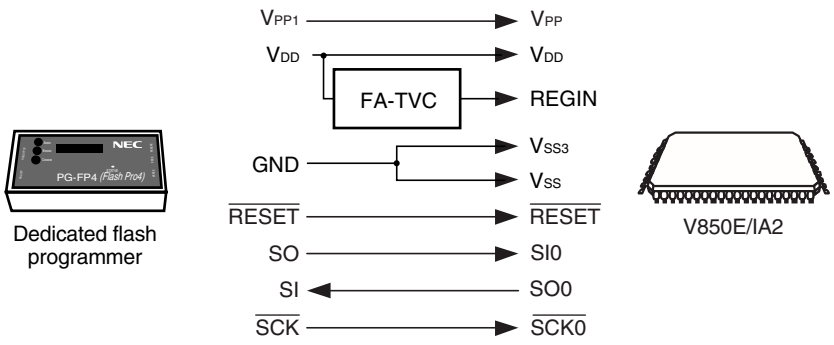


Caution The operating clock amplitude of the V850E/IA2 is 3.3 V.

(2) CSI0

Transfer rate: up to 2 MHz (MSB first)

Figure 15-3. Communication with Dedicated Flash Programmer (CSI0)



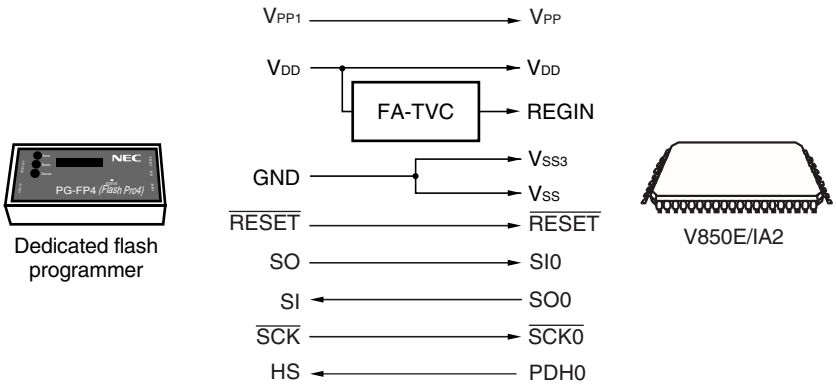
Caution The operating clock amplitude of the V850E/IA2 is 3.3 V.

The dedicated flash programmer outputs transfer clocks and the V850E/IA2 operates as a slave.

(3) Handshake-supported CSI communication

Transfer rate: up to 2 MHz (MSB first)

Figure 15-4. Communication with Dedicated Flash Programmer (Handshake-Supported CSI Communication)



Caution The operating clock amplitude of the V850E/IA2 is 3.3 V.

15.5 Pin Connection

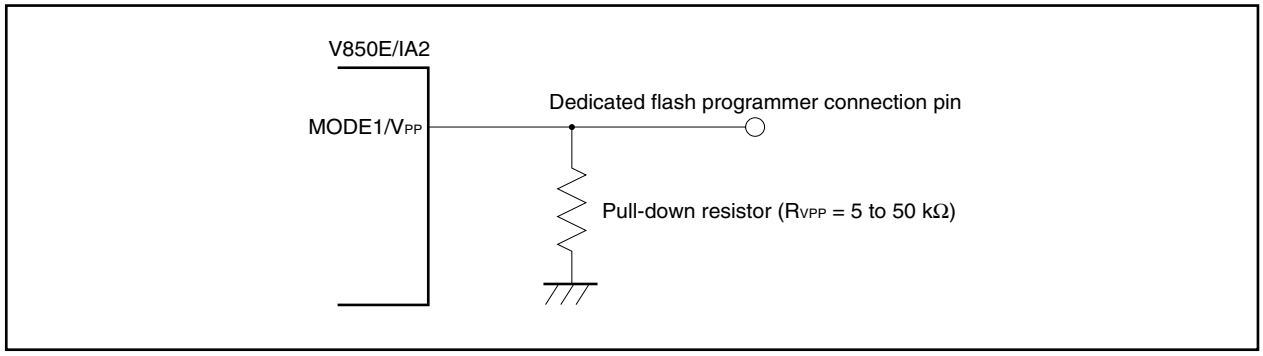
When performing on-board writing, install a connector on the target system to connect to the dedicated flash programmer. Also, install a function on-board to switch from the normal operation mode (single-chip mode or ROMless mode) to the flash memory programming mode.

In the flash memory programming mode, all the pins not used for flash memory programming become the same status as they were immediately after reset in single-chip mode. Therefore, all the ports become output high-impedance status, so that pin connection is required when the external device does not acknowledge the output high-impedance status.

15.5.1 MODE1/V_{PP} pin

In the normal operation mode, 0 V is input to the MODE1/V_{PP} pin. In the flash memory programming mode, 7.8 V writing voltage is supplied to the MODE1/V_{PP} pin. The following shows an example of the connection of the MODE1/V_{PP} pin.

Figure 15-5. Connection Example of MODE1/V_{PP} Pin



15.5.2 Serial interface pin

The following shows the pins used by each serial interface.

Table 15-3. Pins Used by Each Serial Interface

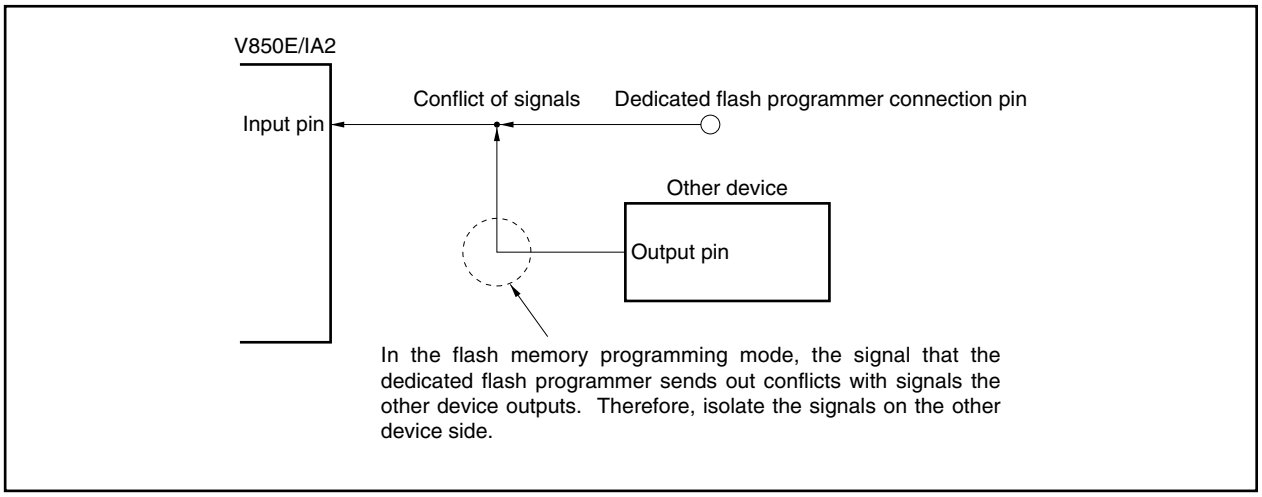
Serial Interface	Pins Used
CSI0	SO0, SI0, $\overline{SCK0}$
CSI0 + HS	SO0, SI0, $\overline{SCK0}$, PDH0
UART0	TXD0, RXD0

When connecting a dedicated flash programmer to a serial interface pin that is connected to other devices on-board, care should be taken to avoid the conflict of signals and the malfunction of other devices, etc.

(1) Conflict of signals

When connecting a dedicated flash programmer (output) to a serial interface pin (input) which is connected to another device (output), a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the other device or set the other device to the output high-impedance status.

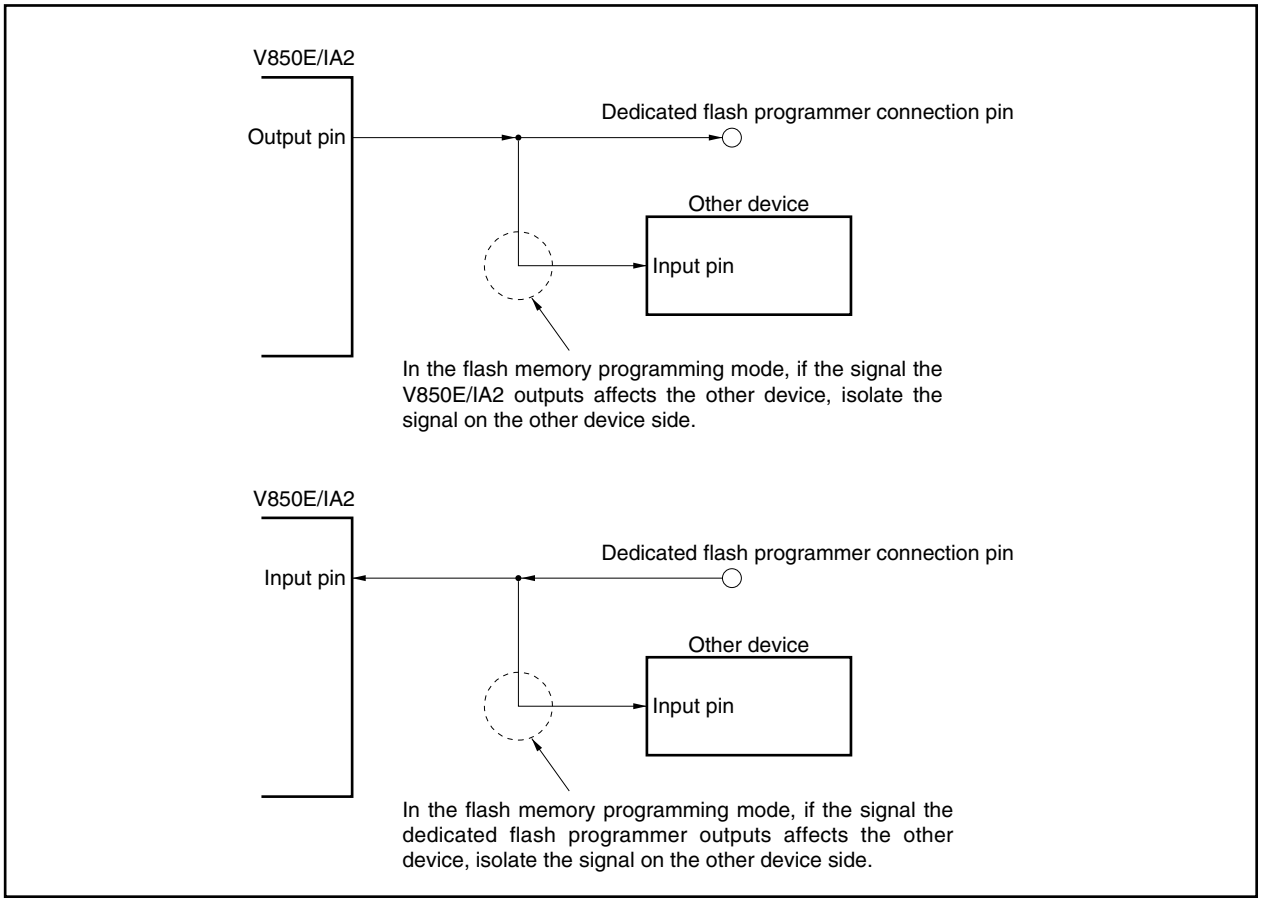
Figure 15-6. Conflict of Signals (Serial Interface Input Pin)



(2) Malfunction of the other device

When connecting a dedicated flash programmer (output or input) to a serial interface pin (input or output) connected to another device (input), the signal output to the other device may cause the device to malfunction. To avoid this, isolate the connection to the other device or make the setting so that the input signal to the other device is ignored.

Figure 15-7. Malfunction of Other Device

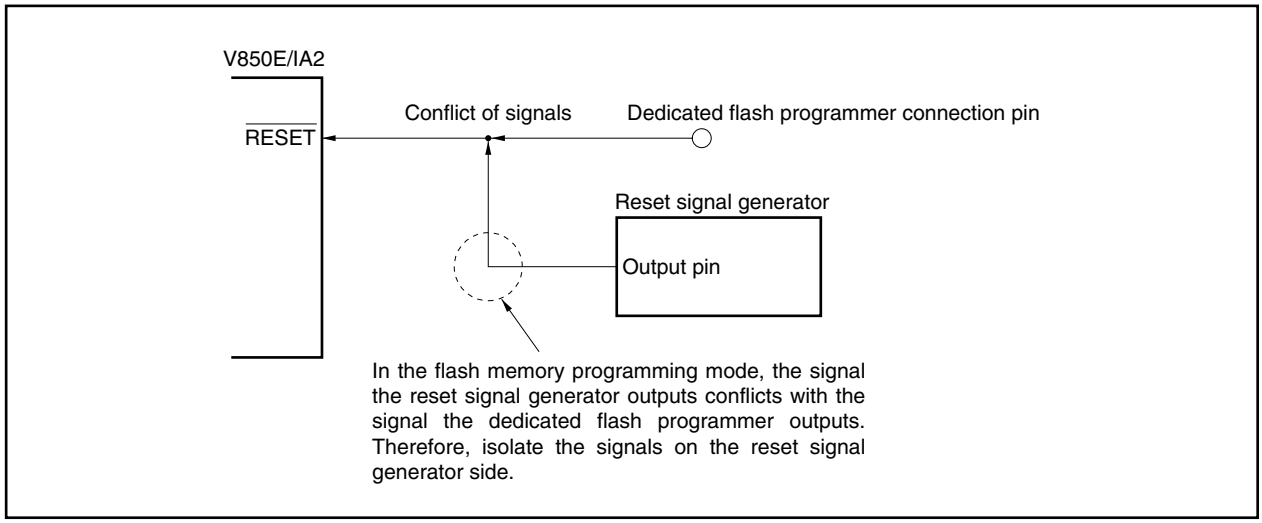


15.5.3 $\overline{\text{RESET}}$ pin

When connecting the reset signals of the dedicated flash programmer to the $\overline{\text{RESET}}$ pin, which is connected, to the reset signal generator on-board, a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the reset signal generator.

When the reset signal is input from the user system in flash memory programming mode, the programming operation will not be performed correctly. Therefore, do not input signals other than the reset signals from the dedicated flash programmer.

Figure 15-8. Conflict of Signals ($\overline{\text{RESET}}$ Pin)



15.5.4 NMI pin

Do not change the input signal to the NMI pin in flash memory programming mode. If it is changed in flash memory programming mode, programming may not be performed correctly.

15.5.5 MODE0, MODE1 pins

To shift to the flash memory programming mode, set MODE0 to high level, apply the writing voltage (7.8 V) to the MODE1/V_{PP} pin, and release reset.

15.5.6 Port pins

When the flash memory programming mode is set, all the port pins except the pins which communicate with the dedicated flash programmer become output high-impedance status. Nothing need be done to these port pins. If problems such as disabling output high-impedance status should occur to the external devices connected to the ports, connect them to V_{DD} or V_{SS} via resistors.

15.5.7 Other signal pins

Connect X1 and X2 to the same status as in the normal operation mode.

The amplitude is 3.3 V.

15.5.8 Power supply

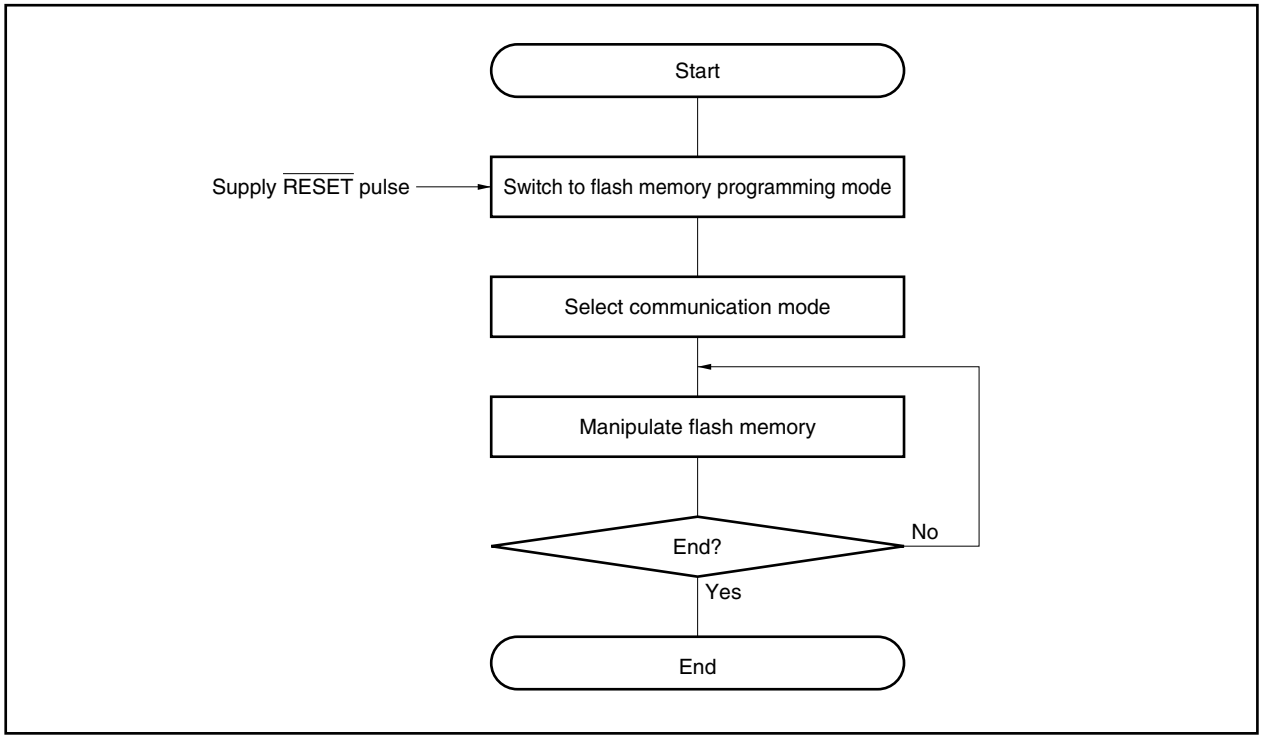
Supply the power supply (V_{DD}, V_{SS}, V_{SS3}, AV_{DD0}, AV_{DD1}, AV_{SS0}, AV_{SS1}, CV_{SS}, RV_{DD}) the same as in normal operation mode. Supply 3.3 V to the REGIN pin from the dual-power-supply adapter (FA-TVC).

★ 15.6 Programming Method

15.6.1 Flash memory control

The following shows the procedure for manipulating the flash memory.

Figure 15-9. Flash Memory Manipulating Procedure



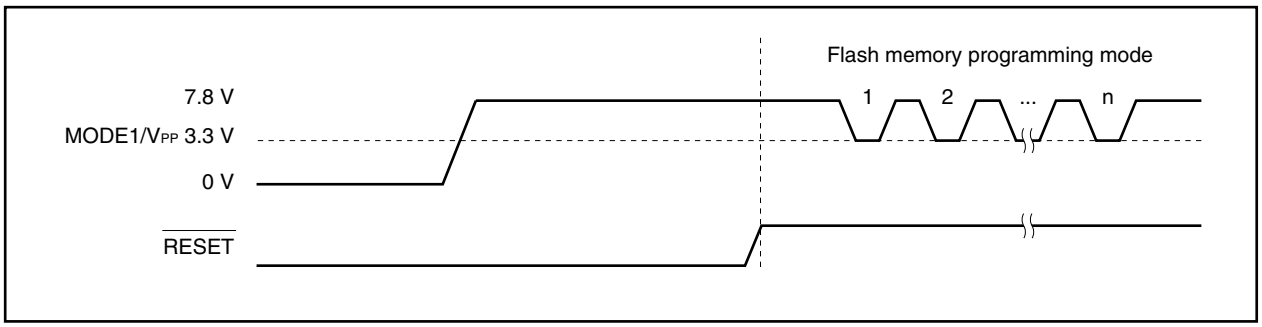
15.6.2 Flash memory programming mode

When rewriting the contents of flash memory using the dedicated flash programmer, set the V850E/IA1 in the flash memory programming mode. To switch to this mode, set the MODE0 and MODE1/V_{PP} pins before canceling reset.

When performing on-board writing, change modes using a jumper, etc.

- MODE0: High-level input
- MODE1/V_{PP}: 7.8 V

Figure 15-10. Flash Memory Programming Mode



15.6.3 Selection of communication mode

In the V850E/IA2, a communication mode is selected by inputting pulses (16 pulses max.) to V_{PP} pin after switching to the flash memory programming mode. The V_{PP} pulse is generated by the dedicated flash programmer.

The following shows the relationship between the number of pulses and the communication mode.

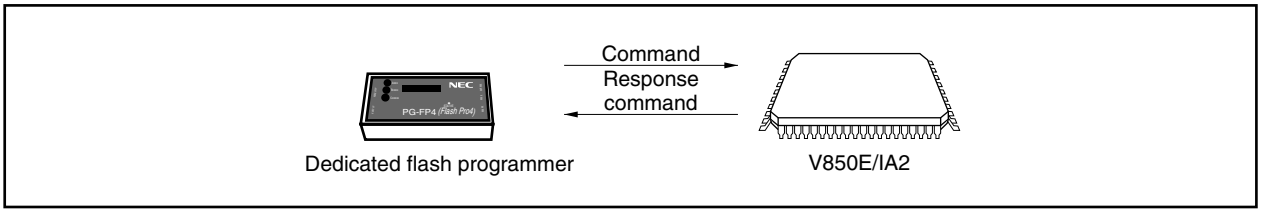
Table 15-4. List of Communication Mode

V_{PP} Pulse	Communication Mode	Remarks
0	CSI0	V850E/IA2 performs slave operation, MSB first
3	Handshake-supported CSI	
8	UART0	Communication rate: 9600 bps (after reset), LSB first
Others	RFU (reserved)	Setting prohibited

15.6.4 Communication commands

The V850E/IA2 communicates with the dedicated flash programmer by means of commands. A command sent from the dedicated flash programmer to the V850E/IA2 is called a “command”. The response signal sent from the V850E/IA2 to the dedicated flash programmer is called the “response command”.

Figure 15-11. Communication Commands



The following shows the commands for controlling flash memory of the V850E/IA2. All of these commands are issued from the dedicated flash programmer, and the V850E/IA2 performs the various processing corresponding to the commands.

Table 15-5. Commands for Controlling Flash Memory

Category	Command Name	Function
Verify	Batch verify command	Compares the contents of the entire memory and the input data.
	Area verify command	Compares the contents of the specified area and the input data.
Erase	Batch erase command	Erases the contents of the entire memory.
	Area erase command	Erases the contents of the specified area.
	Write back command	Writes back the contents which were erased.
Blank check	Batch blank check command	Checks the erase state of the entire memory.
	Area blank check command	Checks the erase state of the specified area.
Data write	High-speed write command	Writes data by the specification of the write address and the number of bytes to be written, and executes verify check.
	Continuous write command	Writes data from the address following the high-speed write command executed immediately before, and executes verify check.
System setting and control	Status read out command	Acquires the status of operations.
	Oscillation frequency setting command	Sets the oscillation frequency.
	Erasing time setting command	Sets the erasing time of batch erase.
	Writing time setting command	Sets the writing time of data write.
	Write back time setting command	Sets the write back time.
	Silicon signature command	Reads outs the silicon signature information.
	Reset command	Escapes from each state.

The V850E/IA2 sends back response commands for the commands issued from the dedicated flash programmer. The following shows the response commands the V850E/IA2 sends out.

Table 15-5. Response Commands

Response Command Name	Function
ACK (acknowledge)	Acknowledges command/data, etc.
NAK (not acknowledge)	Acknowledges illegal command/data, etc.

★ 15.7 Flash Memory Programming by Self-Programming

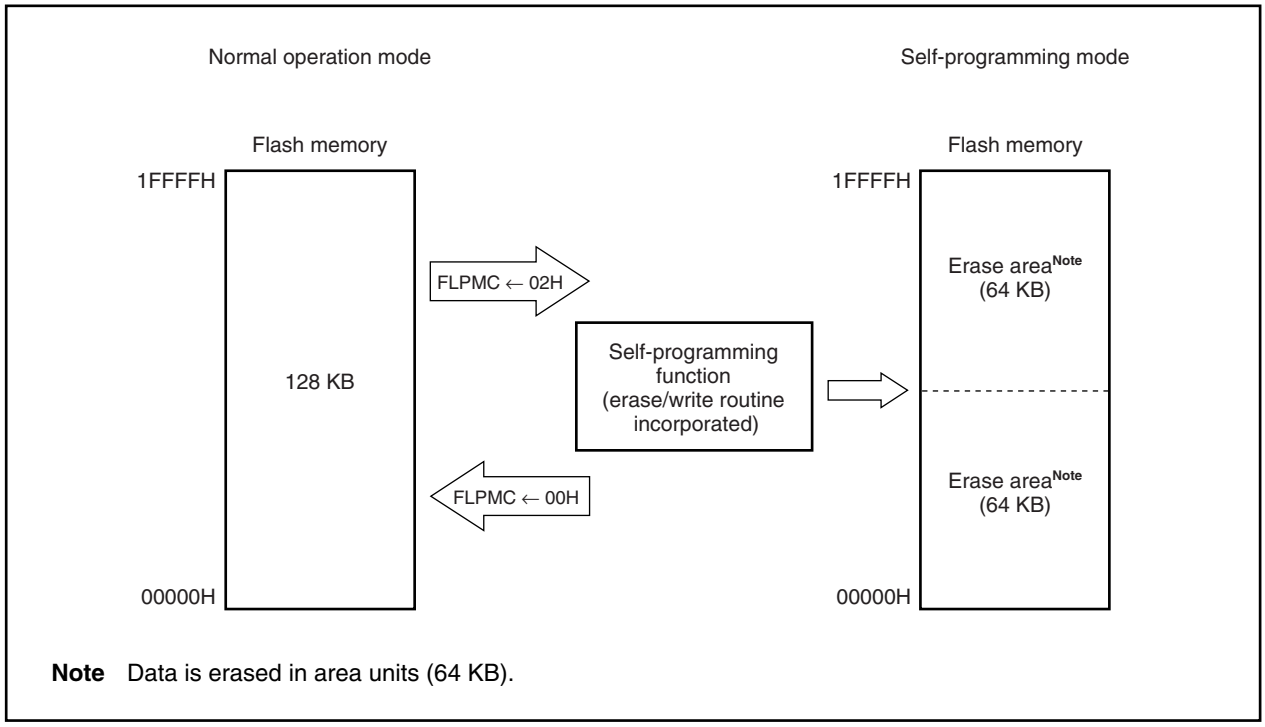
The μPD70F3114 supports a self-programming function to rewrite the flash memory using a user program. By using this function, the flash memory can be rewritten with a user application. This self-programming function can be also used to upgrade the program in the field.

15.7.1 Outline of self-programming

Self-programming implements erasure and writing of the flash memory by calling the self-programming function (device's internal processing) on the program placed in the block 0 space (000000H to 1FFFFFH) and areas other than internal ROM area. To place the program in the block 0 space and internal ROM area, copy the program to areas other than 000000H to 1FFFFFH (e.g. internal RAM area) and execute the program to call the self-programming function.

To call the self-programming function, change the operating mode from normal operation mode to self-programming mode using the flash programming mode control register (FLPMC).

Figure 15-12. Outline of Self-Programming



Note Data is erased in area units (64 KB).

15.7.2 Self-programming function

The μPD70F3114 provides self-programming functions, as shown in Table 15-7. By combining these functions, erasing/writing flash memory becomes possible.

Table 15-7. Function List

Type	Function Name	Function
Erase	Area erase	Erases the specified area.
Write	Continuous write in word units	Continuously writes the specified memory contents from the specified flash memory address, for the number of words specified in 4-byte units.
	Pre-write	Writes 0 to flash memory before erasure.
Check	Erase verify	Checks whether an over erase occurred after erasure.
	Erase byte verify	Checks whether erasure is complete.
	Internal verify	Checks whether the signal level of the post-write data in flash memory is appropriate.
Write back	Area write back	Writes back the flash memory area in which an over erase occurred.
Acquire information	Flash memory information read	Reads out information about flash memory.

15.7.3 Outline of self-programming interface

To execute self-programming using the self-programming interface, the environmental conditions of the hardware and software for manipulating the flash memory must be satisfied.

It is assumed that the self-programming interface is used in an assembly language.

(1) Entry program

This program is to call the internal processing of the device.

It is a part of the application program, and must be executed in memory other than the block 0 space and internal ROM area (flash memory).

(2) Device internal processing

This is manipulation of the flash memory executed inside the device.

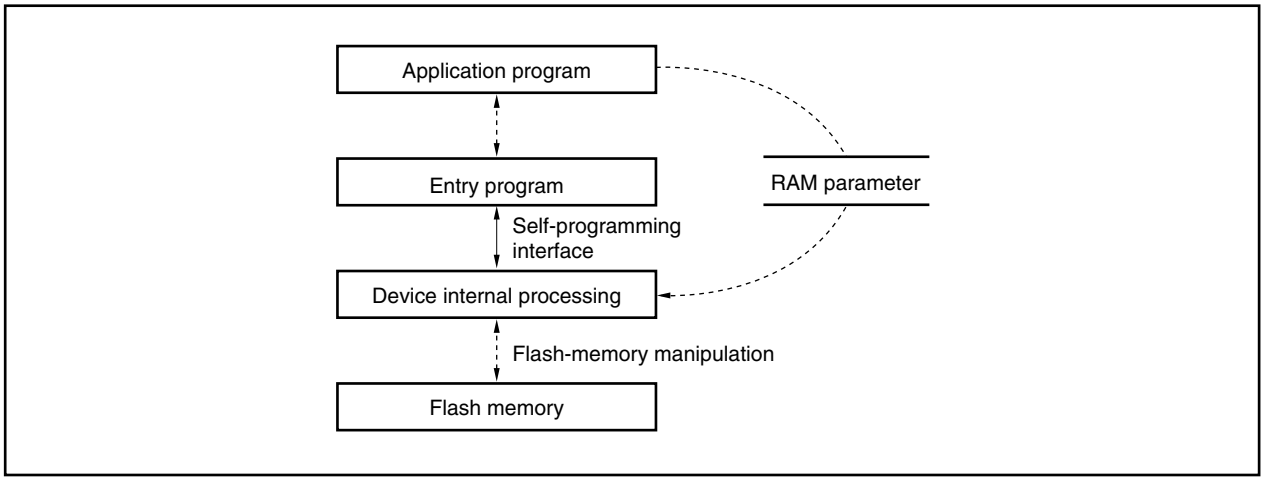
This processing manipulates the flash memory after it has been called by the entry program.

(3) RAM parameter

This is a RAM area to which the parameters necessary for self-programming, such as write time and erase time, are written. It is set by the application program and referenced by the device internal processing.

The self-programming interface is outlined below.

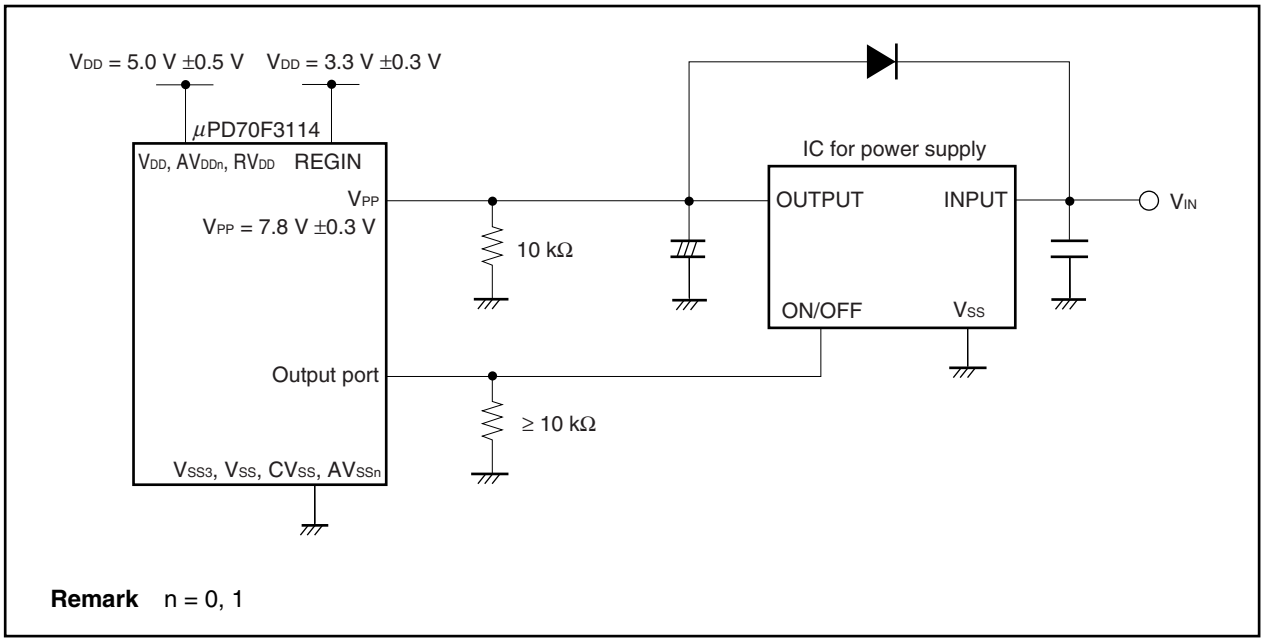
Figure 15-13. Outline of Self-Programming Interface



15.7.4 Hardware environment

To write or erase the flash memory, a high voltage must be applied to the V_{PP} pin. To execute self-programming, a circuit that can generate a write voltage (V_{PP}) and that can be controlled by software is necessary on the application system. An example of a circuit that can select a voltage to be applied to the V_{PP} pin by manipulating a port is shown below.

Figure 15-14. Example of Self-Programming Circuit Configuration



Remark n = 0, 1

The voltage applied to the V_{PP} pin must satisfy the following conditions:

- Hold the voltage applied to the V_{PP} pin at 0 V in the normal operation mode and hold the V_{PP} voltage only while the flash memory is being manipulated.
- The V_{PP} voltage must be stable from before manipulation of the flash memory starts until manipulation is complete.

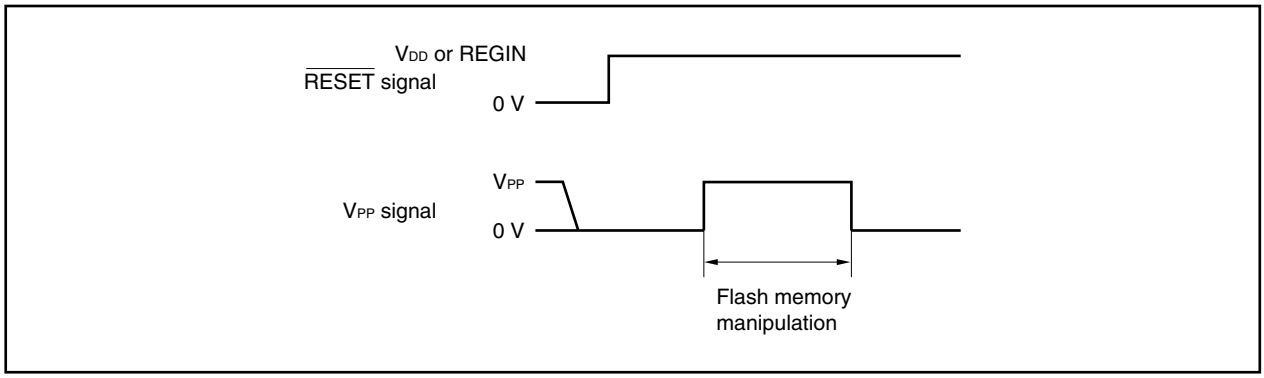
Cautions 1. Apply 0 V to the V_{PP} pin when reset is released.

2. Implement self-programming in single-chip mode 0 or 1.

3. Apply the voltage to the V_{PP} pin in the entry program.

4. If both writing and erasing are executed by using the self-programming function and flash memory programmer on the target board, be sure to communicate with the programmer using CSI0 (do not use the handshake-supported CSI).

Figure 15-15. Timing to Apply Voltage to V_{PP} Pin



15.7.5 Software environment

The following conditions must be satisfied before using the entry program to call the device internal processing.

Table 15-8. Software Environmental Conditions

Item	Description
Location of entry program	Execute the entry program in memory other than the block 0 space and flash memory area. The device internal processing cannot be directly called by the program that is executed on the flash memory.
Execution status of program	The device internal processing cannot be called while an interrupt is being serviced (NP bit of PSW = 0, ID bit of PSW = 1).
Masking interrupts	Mask all the maskable interrupts used. Mask each interrupt by using the corresponding interrupt control register. To mask a maskable interrupt, be sure to specify masking by using the corresponding interrupt control register. Mask the maskable interrupt even when the ID bit of the PSW = 1 (interrupts are disabled).
Manipulation of V_{PP} voltage	Stabilize the voltage applied to the V_{PP} pin (V_{PP} voltage) before starting manipulation of the flash memory. After completion of the manipulation, return the voltage of the V_{PP} pin to 0 V.
Initialization of internal timer	Do not use the internal timer while the flash memory is being manipulated. Because the internal timer is initialized after the flash memory has been used, initialize the timer with the application program to use the timer again.
Stopping reset signal input	Do not input the reset signal while the flash memory is being manipulated. If the reset signal is input while the flash memory is being manipulated, the contents of the flash memory under manipulation become undefined.
Stopping NMI signal input	Do not input the NMI signal while the flash memory is being manipulated. If the NMI signal is input while the flash memory is being manipulated, the flash memory may not be correctly manipulated by the device internal processing. If an NMI occurs while the device internal processing is in progress, the occurrence of the NMI is reflected in the NMI flag of the RAM parameter. If manipulation of the flash memory is affected by the occurrence of the NMI, the function of each self-programming function is reflected in the return value.
Reserving stack area	The device internal processing takes over the stack used by the user program. It is necessary that an area of 300 bytes be reserved for the stack size of the user program when the device internal processing is called. r3 is used as the stack pointer.
Saving general-purpose registers	The device internal processing rewrites the contents of r6 to r14, r20, and r31 (lp). Save and restore these register contents as necessary.

15.7.6 Self-programming function number

To identify a self-programming function, the following numbers are assigned to the respective functions. These function numbers are used as parameters when the device internal processing is called.

Table 15-9. Self-Programming Function Number

Function No.	Function Name
0	Acquiring flash information
1	Erasing area
2 to 4	RFU
5	Area write back
6 to 8	RFU
9	Erase byte verify
10	Erase verify
11 to 15	RFU
16	Continuous write in word units
17 to 19	RFU
20	Pre-write
21	Internal verify
Other	Prohibited

Remark RFU: Reserved for Future Use

15.7.7 Calling parameters

The arguments used to call the self-programming function are shown in the table below. In addition to these arguments, parameters such as the write time and erase time are set to the RAM parameters indicated by ep (r30).

Table 15-10. Calling Parameters

Function Name	First Argument (r6) Function No.	Second Argument (r7)	Third Argument (r8)	Fourth Argument (r9)	Return Value (r10)
Acquiring flash information	0	Option number[Note 1]	–	–	**Note 1**
Erasing area	1	Area erase start address	–	–	0: Normal completion Other than 0: Error
Area write back	5	None (acts on erase manipulation area immediately before)	–	–	None
Erase byte verify	9	Verify start address	Number of bytes to be verified	–	0: Normal completion Other than 0: Error
Erase verify	10	None (acts on erase manipulation area immediately before)	–	–	0: Normal completion Other than 0: Error
Continuous write in word units[Note 2]	16	Write start address[Note 3]	Start address of write source data[Note 3]	Number of words to be written (word units)	0: Normal completion Other than 0: Error
Pre-write	20	Write start address	Number of bytes to be written	–	0: Normal completion Other than 0: Error
Internal verify	21	Verify start address	Number of bytes to be verified	–	0: Normal completion Other than 0: Error

Notes 1. See **15.7.10 Flash information** for details.

2. Prepare write source data in memory other than the flash memory when data is written continuously in word units.

3. This address must be at a 4-byte boundary.

Caution For all the functions, ep (r30) must indicate the first address of the RAM parameter.

15.7.8 Contents of RAM parameters

Reserve the following 48-byte area in the internal RAM or external RAM for the RAM parameters, and set the parameters to be input. Set the base addresses of these parameters to ep (r30).

Table 15-11. Description of RAM Parameter

Address	Size	I/O	Description
ep+0	4 bytes	–	For internal operations
ep+4:Bit 5[Note 1]	1 bit	Input	Operation flag (Be sure to set this flag to 1 before calling the device internal processing.) 0: Normal operation in progress 1: Self-programming in progress
ep+4:Bit 7[Notes 2, 3]	1 bit	Output	NMI flag 0: NMI not detected 1: NMI detected
ep+8	4 bytes	Input	Erase time (unsigned 4 bytes) Expressed as 1 count value in units of the internal operation unit time (100 μs). Set value = Erase time (μs)/internal operation unit time (μs) Example: If erase time is 0.4 s → 0.4 × 1,000,000/100 = 4,000 (integer operation)
ep+0xc	4 bytes	Input	Write back time (unsigned 4 bytes) Expressed as 1 count value in units of the internal operation unit time (100 μs). Set value = Write back time (μs)/internal operation unit time (μs) Example: If write back time is 1 ms → 1 × 1,000/100 = 10 (integer operation)
ep+0x10	2 bytes	Input	Timer set value for creating internal operation unit time (unsigned 2 bytes) Write a set value that makes the value of timer 4 the internal operation unit time (100 μs). Set value = Operating frequency (Hz)/1,000,000 × Internal operation unit time (μs)/ Timer division ratio (4) + 1[Note 4] Example: If the operating frequency is 40 MHz → 40,000,000/1,000,000 × 100/4 + 1 = 1,001 (integer operation)
ep+0x12	2 bytes	Input	Timer set value for creating write time (unsigned 2 bytes) Write a set value that makes the value of timer 4 the write time. Set value = Operating frequency (Hz)/Write time (μs)/Timer division ratio (4) + 1[Note 4] Example: If the operating frequency is 40 MHz and the write time is 20 μs → 40,000,000/1,000,000 × 20/4 + 1 = 201 (integer operation)
ep+0x14	28 bytes	–	For internal operations

Notes 1. Fifth bit of address of ep+4 (least significant bit is bit 0.)

2. Seventh bit of address of ep+4 (least significant bit is bit 0.)

3. Clear the NMI flag by the user program because it is not cleared by the device internal processing.

4. The device internal processing sets this value minus 1 to the timer. Because the fraction is rounded up, add 1 as indicated by the expression of the set value.

Caution Be sure to reserve the RAM parameter area at a 4-byte boundary.

15.7.9 Errors during self-programming

The following errors related to manipulation of the flash memory may occur during self-programming. An error occurs if the return value (r10) of each function is not 0.

Table 15-12. Errors During Self-Programming

Error	Function	Description
Overerase error	Erase verify	Excessive erasure occurs.
Undererase error (blank check error)	Erase byte verify	Erasure is insufficient. Additional erase operation is needed.
Verify error	Continuous write in word units	The written data cannot be correctly read. Either an attempt has been made to write to flash memory that has not been erased, or writing is not sufficient.
Internal verify error	Internal verify	The written data is not at the correct signal level.

Caution The overerase error and undererase error may simultaneously occur in the entire flash memory.

15.7.10 Flash information

For the flash information acquisition function (function No. 0), the option number (r7) to be specified and the contents of the return value (r10) are as follows. To acquire all flash information, call the function as many times as required in accordance with the format shown below.

Table 15-13. Flash Information

Option No. (r7)	Return Value (r10)
0	Specification prohibited
1	Specification prohibited
2	Bit representation of return value (MSB: bit 31) FFFFFFFFFFFFFFFFAAAAAAAAFFFFFFFF (LSB: bit 0) Bits 31 to 16: FFFFFFFFFFFFFFFF (reserved for future use) Mask bits 31 to 16 because they are not normally 0. Bits 15 to 8: AAAAAAAA (number of areas) (unsigned 8 bits) Bits 7 to 0: FFFFFFFF (reserved for future use) Mask bits 7 to 0 because they are not normally 0.
3+0	End address of area 0
3+1	End address of area 1

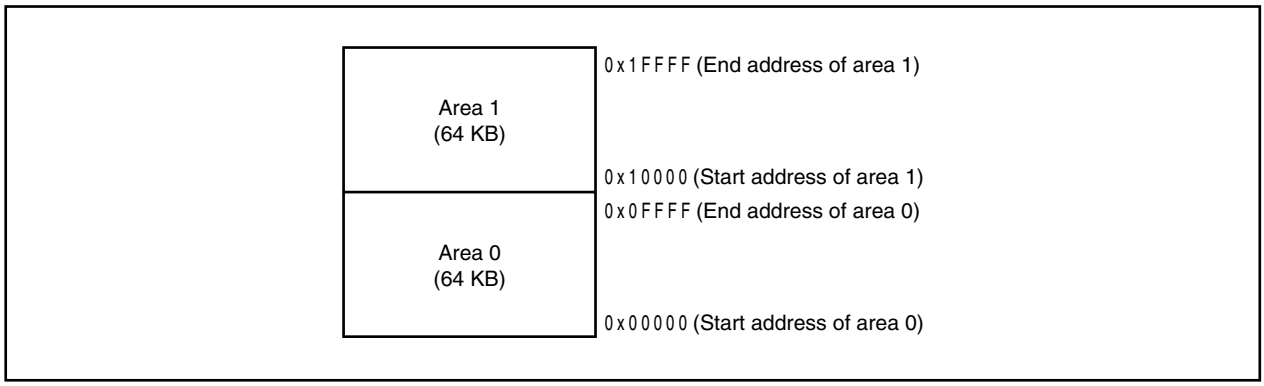
Cautions 1. The start address of area 0 is 0. The "end address + 1" of the preceding area is the start address of the next area.

2. The flash information acquisition function does not check values such as the maximum number of areas specified by the argument of an option. If an illegal value is specified, an undefined value is returned.

15.7.11 Area number

The area numbers and memory map of the μPD70F3114 are shown below.

Figure 15-16. Area Configuration



15.7.12 Flash programming mode control register (FLPMC)

The flash programming mode control register (FLPMC) is a register used to enable/disable writing to flash memory and to specify the self-programming mode.

This register can be read/written in 8-bit or 1-bit units (the VPP bit (bit 2) is read-only).

Cautions 1. Be sure to transfer control to the internal RAM or external memory beforehand to manipulate the FLSPM bit. However, in on-board programming mode set by the flash programmer, the specification of FLSPM bit is ignored.

2. Do not change the initial value of bits 0 and 4 to 7.

	7	6	5	4	<3>	<2>	<1>	0	Address	Initial value[Note]
FLPMC	0	0	0	0	VPPDIS	VPP	FLSPM	0	FFFFF8D4H	08H/0CH/00H

Note 08H: When writing voltage is not applied to the V_{PP} pin
0CH: When writing voltage is applied to the V_{PP} pin
00H: Product not provided with flash memory (μPD703114)

Bit position	Bit name	Function
3	VPPDIS	Enables/disables writing/erasing on-chip flash memory. When this bit is 1, writing/erasing on-chip flash memory is disabled even if a high voltage is applied to the V_{PP} pin. 0: Enables writing/erasing flash memory 1: Disables writing/erasing flash memory
2	VPP	Indicates the voltage applied to the V_{PP} pin reaches the writing-enabled level (read-only). This bit is used to check whether writing is possible or not in the self-programming mode. 0: Indicates high-voltage application to V_{PP} pin is not detected (the voltage has not reached the writing voltage enable level) 1: Indicates high-voltage application to V_{PP} pin is detected (the voltage has reached the writing voltage enable level)
1	FLSPM	Controls switching between internal ROM and the self-programming interface. This bit can switch the mode between the normal mode set by the mode pin on the application system and the self-programming mode. The setting of this bit is valid only if the voltage applied to the V_{PP} pin reaches the writing voltage enable level. 0: Normal mode (for all addresses, instruction fetch is performed from on-chip flash memory) 1: Self-programming mode (device internal processing is started)

Setting data to the flash programming mode control register (FLPMC) is performed in the following sequence.

<1> Disable interrupts (set the NP bit and ID bit of the PSW to 1).
<2> Prepare the data to be set in the specific register in a general-purpose register.
<3> Write data to the peripheral command register (PHCMD).
<4> Set the flash programming mode control register (FLPMC) by executing the following instructions.
- Store instruction (ST/SST instructions)
- Bit manipulation instruction (SET1/CLR1/NOT1 instructions)

<5> Insert NOP instructions (5 instructions (<5> to <9>)).
<10> Cancel the interrupt disabled state (reset the NP bit of the PSW to 0).

[Description example]

```
<1>  LDSR  rX, 5
<2>  MOV   0x02, r10
<3>  ST.B  r10, PHCMD[r0]
<4>  ST.B  r10, FLPMC[r0]
<5>  NOP
<6>  NOP
<7>  NOP
<8>  NOP
<9>  NOP
<10> LDSR  rY, 5
```

Remark rX: Value written to the PSW
rY: Value returned to the PSW

No special sequence is required for reading a specific register.

Cautions 1. If an interrupt is acknowledged between when PHCMD is issued (<3>) and writing to a specific register (<4>) immediately after issuing PHCMD, writing to the specific register may not be performed and a protection error may occur (the PRERR bit of the PHS register = 1). Therefore, set the NP bit of the PSW to 1 (<1>) to disable interrupt acknowledgment. Similarly, disable acknowledgment of interrupts when a bit manipulation instruction is used to set a specific register.

2. Use the same general-purpose register used to set a specific register (<3>) for writing to the PHCMD register (<4>) even though the data written to the PHCMD register is dummy data. This is the same as when a general-purpose register is used for addressing.

3. Before executing this processing, complete all DMA transfer operations.

15.7.13 Calling device internal processing

This section explains the procedure to call the device internal processing from the entry program.

Before calling the device internal processing, make sure that all the conditions of the hardware and software environments are satisfied and that the necessary arguments and RAM parameters have been set. Call the device internal processing by setting the FLSPM bit of the flash programming mode control register (FLPMC) to 1 and then executing the trap 0x1f instruction. The processing is always called using the same procedure. It is assumed that the program of this interface is described in an assembly language.

<1> Set the FLPMC register as follows:
- VPPDIS bit = 0 (to enable writing/erasing flash memory)
- FLSPM bit = 1 (to select self-programming mode)

<2> Clear the NP bit of the PSW to 0 (to enable NMIs (only when NMIs are used on the application)).
<3> Execute trap 0x1f to transfer the control to the device's internal processing.
<4> Set the NP bit and ID bit of the PSW to 1 (to disable all interrupts).
<5> Set the value to the peripheral command register (PHCMD) that is to be set to the FLPMC register.
<6> Set the FLPMC register as follows:
- VPPDIS bit = 1 (to disable writing/erasing flash memory)
- FLSPM bit = 0 (to select normal operation mode)

<7> Wait for the internal manipulation setup time (see **15.7.13 (5) Internal manipulation setup parameter**).

(1) Parameter

r6: First argument (sets a self-programming function number)
r7: Second argument
r8: Third argument
r9: Fourth argument
ep: First address of RAM parameter

(2) Return value

r10: Return value (return value from device internal processing of 4 bytes)
ep+4:Bit 7: NMI flag (flag indicating whether an NMI occurred while the device internal processing was being executed)
0: NMI did not occur while device internal processing was being executed.
1: NMI occurred while device internal processing was being executed.

If an NMI occurs while control is being transferred to the device internal processing, the NMI request may never be reflected. Because the NMI flag is not internally reset, this bit must be cleared before calling the device internal processing. After the control returns from the device internal processing, NMI dummy processing can be executed by checking the status of this flag using software.

(3) Description

Transfer control to the device internal processing specified by a function number using the trap instruction. To do this, the hardware and software environmental conditions must be satisfied. Even if trap 0x1f is used in the user application program, trap 0x1f is treated as another operation after the FLPMC register has been set. Therefore, use of the trap instruction is not restricted on the application.

(4) Program example

An example of a program in which the entry program is executed as a subroutine is shown below. In this example, the return address is saved to the stack and then the device internal processing is called. This program must be located in memory other than the block 0 space and flash memory area.

```
ISETUP     130                              -- Internal manipulation setup parameter
EntryProgram:
   add         -4, sp                       -- Prepare
   st.w        lp, 0[sp]                    -- Save return address
   movea       lo(0x00a0), r0, r10          --
   ldsr        r10, 5                       -- PSW = NP, ID
   mov         lo(0x0002), r10              --
   st.b        r10, PHCMD[r0]               -- PHCMD = 2
   st.b        r10, FLPMC[r0]               -- VPPDIS = 0, FLSPM = 1
   nop
   nop
   nop
   nop
   nop
   movea       lo(0x0020), r0, r10          --
   ldsr        r10, 5                       -- PSW = ID
   trap        0x1f                         -- Device Internal Process
   movea       lo(0x00a0), r0, r6           --
   ldsr        r6, 5                        -- PSW = NP, ID
   mov         lo(0x08), r6
   st.b        r6, PHCMD[r0]                -- PHCMD = 8
   st.b        r6, FLPMC[r0]                -- VPPDIS = 1, FLSPM = 0
   nop
   nop
   nop
   nop
   nop
   mov         ISETUP, lp                   -- loop time = 130
loop:
   divh        r6, r6                       -- To kill time
   add         -1, lp                       -- Decrement counter
   jne         loop                         --
   ld.w        0[sp], lp                    -- Reload lp
   add         4, sp                        -- Dispose
   jmp         [lp]                         -- Return to caller
```

(5) Internal manipulation setup parameter

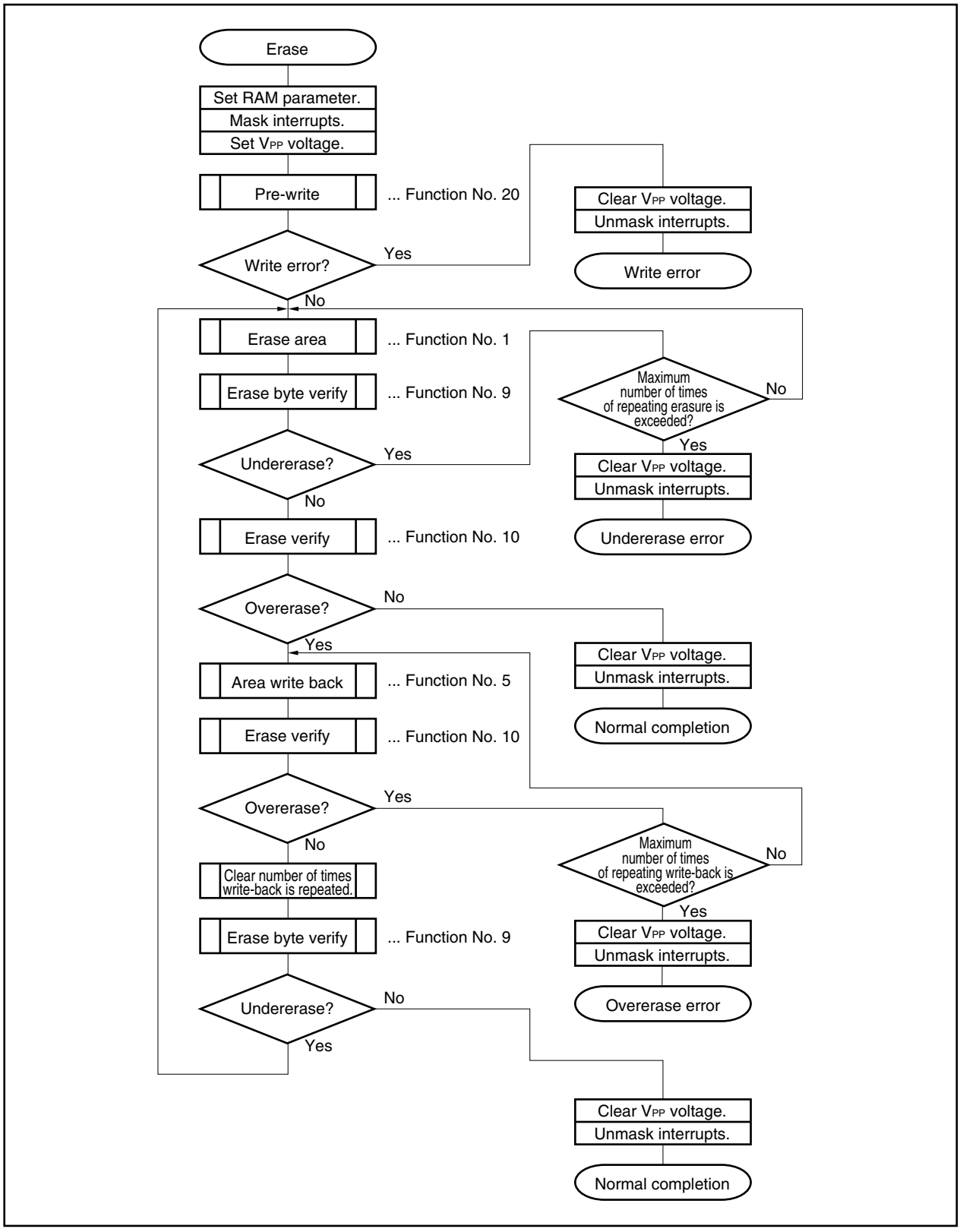
If the self-programming mode is switched to the normal operation mode, the μPD70F3114 must wait for 100 μs before it accesses the flash memory. In the program example in (4) above, the elapse of this wait time is ensured by setting ISETUP to "104" (@ 40 MHz operation). The total number of execution clocks in this example is 39 clocks (divh instruction (35 clocks) + add instruction (1 clock) + jne instruction (3 clocks)). Ensure that a wait time of 100 μs elapses by using the following expression.

39 clocks (total number of execution clocks) × 25 ns (@ 40 MHz operation) × 104 (ISETUP) = 101.4 μs (wait time)

15.7.14 Erasing flash memory flow

The procedure to erase the flash memory is illustrated below. The processing of each function number must be executed in accordance with the specified calling procedure.

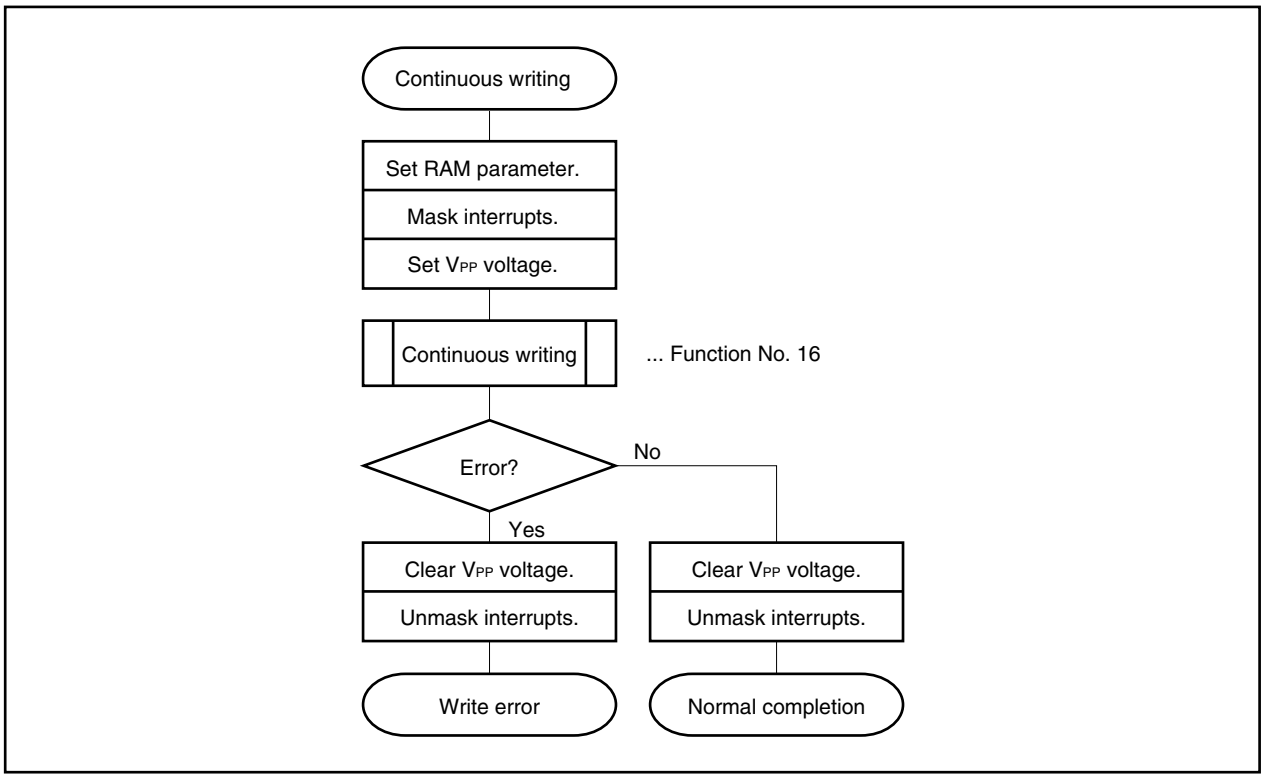
Figure 15-17. Erasing Flash Memory Flow



15.7.15 Continuous writing flow

The procedure to write data all at once to the flash memory by using the function to continuously write data in word units is illustrated below. The processing of each function number must be executed in accordance with the specified calling procedure.

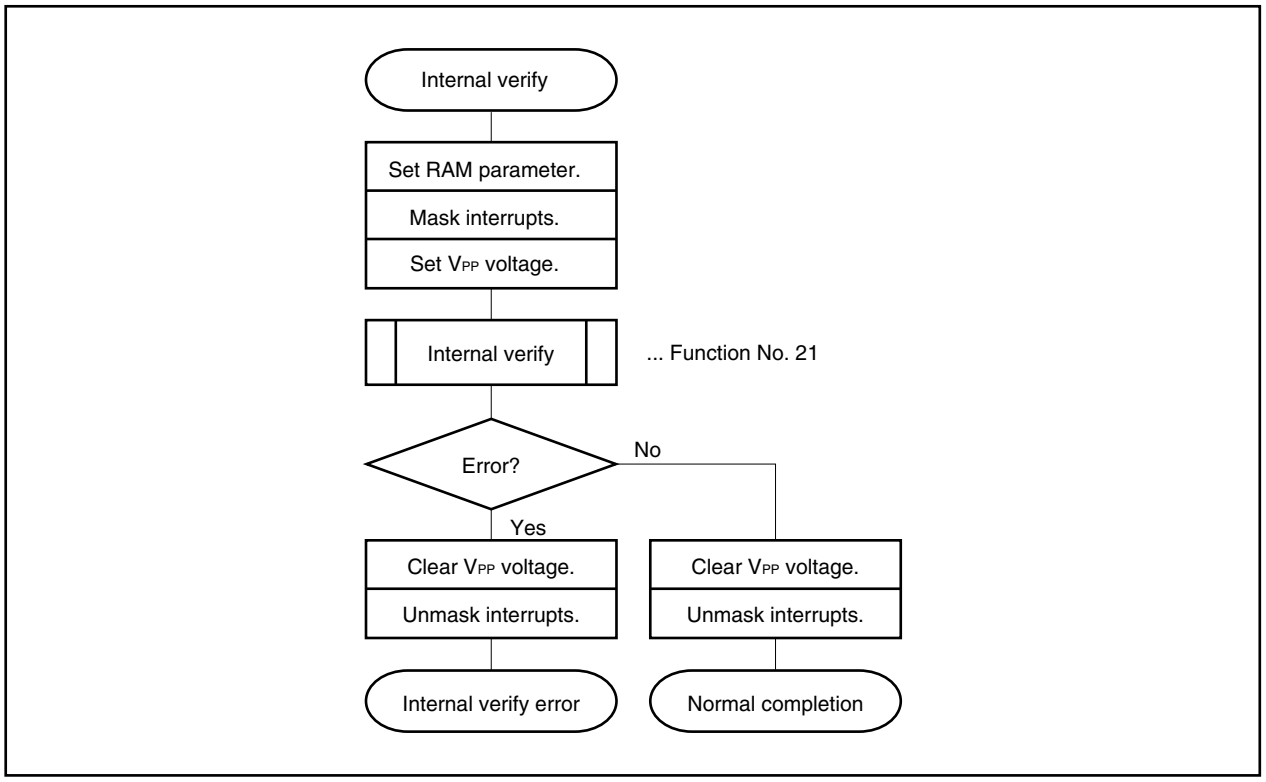
Figure 15-18. Continuous Writing Flow



15.7.16 Internal verify flow

The procedure of internal verification is illustrated below. The processing of each function number must be executed in accordance with the specified calling procedure.

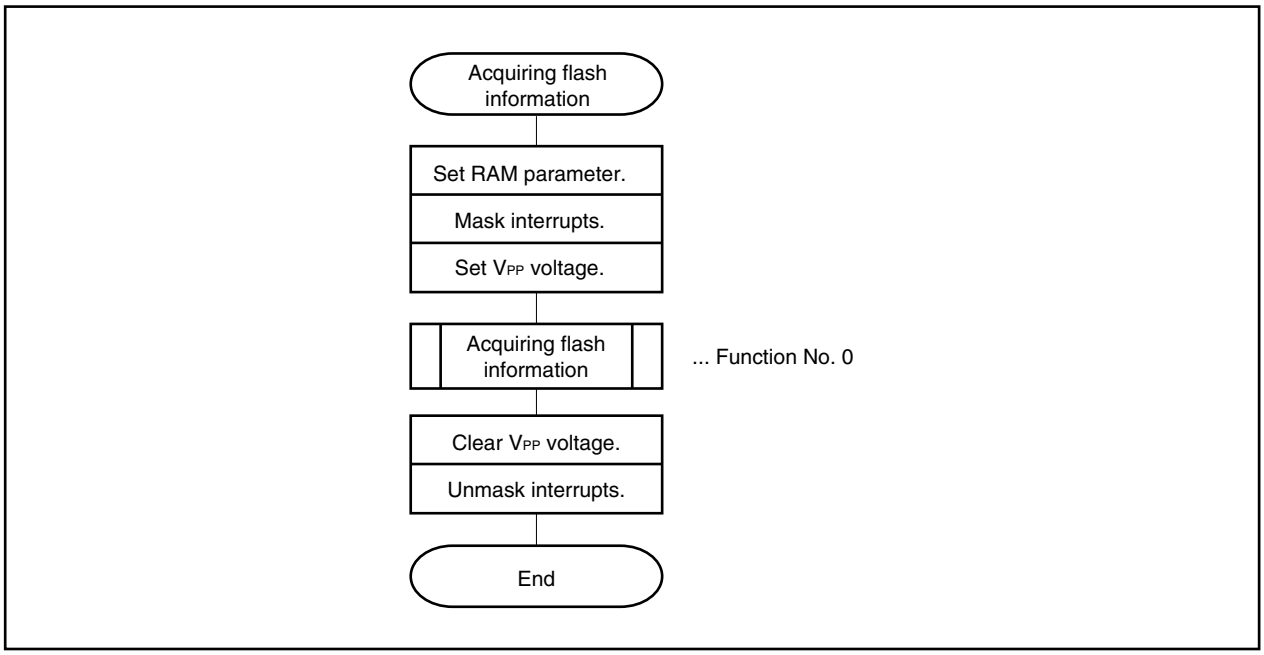
Figure 15-19. Internal Verify Flow



15.7.17 Acquiring flash information flow

The procedure to acquire the flash information is illustrated below. The processing of each function number must be executed in accordance with the specified calling procedure.

Figure 15-20. Acquiring Flash Information Flow



15.7.18 Self-programming library

V850 Series Flash Memory Self-Programming User's Manual is available for reference when executing self-programming.

In this manual, the library uses the self-programming interface of the V850 Series and can be used in C as a utility and as part of the application program. To use the library, thoroughly evaluate it on the application system.

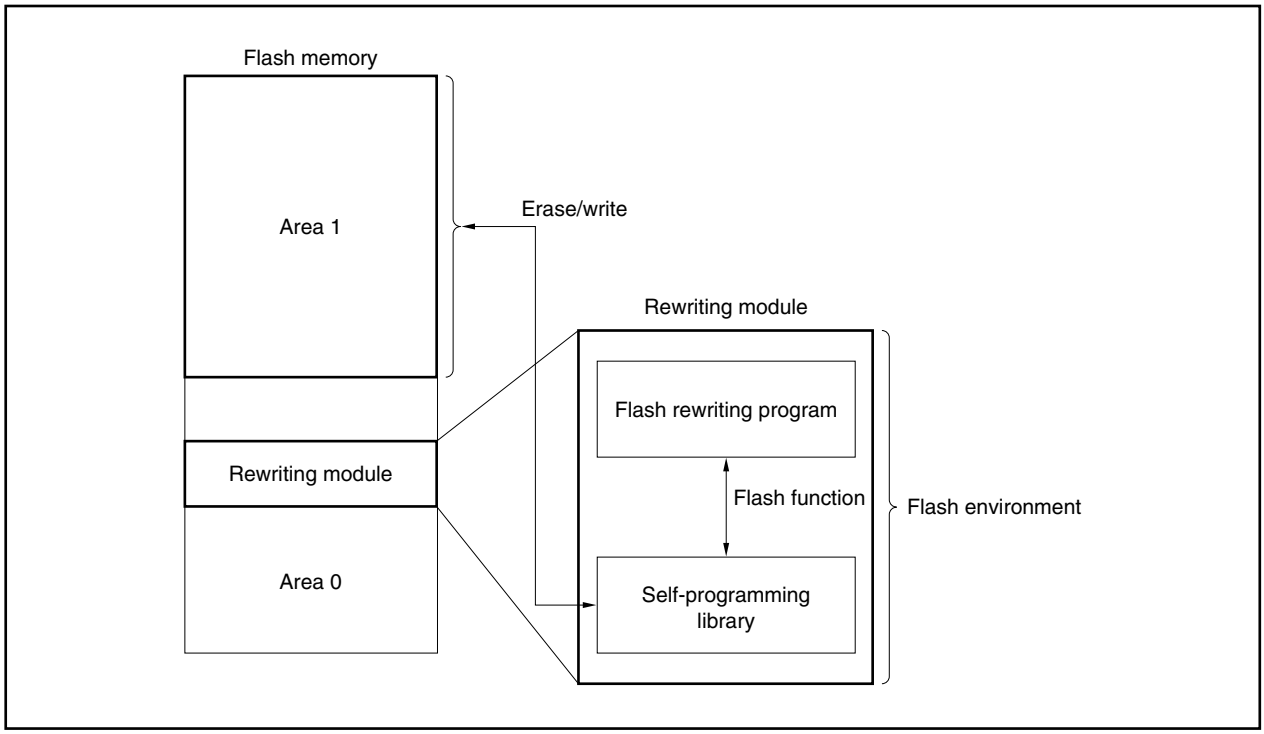
(1) Functional outline

Figure 15-21 outlines the function of the self-programming library. In this figure, a rewriting module is located in area 0 and the data in area 1 is rewritten or erased.

The rewriting module is a user program to rewrite the flash memory. The other areas can be also rewritten by using the flash functions included in this self-programming library. The flash functions expand the entry program in the external memory or internal RAM and call the device internal processing.

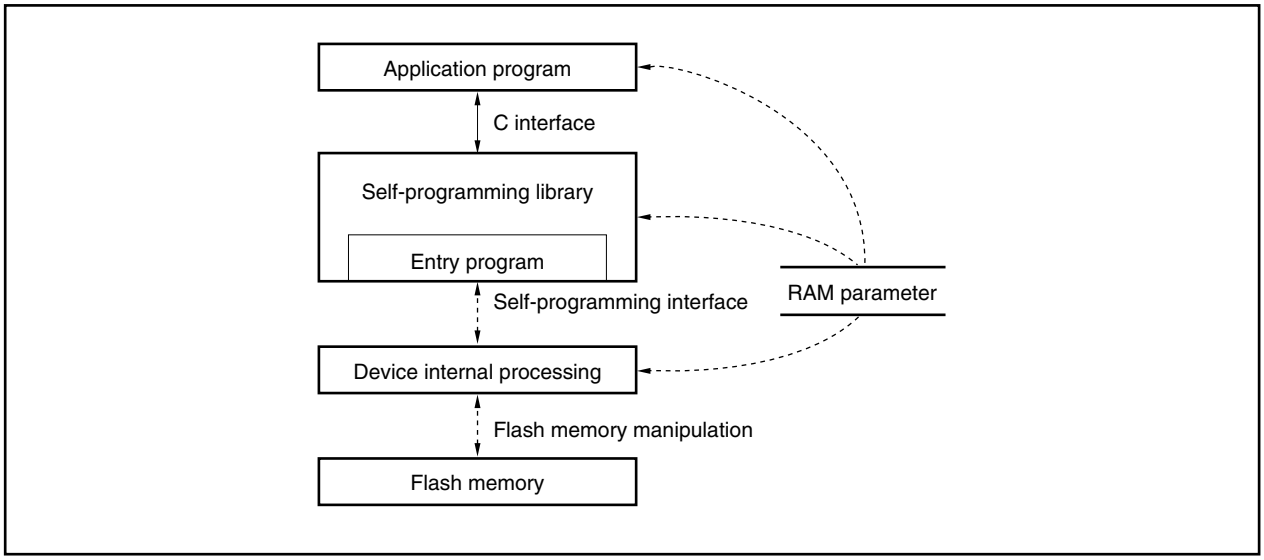
When using the self-programming library, make sure that the hardware conditions, such as the write voltage, and the software conditions, such as interrupts, are satisfied.

Figure 15-21. Functional Outline of Self-Programming Library



The configuration of the self-programming library is outlined below.

Figure 15-22. Outline of Self-Programming Library Configuration



★ 15.8 How to Distinguish Flash Memory and Mask ROM Versions

It is possible to distinguish a flash memory version (μPD70F3114) and a mask ROM version (μPD703114) by means of software, using the methods shown below.

<1> Disable interrupts (set the NP bit of PSW to 1).
<2> Write data to the peripheral command register (PHCMD).
<3> Set the VPPDIS bit of the flash programming mode control register (FLPMC) to 1.
<4> Insert NOP instructions (5 instructions (<4> to <8>)).
<9> Cancel the interrupt disabled state (reset the NP bit of the PSW to 0).
<10> Read the VPPDIS bit of the flash programming mode control register (FLPMC).
- If the value read is 0: Mask ROM version (μPD703114)
- If the value read is 1: Flash memory version (μPD70F3114)

[Description example]

```
<1>  LDSR   rX, 5
<2>  ST.B   r10, PHCMD[r0]
<3>  SET1   3, FLPMC[r0]
<4>  NOP
<5>  NOP
<6>  NOP
<7>  NOP
<8>  NOP
<9>  LDSR   rY, 5
<10> TST1   3, FLPMC[r0]
     BNZ                <Start address of self-programming routine>
     BR                 <Routine when writing is not performed>
```

Remark rX: Value written to the PSW
rY: Value returned to the PSW

Cautions 1. If an interrupt is acknowledged between when PHCMD is issued (<2>) and writing to a specific register (<3>) immediately after issuing PHCMD, writing to a specific register may not be performed and a protection error may occur (the PRERR bit of the PHS register = 1). Therefore, set the NP bit of the PSW to 1 (<1>) to disable interrupt acknowledgment. Similarly, disable acknowledgment of interrupts when a bit manipulation instruction is used to set a specific register.

2. When a store instruction is used for setting a specific register, be sure to use the same general-purpose register used to set the specific register for writing to the PHCMD register even though the data written to the PHCMD register is dummy data. This is the same as when a general-purpose register is used for addressing.

3. Before executing this processing, complete all DMA transfer operations.

CHAPTER 16 ELECTRICAL SPECIFICATIONS

16.1 Normal Operation Mode

Absolute Maximum Ratings (T_A = 25°C)

Parameter	Symbol	Conditions		Ratings	Unit
Power supply voltage	REGIN	REGIN pin		–0.5 to +4.6	V
	V_{DD}	V_{DD} pin		–0.5 to +7.0	V
	RV_{DD}	RV_{DD} pin		–0.5 to +7.0	V
	CV_{SS}	CV_{SS} pin		–0.5 to +0.5	V
	AV_{DD}	AV_{DD0}, AV_{DD1} pins		–0.5 to V_{DD} + 0.5[Note 1]	V
	AV_{SS}	AV_{SS0}, AV_{SS1} pins		–0.5 to +0.5	V
Input voltage	V_{I1}	Other than X1 and V_{PP} pins		–0.5 to V_{DD} + 0.5[Note 1]	V
	V_{I2}	V_{PP} pin (μPD70F3114 only)[Note 2]		–0.5 to +8.5	V
Clock input voltage	V_K	X1 pin		–0.5 to REGIN + 1.0[Note 1]	V
Analog input voltage	V_{IAN}	ANI00 to ANI05 pins, ANI10 to ANI17 pins	$AV_{DD} > V_{DD}$	–0.5 to V_{DD} + 0.5[Note 1]	V
			$V_{DD} \geq AV_{DD}$	–0.5 to AV_{DD} + 0.5[Note 1]	V
Output current, low	I_{OL}	Per pin for the TO000 to TO005 and TO010 to TO015 pins		20	mA
		Per pin other than for the TO000 to TO005 and TO010 to TO015 pins		4.0	mA
		Total for all pins		180	mA
Output current, high	I_{OH}	Per pin		–4.0	mA
		Total for all pins		–100	mA
Operating ambient temperature	T_A			–40 to +85	°C
Storage temperature	T_{stg}			–65 to +150	°C

Notes 1. Be sure not to exceed the absolute maximum ratings (MAX. value) of each power supply voltage.

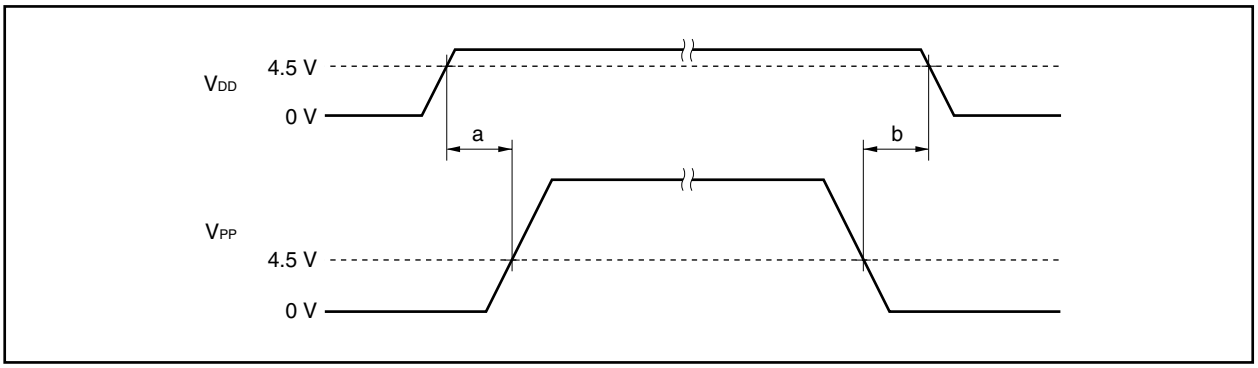
2. Make sure that the following conditions of the V_{PP} voltage application timing are satisfied when the flash memory is written.

- When supply voltage rises

  V_{PP} must exceed V_{DD} 10 μs or more (2 ms when the power supply voltage is stepped down via a regulator) after V_{DD} has reached the lower-limit value (4.5 V) of the operating voltage range (see a in the figure below).

- When supply voltage drops

  V_{DD} must be lowered 10 μs or more after V_{PP} falls below the lower-limit value (4.5 V) of the operating voltage range of V_{DD} (see b in the figure below).



Cautions 1. Do not directly connect 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 shown below for DC characteristics and AC characteristics are within the range for normal operation and quality assurance.

Capacitance (T_A = 25°C, REGIN = V_{DD} = RV_{DD} = V_{SS3} = V_{SS} = CV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	f_C = 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

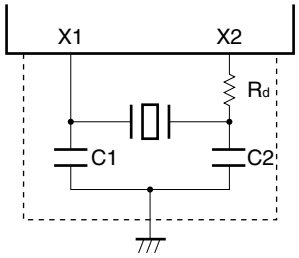
Operation Mode	Internal System Clock Frequency (f_{XX})	Operating Ambient Temperature (T_A)	Power Supply Voltage	
			REGIN	V_{DD} = RV_{DD}
Direct mode	4 to 25 MHz	–40 to +85°C	3.3 V ±0.3 V	5.0 V ±0.5 V
PLL mode	4 to 40 MHz	–40 to +85°C	3.3 V ±0.3 V	5.0 V ±0.5 V

Caution When interfacing to the external devices using the CLKOUT signal, make the internal system clock frequency (f_{XX}) 32 MHz or lower.

Clock Oscillator Characteristics

(T_A = –40 to +85°C, REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V)

(a) Ceramic resonator or crystal resonator connection



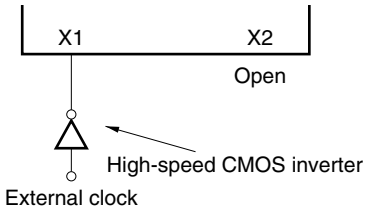
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency	f_X		4		6.4	MHz

Remarks 1. Connect the oscillator as close to the X1 and X2 pins as possible.

2. Do not wire any other signal lines in the area indicated by the broken lines.

3. For the resonator selection and oscillator constant, customers are required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

(b) External clock input



Cautions 1. Connect the high-speed CMOS inverter as close to the X1 pin as possible.

2. Thoroughly evaluate the matching between the V850E/IA2 and the high-speed CMOS inverter.

3. When an internal regulator is used, the external clock must not be used.
This is because a malfunction may occur if the 3.3 V system voltage supplied by the internal regulator and the voltage of the external clock differ in potential.
When using an external clock, do not use the internal regulator and externally supply the REGIN pin with a 3.3 V system voltage of the same potential as the external clock.

Recommended Oscillator Constant

(a) Ceramic resonator

(i) Murata Manufacturing Co., Ltd. (T_A = −40 to +85°C)

Type	Product Name	Oscillation Frequency	Recommended Circuit Constant			Recommended Voltage Range	
		f_X (MHz)	C1 (pF)	C2 (pF)	R_d (Ω)	MIN. (V)	MAX. (V)
Surface mount	CSTCR4M00G55-R0	4.0	On-chip	On-chip	0	3.0	3.6
	CSTCR6M00G55-R0	6.0	On-chip	On-chip	0	3.0	3.6

Caution This oscillator constant is a reference value based on evaluation under a specific environment by the resonator manufacturer.

If optimization of oscillator characteristics is necessary in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.

The oscillation voltage and oscillation frequency indicate only oscillator characteristics. Use the V850E/IA2 so that the internal operating conditions are within the specifications of the DC and AC characteristics.

DC Characteristics (T_A = –40 to +85°C, REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V)

Parameter		Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high		V_{IH1}	Pins for bus control[Note 1]		2.2		V_{DD}	V
		V_{IH2}	Port pins[Note 2]		0.7V_{DD}		V_{DD}	V
		V_{IH3}	Port pins other than **Notes 1, 2,** $\overline{\text{RESET}}$ pin		0.8V_{DD}		V_{DD}	V
		V_{IH4}	X1 pin		0.8REGIN		REGIN + 0.3	V
Input voltage, low		V_{IL1}	Pins for bus control[Note 1]		0		0.8	V
		V_{IL2}	Port pins[Note 2]		0		0.3V_{DD}	V
		V_{IL3}	Port pins other than **Notes 1, 2,** $\overline{\text{RESET}}$ pin		0		0.2V_{DD}	V
		V_{IL4}	X1 pin		–0.5		0.15REGIN	V
Output voltage, high		V_{OH}	I_{OH} = –2.5 mA		V_{DD} – 1.0			V
Output voltage, low		V_{OL1}	PWM output[Note 3]	I_{OL} = 15 mA			2.0	V
				I_{OL} = 2.5 mA			0.4	V
		V_{OL2}	Pins other than **Note 3**	I_{OL} = 2.5 mA			0.4	V
Input leakage current, high		I_{LIH}	V_I = V_{DD}				10	μA
Input leakage current, low		I_{LIL}	V_I = 0 V				–10	μA
Output leakage current, high		I_{LOH}	V_O = V_{DD}				10	μA
Output leakage current, low		I_{LOL}	V_O = 0 V				–10	μA
Analog pin input leakage current		I_{LIAN}	ANI00 to ANI05, ANI10 to ANI17 pins				±10	μA
Power supply current[Note 4]	During normal operation	I_{DD1}	REGIN	**Note 5**, μPD703114		1.8f_{XX} + 15	3.0f_{XX} + 30	mA
				Note 5, μPD70F3114		2.0f_{XX} + 15	3.2f_{XX} + 30	mA
			V_{DD} + RV_{DD}	**Note 6**		30	45	mA
	In HALT mode	I_{DD2}	REGIN	**Note 5**		0.8f_{XX} + 10	1.2f_{XX} + 15	mA
			V_{DD} + RV_{DD}	**Note 6**		15	30	mA
	In IDLE mode	I_{DD3}	REGIN			8	15	mA
			V_{DD} + RV_{DD}	**Note 6**		0.5	1.0	mA
	In STOP mode	I_{DD4}	REGIN	μPD703114		25	300	μA
				μPD70F3114		25	600	μA
			V_{DD} + RV_{DD}	**Note 6**		30	60	μA

Notes 1. AD0/PDL0 to AD15/PDL15, A16/PDH0 to A21/PDH5, $\overline{\text{LWR}}$/PCT0, $\overline{\text{UWR}}$/PCT1, $\overline{\text{RD}}$/PCT4, ASTB/PCT6, $\overline{\text{WAIT}}$/PCM0, CLKOUT/PCM1

2. P31/TXD0, P33/SO1/TXD1, P41/SO0

3. TO000 to TO005, TO010 to TO015

4. Value in the PLL mode

5. Determine the value by calculating f_{XX} from the operating conditions.

6. The current of the TO000 to TO005 and TO010 to TO015 pins is not included.

Remark f_{XX}: Internal system clock frequency

Data Retention Characteristics (T_A = –40 to +85°C)

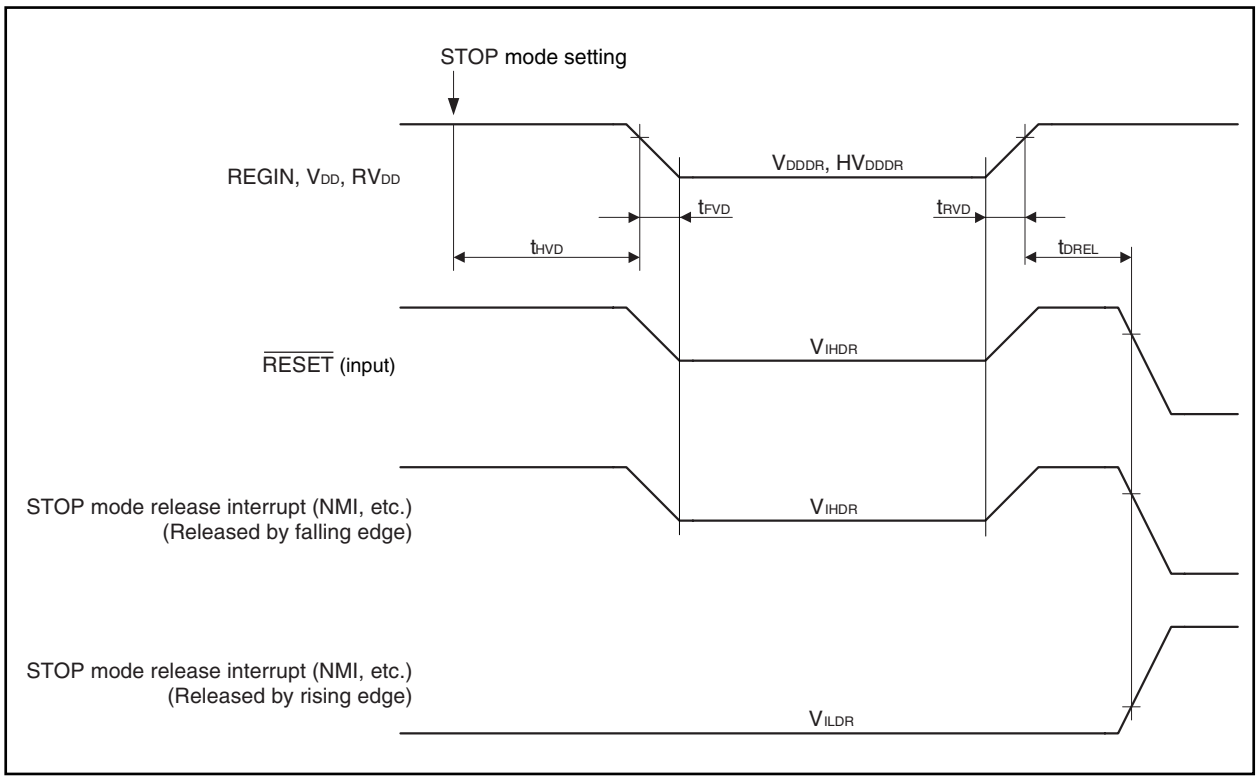
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Data retention voltage	V_{DDDR}	STOP mode, REGIN = V_{DDDR}		1.5		3.6	V
	HV_{DDDR}	STOP mode, V_{DD} = RV_{DD} = HV_{DDDR}		3.6		5.5	V
Data retention current	I_{DDDR}	REGIN = V_{DDDR}	μPD703114		25	300	μA
			μPD70F3114		25	600	μA
	HI_{DDDR}	V_{DD} = RV_{DD} = HV_{DDDR}, **Note 1**			30	60	μA
Power supply voltage rise time	t_{RVD}			200			μs
Power supply voltage fall time	t_{FVD}			200			μs
Power supply voltage retention time (from STOP mode setting)	t_{HVD}			0			ms
STOP release signal input time	t_{DREL}			0			ns
Data retention input voltage, high	V_{IHDR}	**Note 2**		0.8HV_{DDDR}		HV_{DDDR}	V
Data retention input voltage, low	V_{ILDR}	**Note 2**		0		0.2HV_{DDDR}	V

Notes 1. The current of the TO000 to TO005 and TO010 to TO015 pins is not included.

2. P00/NMI, P01/ESO0/INTP0, P02/ESO1/INTP1, P03/ADTRG0/INTP2, P04/ADTRG1/INTP3, P05/INTP4/TO3OFF, P10/TIUD10/TO10, P11/TCUD10/INTP100, P12/TCLR10/INTP101, P20/TI2/INTP20, P21/TO21/INTP21 to P24/TO24/INTP24, P25/TCLR2/INTP25, P26/TI3/TCLR3/INTP30, P27/TO3/INTP31, P30/RXD0, P32/RXD1/SI1, P34/$\overline{\text{ASCK1}}$/$\overline{\text{SCK1}}$, P40/SI0, P42/$\overline{\text{SCK0}}$, MODE0, MODE1, CKSEL, $\overline{\text{RESET}}$

Caution Enter or restore from the STOP mode when REGIN = 3.0 to 3.6 V and V_{DD} = RV_{DD} = 4.5 to 5.5 V.

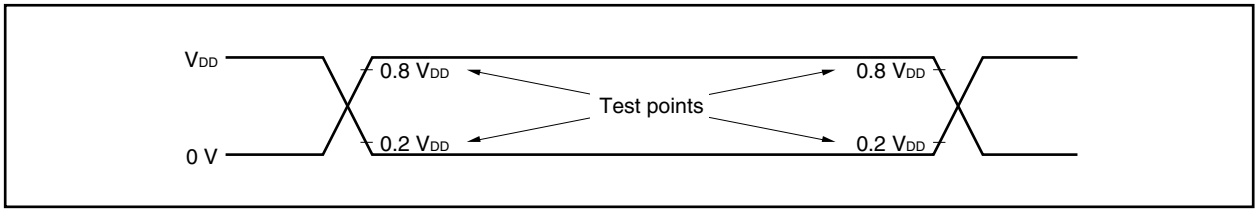
Remark The TYP. value is a reference value for when T_A = 25°C.



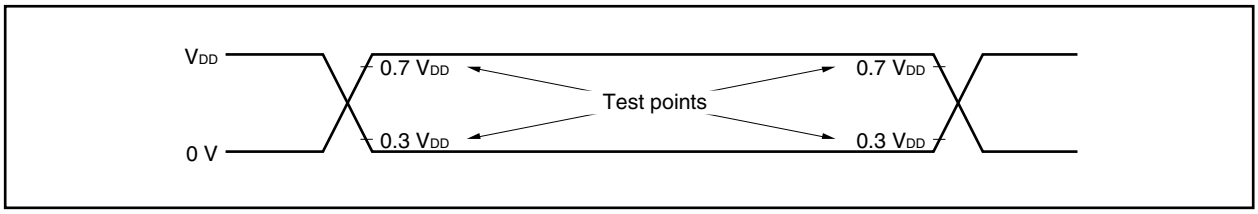
AC Characteristics (T_A = –40 to +85°C, REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V, output pin load capacitance: C_L = 50 pF)

AC test input test points

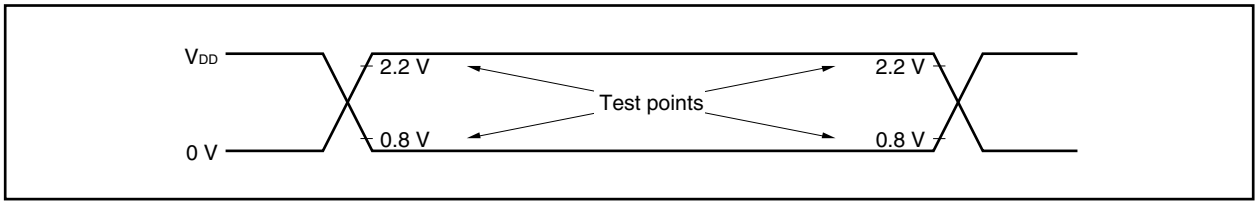
(a) Other than (b), (c), and (d) below



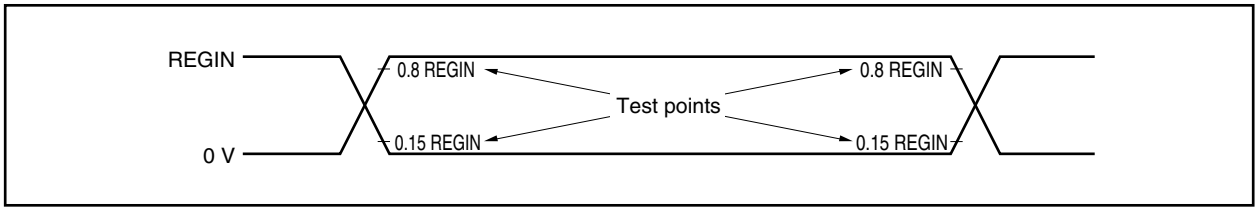
(b) P31/TXD0, P33/SO1/TXD1, P41/SO0



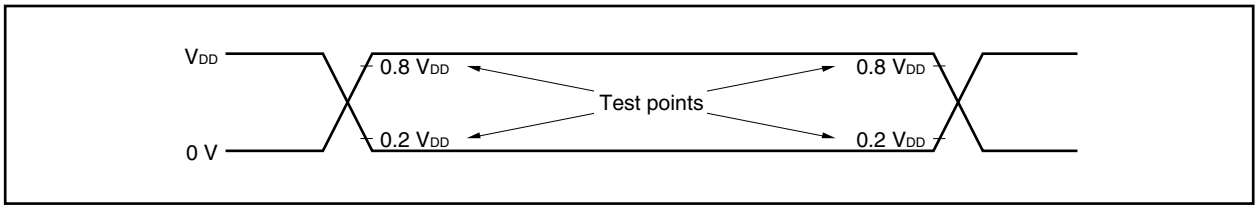
(c) AD0/PDL0 to AD15/PDL15, A16/PDH0 to A21/PDH5, $\overline{\text{LWR}}$/PCT0, $\overline{\text{UWR}}$/PCT1, $\overline{\text{RD}}$/PCT4, ASTB/PCT6, $\overline{\text{WAIT}}$/PCM0, CLKOUT/PCM1



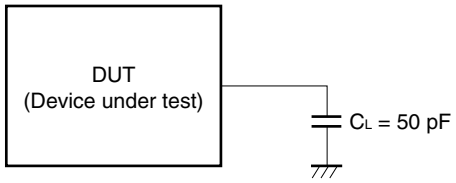
(d) X1



AC test output test points



Load condition



Caution In cases where the load capacitance is greater than 50 pF due to the circuit configuration, insert a buffer or other element to reduce the device's load capacitance to 50 pF or lower.

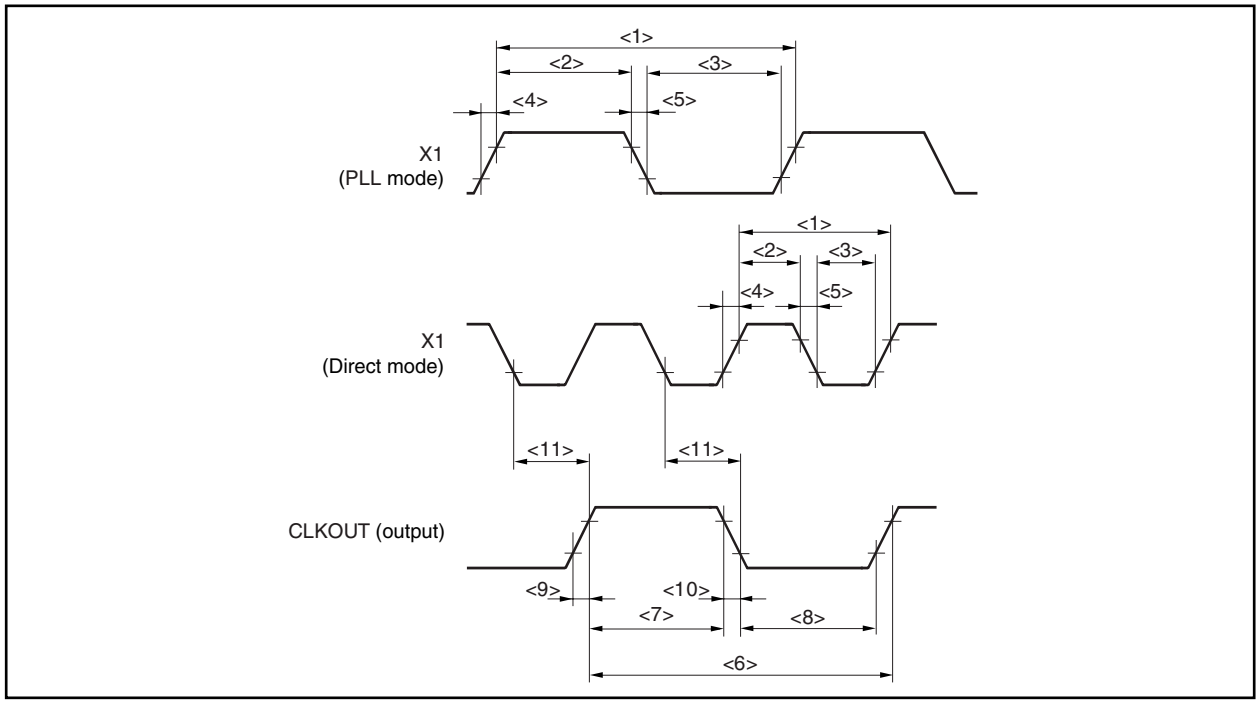
(1) Clock timing

(T_A = –40 to +85°C, REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V, output pin load capacitance: C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
X1 input cycle	t_{CYX}	<1>	Direct mode	20	125	ns
			PLL mode	156	250	ns
X1 input high-level width	t_{WXH}	<2>	Direct mode	6		ns
			PLL mode	50		ns
X1 input low-level width	t_{WXL}	<3>	Direct mode	6		ns
			PLL mode	50		ns
X1 input rise time	t_{XR}	<4>	Direct mode		4	ns
			PLL mode		10	ns
X1 input fall time	t_{XF}	<5>	Direct mode		4	ns
			PLL mode		10	ns
CPU operation frequency	f_{XX}	–		4	40	MHz
			CLKOUT signal used[Note]	4	32	MHz
CLKOUT output cycle	t_{CYK}	<6>		25	250	ns
			CLKOUT signal used[Note]	31.25	250	ns
CLKOUT high-level width	t_{WKH}	<7>		0.5T – 9		ns
CLKOUT low-level width	t_{WKL}	<8>		0.5T – 11		ns
CLKOUT rise time	t_{KR}	<9>			11	ns
CLKOUT fall time	t_{KF}	<10>			9	ns
Delay time from X1↓ to CLKOUT	t_{DXK}	<11>	Direct mode		40	ns

Note When interfacing to the external devices using the CLKOUT signal, make the internal system clock frequency (f_{XX}) 32 MHz or lower.

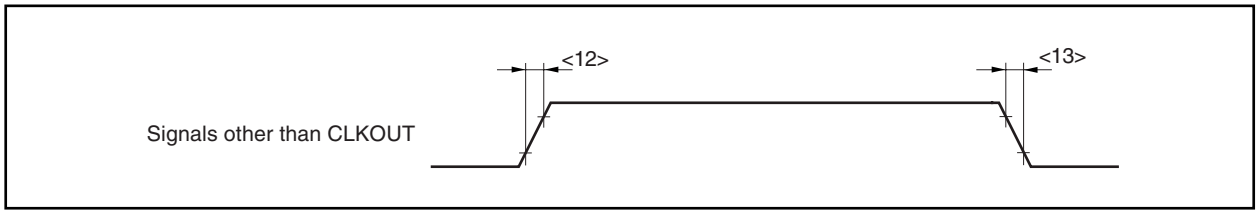
Remark T = t_{CYK}



(2) Output waveform (except for CLKOUT)

(T_A = –40 to +85°C, REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V, output pin load capacitance: C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output rise time	t_{OR}	<12>			15	ns
Output fall time	t_{OF}	<13>			15	ns

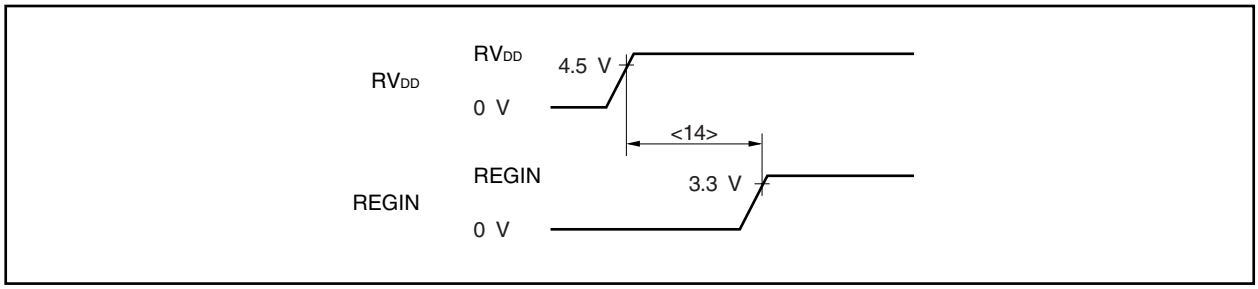


(3) Regulator output stabilization time

(T_A = –40 to +85°C, REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Regulator output stabilization time	t_{RG}	<14>	External NPN transistor: 2SD1950 (VL compliant product) or 2SD1581 Stabilization capacitance: C = 22 μF (electrolytic capacitor) Bias resistance between B and E: R = 110 kΩ	2		ms

Caution The regulator output stabilization time (t_{RG}) varies depending on the external transistor, stabilization capacitance, and bias resistance between B and E.



(4) Reset timing

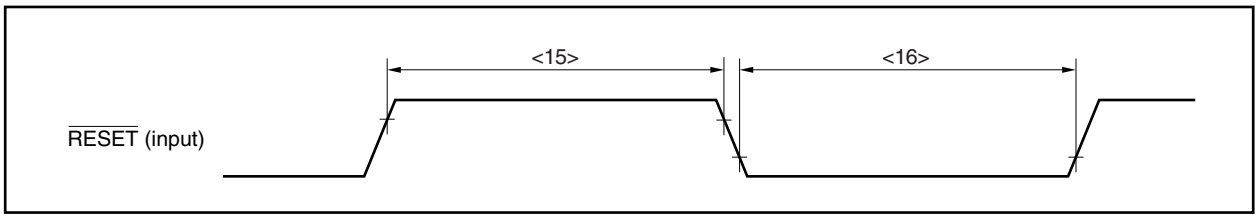
(T_A = –40 to +85°C, REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ pin high-level width	t_{WRSH}	<15>		500		ns
$\overline{\text{RESET}}$ pin low-level width	t_{WRSL}	<16>	At power-on	500 + T_{OS} + t_{RG}		ns
			At STOP mode release[Note]	500 + T_{OS}		ns
			Other than at power-on and at STOP mode release	500		ns

Note Release the STOP mode in the range of REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V.

Caution Thoroughly evaluate the oscillation stabilization time.

Remark T_{OS}: Oscillation stabilization time
t_{RG}: Regulator output stabilization time



(5) Multiplexed bus timing

(a) CLKOUT asynchronous (T_A = –40 to +85°C, REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V, output pin load capacitance: C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address setup time (to ASTB↓)	t_{SAST}	<17>		$(0.5 + w_{AS})T - 16$		ns
Address hold time (from ASTB↓)	t_{HSTA}	<18>		$(0.5 + w_{AH})T - 15$		ns
Address float delay time from $\overline{RD}$↓	t_{FRDA}	<19>			11	ns
Data input setup time from address	t_{SAID}	<20>			$(2 + w + w_{AS} + w_{AH})T - 40$	ns
Data input setup time from $\overline{RD}$↓	t_{SRDID}	<21>			$(1 + w)T - 40$	ns
Delay time from ASTB↓ to $\overline{RD}$, $\overline{LWR}$, $\overline{UWR}$↓	$t_{DSTRDWR}$	<22>		$(0.5 + w_{AH})T - 15$		ns
Data input hold time (from $\overline{RD}$↑)	t_{HRDID}	<23>		0		ns
Address output time from $\overline{RD}$↑	t_{DRDA}	<24>		$(1 + i)T - 15$		ns
Delay time from $\overline{RD}$, $\overline{LWR}$, $\overline{UWR}$↑ to ASTB↑	$t_{DRDWRST}$	<25>		$0.5T - 15$		ns
Delay time from $\overline{RD}$↑ to ASTB↓	t_{DRDST}	<26>		$(1.5 + i + w_{AS})T - 15$		ns
$\overline{RD}$, $\overline{LWR}$, $\overline{UWR}$ low-level width	t_{WRDWRL}	<27>		$(1 + w)T - 22$		ns
ASTB high-level width	t_{WSTH}	<28>		$(1 + w_{AS})T - 15$		ns
Data output time from $\overline{LWR}$, $\overline{UWR}$↓	t_{DWROD}	<29>			10	ns
Data output setup time (to $\overline{LWR}$, $\overline{UWR}$↑)	t_{SODWR}	<30>		$(1 + w)T - 25$		ns
Data output hold time (from $\overline{LWR}$, $\overline{UWR}$ ↑)	t_{HWROD}	<31>		$T - 20$		ns
$\overline{WAIT}$ data output hold time (to address)	t_{SAWT1}	<32>	$w \geq 1$		$(1.5 + w_{AS} + w_{AH})T - 40$	ns
	t_{SAWT2}	<33>			$(1.5 + w + w_{AS} + w_{AH})T - 40$	ns
$\overline{WAIT}$ hold time (from address)	t_{HAWT1}	<34>	$w \geq 1$	$(0.5 + w + w_{AS} + w_{AH})T$		ns
	t_{HAWT2}	<35>		$(1.5 + w + w_{AS} + w_{AH})T$		ns
$\overline{WAIT}$ setup time (to ASTB↓)	t_{SSTWT1}	<36>	$w \geq 1$		$(1 + w_{AH})T - 32$	ns
	t_{SSTWT2}	<37>			$(1 + w + w_{AH})T - 32$	ns
$\overline{WAIT}$ hold time (from ASTB↓)	t_{HSTWT1}	<38>	$w \geq 1$	$(w + w_{AH})T$		ns
	t_{HSTWT2}	<39>		$(1 + w + w_{AH})T$		ns

Remarks 1. $T = t_{CYK}$

2. w_{AS}: Number of address setup wait states (0 or 1)

3. w_{AH}: Number of address hold wait states (0 or 1)

4. w: Number of wait clocks inserted in the bus cycle
The sampling timing changes when a programmable wait is inserted.

5. i: Number of idle states inserted after the read cycle (0 or 1)

6. Observe at least one of the data input hold times t_{HKID} or t_{HRDID}.

7. To understand how the number of wait cycles to be inserted is determined, refer to **4.6.3 Relationship between programmable wait and external wait.**

(b) CLKOUT synchronous (T_A = –40 to +85°C, REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V, output pin load capacitance: C_L = 50 pF)

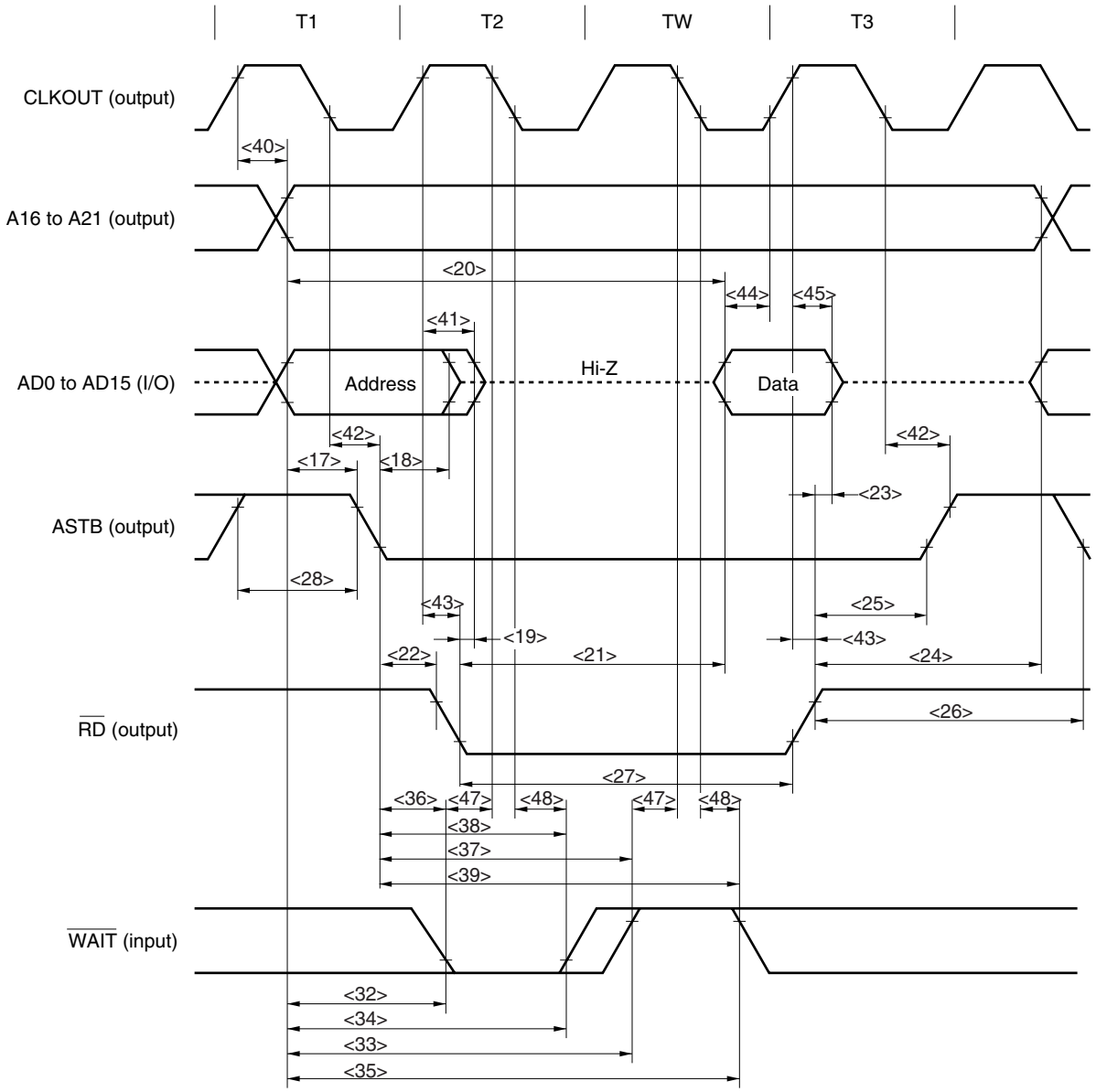
Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from CLKOUT↑ to address	t_{DKA}	<40>		–7	19	ns
Delay time from CLKOUT↑ to address float	t_{FKA}	<41>		–12	15	ns
Delay time from CLKOUT↓ to ASTB	t_{DKST}	<42>		–3 + w_{AH}T	19 + w_{AH}T	ns
Delay time from CLKOUT↑ to $\overline{RD}$, $\overline{LWR}$, $\overline{UWR}$	t_{DKRDWR}	<43>		–5	19	ns
Data input setup time (to CLKOUT↑)	t_{SIDK}	<44>		21		ns
Data input hold time (from CLKOUT↑)	t_{HKID}	<45>		5		ns
Delay time from CLKOUT↑ to data output	t_{DKOD}	<46>			19	ns
$\overline{WAIT}$ setup time (to CLKOUT↓)	t_{SWTK}	<47>		21		ns
$\overline{WAIT}$ hold time (from CLKOUT↓)	t_{HKWT}	<48>		5		ns

Remarks 1. T = t_{CYK}

2. w_{AH}: Number of address hold wait states (0 or 1)

3. Observe at least one of the data input hold times t_{HKID} or t_{HRDID}.

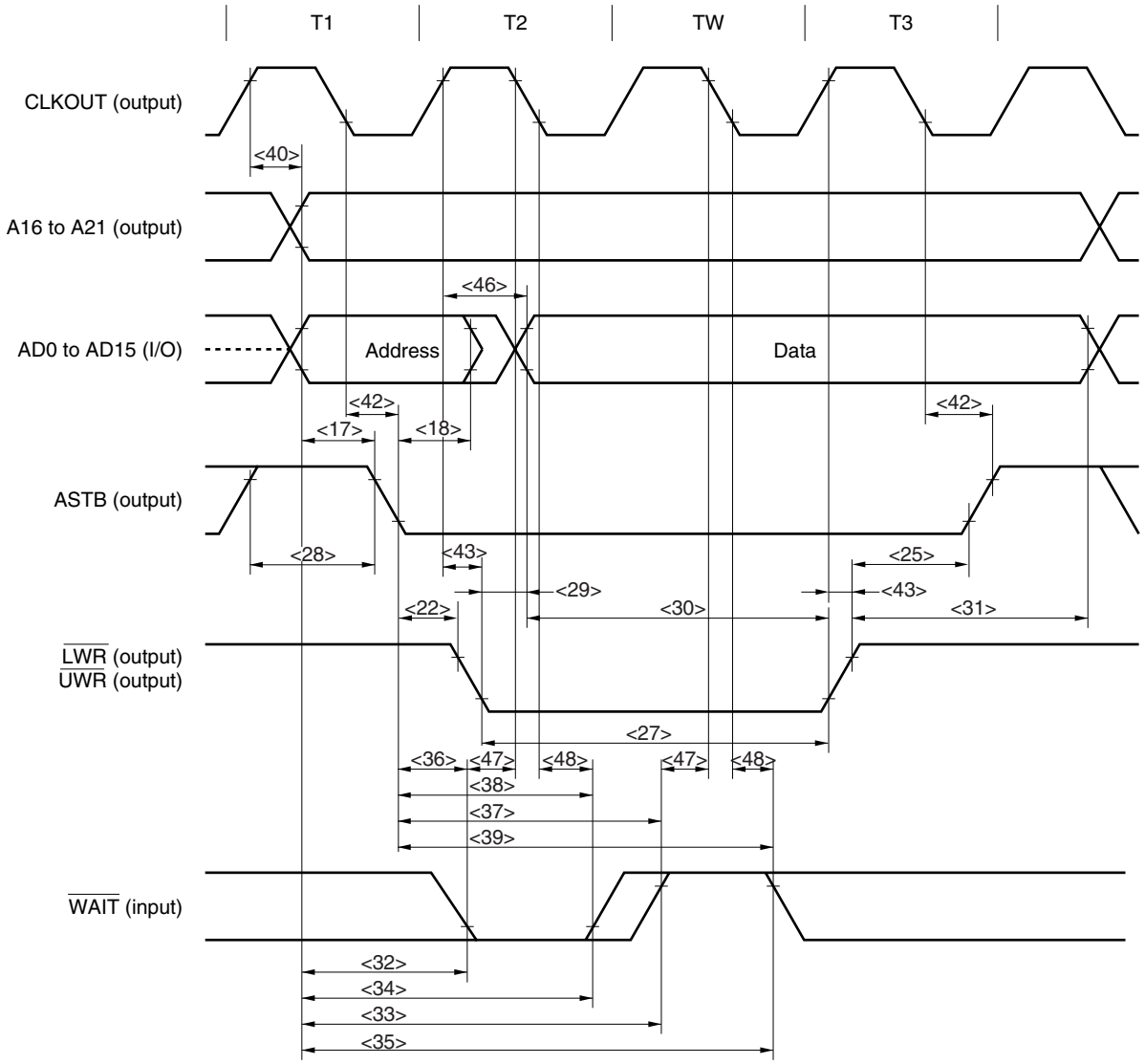
(c) Read cycle (CLKOUT synchronous/asynchronous, 1 wait)



Caution When using the CLKOUT signal for interfacing with external devices, set the internal system clock frequency (fxx) to 32 MHz or lower.

Remark $\overline{\text{LWR}}$ and $\overline{\text{UWR}}$ are high level.

(d) Write cycle (CLKOUT synchronous/asynchronous, 1 wait)



Caution When using the CLKOUT signal for interfacing with external devices, set the internal system clock frequency (fxx) to 32 MHz or lower.

Remark $\overline{RD}$ is high level.

(6) Interrupt timing

(T_A = −40 to +85°C, REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
NMI high-level width	t_{WNIH}	<49>		500		ns
NMI low-level width	t_{WNIL}	<50>		500		ns
INTPn high-level width	t_{WITH}	<51>	n = 0 to 4	500		ns
			n = 100, 101, 30, 31	5T + 10		ns
			n = 20 to 25 (when analog filter specified)	250		ns
			n = 20 to 25 (when digital filter specified)	5T + 10		ns
INTPn low-level width	t_{WITL}	<52>	n = 0 to 4	500		ns
			n = 100, 101, 30, 31	5T + 10		ns
			n = 20 to 25 (when analog filter specified)	250		ns
			n = 20 to 25 (when digital filter specified)	5T + 10		ns

Remark T: Digital filter sampling clock

T can be selected by setting the following registers.

- INTP100, INTP101:

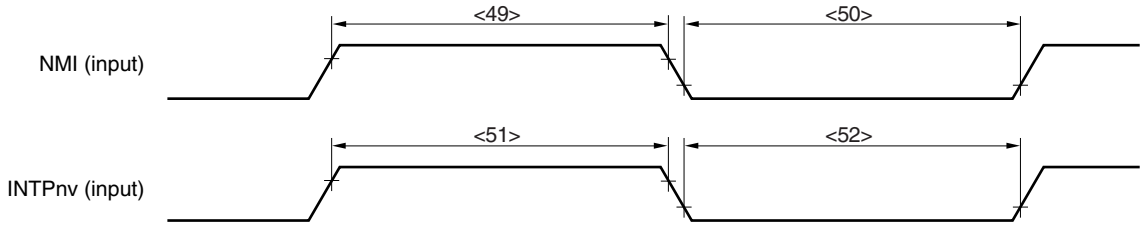
  Can be selected from f_{XX}/2, f_{XX}/4, f_{XX}/8, and f_{XX}/16 by setting the NRC101 and NRC100 bits of the timer 10 noise elimination time select register (NRC10) (f_{XX}: Internal system clock).

- INTP30:

  Can be selected from f_{XXTM3}/2, f_{XXTM3}/4, f_{XXTM3}/8, and f_{XXTM3}/16 by setting the NRC31 and NRC30 bits of the timer 3 noise elimination time selection register (NRC3) (f_{XXTM3}: Clock selected with the timer 3 clock selection register (PRM03)).

- INTP31:

  Can be selected from f_{XXTM3}/32, f_{XXTM3}/64, f_{XXTM3}/128, and f_{XXTM3}/256 by setting the NRC33 and NRC32 bits of the NRC3 register (f_{XXTM3}: Clock selected with the PRM03 register).



Remark n = 0 to 4, 100, 101, 20 to 25, 30, 31

(7) Timer input timing

(T_A = –40 to +85°C, REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
TIUD10, TCUD10 high-/low-level width	t_{WUDH}, t_{WUDL}	<53>		5T + 10		ns
TIUD10, TCUD10 input time difference	t_{PHUD}	<54>		5T + 10		ns
TCLRn high-/low-level width	t_{WTCH}, t_{WTCL}	<55>	n = 10, 2 (other than for through input), 3	5T + 10		ns
			n = 2 (for through input[Note])	2T + 10		ns
TIm high-/low-level width	t_{WTIH}, t_{WTIL}	<56>	m = 2 (other than for through input), 3	5T + 10		ns
			m = 2 (for through input[Note])	2T + 10		ns

Note When setting the CESE1 and CESE0 bits of timer 2 count clock/control edge selection register 0 (CSE0) to 1 and 0, respectively.

Remarks 1. T: Digital filter sampling clock

T can be selected by setting the following registers.

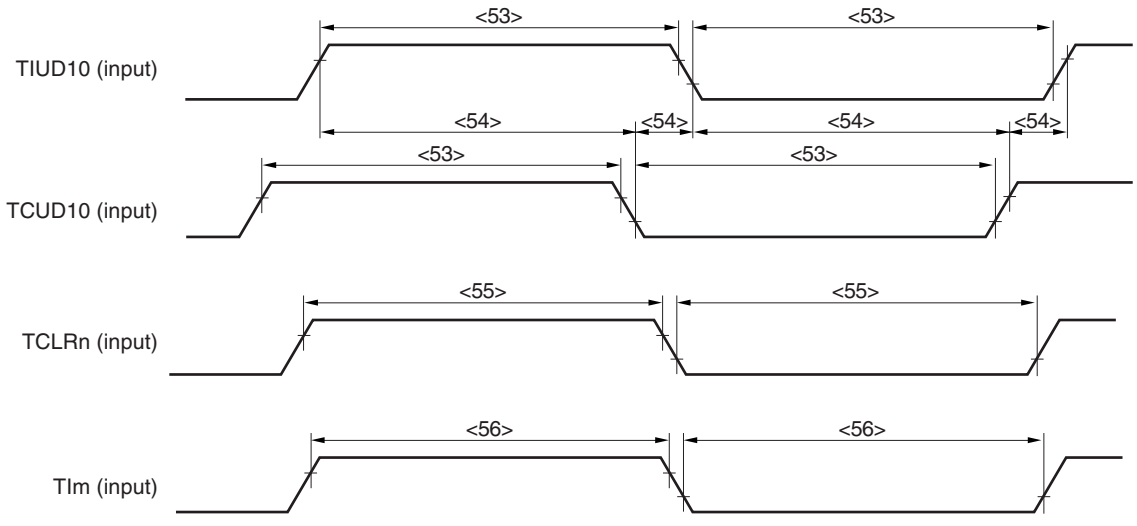
- TIUD10, TCUD10, TCLR10:

  Can be selected from f_{XX}/2, f_{XX}/4, f_{XX}/8, and f_{XX}/16 by setting the NRC101 and NRC100 bits of the timer 10 noise elimination time select register (NRC10).
- TCLR2, TI2:

  Fixed to f_{XX}/2.
- TCLR3, TI3:

  Can be selected from f_{XXTM3}/2, f_{XXTM3}/4, f_{XXTM3}/8, and f_{XXTM3}/16 by setting the NRC31 and NRC30 bits of the timer 3 noise elimination time selection register (NRC3) (f_{XXTM3}: Clock selected with the timer 3 clock selection register (PRM03)).

2. fx: Internal system clock frequency



Remark n = 10, 2, 3

m = 2, 3

(8) Timer operating frequency

(T_A = –40 to +85°C, REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V, output pin load capacitance: C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Timer 00, timer 01 operating frequency	T_0			40	MHz
Timer 10 operating frequency	T_1			20	MHz
Timer 20, timer 21 operating frequency	T_2			20	MHz
Timer 3 operating frequency	T_3			32	MHz

Remarks 1. T_0: f_{XX} or $f_{XX}/2$ can be selected using the timer 0 clock selection register (PRM01).
T_1: Select $f_{XX}/2$ by setting the timer 1/timer 2 clock selection register (PRM02) to 01H.
T_2: Select $f_{XX}/2$ by setting the timer 1/timer 2 clock selection register (PRM02) to 01H.
T_3: f_{XX} or $f_{XX}/2$ can be selected using the timer 3 clock selection register (PRM03).

2. f_{XX}: Internal system clock frequency

(9) CSI timing (1/2)

(a) Master mode (T_A = –40 to +85°C, REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V, output pin load capacitance: C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKn}$ cycle	t_{CYSK1}	<57>	Output	200		ns
$\overline{SCKn}$ high-level width	t_{WSK1H}	<58>	Output	$0.5t_{CYSK1}$ – 25		ns
$\overline{SCKn}$ low-level width	t_{WSK1L}	<59>	Output	$0.5t_{CYSK1}$ – 25		ns
SIn setup time (to $\overline{SCKn}$↑)	t_{SSISK}	<60>		35		ns
SIn hold time (from $\overline{SCKn}$↑)	t_{HSKSI}	<61>		30		ns
SOn output delay time (from $\overline{SCKn}$↓)	t_{DSKSO}	<62>			30	ns
SOn output hold time (from $\overline{SCKn}$↑)	t_{HSKSO}	<63>		$0.5t_{CYSK1}$ – 20		ns

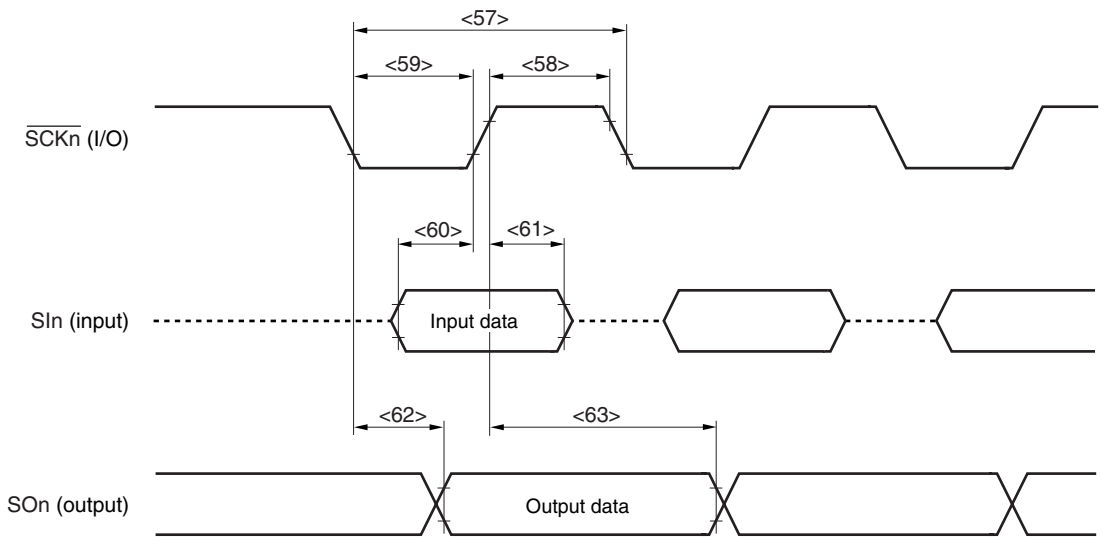
Remark n = 0, 1

(b) Slave mode (T_A = –40 to +85°C, REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V, output pin load capacitance: C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKn}$ cycle	t_{CYSK1}	<57>	Input	200		ns
$\overline{SCKn}$ high-level width	t_{WSK1H}	<58>	Input	90		ns
$\overline{SCKn}$ low-level width	t_{WSK1L}	<59>	Input	90		ns
SIn setup time (to $\overline{SCKn}$↑)	t_{SSISK}	<60>		50		ns
SIn hold time (from $\overline{SCKn}$↑)	t_{HSKSI}	<61>		50		ns
SOn output delay time (from $\overline{SCKn}$↓)	t_{DSKSO}	<62>			50	ns
SOn output hold time (from $\overline{SCKn}$↑)	t_{HSKSO}	<63>		t_{WSK1H}		ns

Remark n = 0, 1

(9) CSI timing (2/2)



Remarks 1. The broken lines indicate high impedance.

2. n = 0, 1

(10) UART0 timing

(T_A = –40 to +85°C, REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V, output pin load capacitance: C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
UART0 baud rate generator input frequency	f_{BRG}			20	MHz

Remarks 1. UART0 baud rate generator input frequency (f_{BRG}):
Can be selected from f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$, $f_{XX}/256$, $f_{XX}/512$, $f_{XX}/1024$, and $f_{XX}/2048$ by setting the TPS3 to TPS0 bits of clock select register 0 (CKSR0).

2. f_{XX}: Internal system clock frequency

(11) UART1 timing (1/2)

(a) Clocked master mode

(T_A = –40 to +85°C, REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V, output pin load capacitance: C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{ASCK1}$ cycle	t_{CYSK0}	<64>	Output	1000		ns
$\overline{ASCK1}$ high-level width	t_{WSK0H}	<65>	Output	kT – 20		ns
$\overline{ASCK1}$ low-level width	t_{WSK0L}	<66>	Output	kT – 20		ns
RXD1 setup time (to $\overline{ASCK1}$↑)	t_{SRXSK}	<67>		1.5T + 35		ns
RXD1 hold time (from $\overline{ASCK1}$↑)	t_{HSKRX}	<68>		0		ns
TXD1 output delay time (from $\overline{ASCK1}$↓)	t_{DSKTX}	<69>			T + 10	ns
TXD1 output hold time (from $\overline{ASCK1}$↑)	t_{HSKTX}	<70>		(k + 1)T – 20		ns

Remarks 1. T = $2t_{CYK}$

2. k: Setting value of prescaler compare register 1 (PRSCM1) of UART1

(b) Clocked slave mode

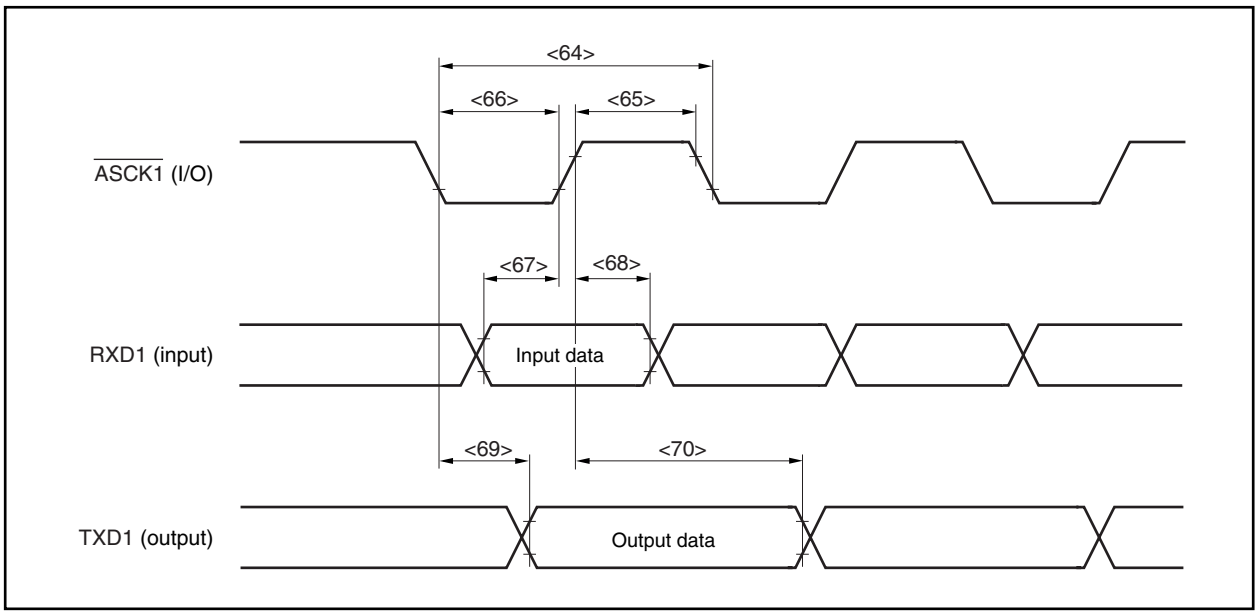
(T_A = –40 to +85°C, REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V, output pin load capacitance: C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{ASCK1}$ cycle	t_{CYSK0}	<64>	Input	1000		ns
$\overline{ASCK1}$ high-level width	t_{WSK0H}	<65>	Input	4T + 80		ns
$\overline{ASCK1}$ low-level width	t_{WSK0L}	<66>	Input	4T + 80		ns
RXD1 setup time (to $\overline{ASCK1}$↑)	t_{SRXSK}	<67>		T + 10		ns
RXD1 hold time (from $\overline{ASCK1}$↑)	t_{HSKRX}	<68>		T + 10		ns
TXD1 output delay time (from $\overline{ASCK1}$↓)	t_{DSKTX}	<69>			2.5T + 45	ns
TXD1 output hold time (from $\overline{ASCK1}$↑)	t_{HSKTX}	<70>		(k + 1.5)T		ns

Remarks 1. T = $2t_{CYK}$

2. k: Setting value of prescaler compare register 1 (PRSCM1) of UART1

(11) UART1 timing (2/2)



A/D Converter Characteristics (T_A = –40 to +85°C, REGIN = 3.0 to 3.6 V, $AV_{DD} = V_{DD} = RV_{DD}$ = 5 .0 V ±0.5 V, $V_{SS} = V_{SS3} = V_{SS} = CV_{SS}$ = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	–		10			bit
Overall error[Note 1]	–				±4	LSB
Quantization error	–				±1/2	LSB
Conversion time	t_{CONV}		5		10	μs
Sampling time	t_{SAMP}		833			ns
Zero-scale error[Note 1]	–				±4	LSB
Full-scale error[Note 1]	–				±4	LSB
Differential linearity error[Note 1]	–				±4	LSB
Integral linearity error[Note 1]	–				±4	LSB
Analog input voltage	V_{IAN}		–0.3		AV_{DD} + 0.3	V
Analog reference voltage	AV_{DD}		4.5		5.5	V
AVDD power supply current[Note 2]	AI_{DD}			4	8	mA

Notes 1. Quantization error (±0.5 LSB) is not included.

2. The V850E/IA2 incorporates two A/D converters. This is the rated value for one converter.

Remark LSB: Least Significant Bit

16.2 Flash Memory Programming Mode

Basic Characteristics (T_A =10 to 40°C (during rewrite), T_A = –40 to +85°C (except during rewrite), REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operating frequency	f_X		4		40	MHz
V_{PP} supply voltage	V_{PP1}	During flash memory programming	7.5	7.8	8.1	V
	V_{PPL}	V_{PP} low-level detection	–0.3		0.2REGIN	V
	V_{PPM}	V_{PP}, REGIN level detection	0.65REGIN		REGIN + 0.3	V
	V_{PPH}	V_{PP} high-voltage level detection	7.5	7.8	8.1	V
V_{DD3} supply current	I_{DD1}	V_{PP} = V_{PP1}			3.2f_{XX} + 30	mA
V_{PP} supply current	I_{PP}	V_{PP} = 7.8 V			100	mA
Step erase time	t_{ER}	**Note 1**	0.398	0.4	0.402	s
Overall erase time	t_{ERA}	When the step erase time = 0.4 s, **Note 2**			40	s
Write-back time	t_{WB}	**Note 3**	0.99	1	1.01	ms
Number of write-backs per write-back command	C_{WB}	When the write-back time = 1 ms, **Note 4**			300	Count/ write-back command
Number of erase/write-backs	C_{ERWB}				16	Count
Step writing time	t_{WT}	**Note 5**	18	20	22	μs
Overall writing time per word	t_{WTW}	When the step writing time = 20 μs (1 word = 4 bytes), **Note 6**	20		200	μs/word
Number of rewrites	C_{ERWR}	1 erase + 1 write after erase = 1 rewrite, **Note 7**		100		Count

Notes 1. The recommended setting value of the step erase time is 0.4 s.

2. The prewrite time prior to erasure and the erase verify time (write-back time) are not included.

3. The recommended setting value of the write-back time is 1 ms.

4. Write-back is executed once by the issuance of the write-back command. Therefore, the retry count must be the maximum value minus the number of commands issued.

5. The recommended setting value of the step writing time is 20 μs.

6. 20 μs is added to the actual writing time per word. The internal verify time during and after the writing is not included.

7. When writing initially to shipped products, it is counted as one rewrite for both “erase to write” and “write only”.

Example (P: Write, E: Erase)

Shipped product ——→ P → E → P → E → P: 3 rewrites

Shipped product → E → P → E → P → E → P: 3 rewrites

Remark When the PG-FP4 is used, a time parameter required for writing/erasing by downloading parameter files is automatically set. Do not change the settings unless otherwise specified.

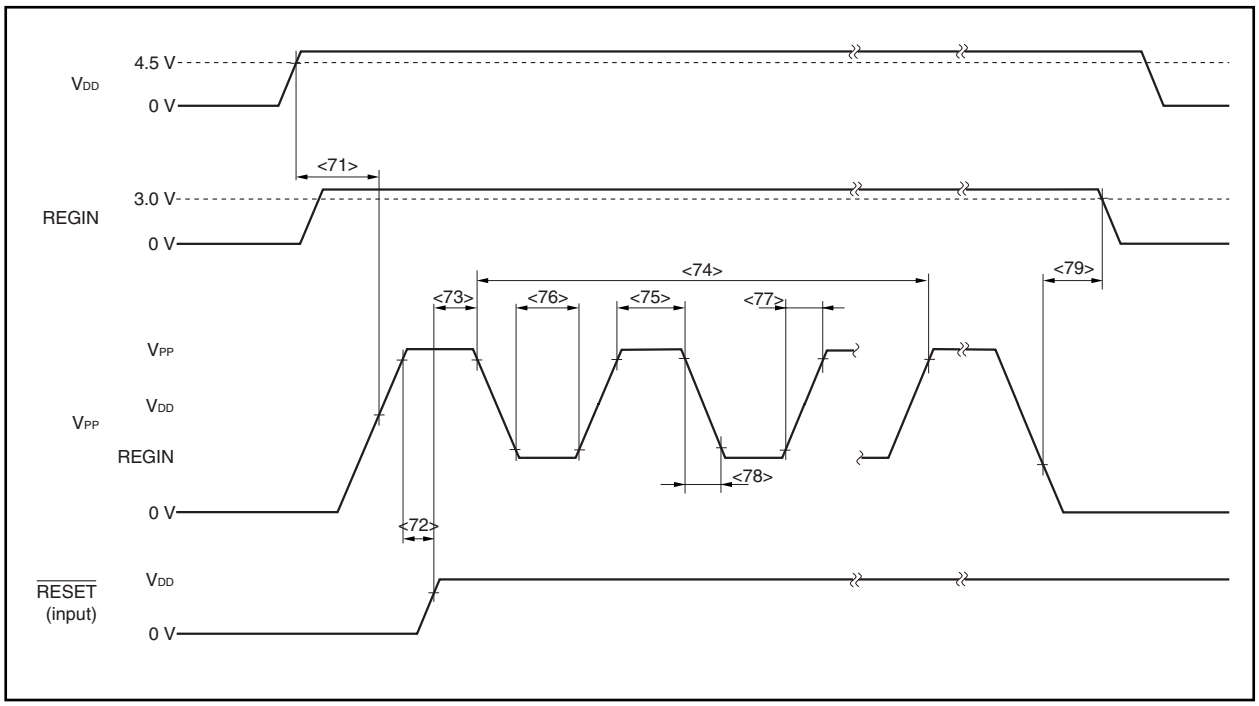
Serial Write Operation Characteristics

(T_A = 10 to +40°C, REGIN = 3.0 to 3.6 V, V_{DD} = RV_{DD} = 5.0 V ±0.5 V, V_{SS3} = V_{SS} = CV_{SS} = 0 V)

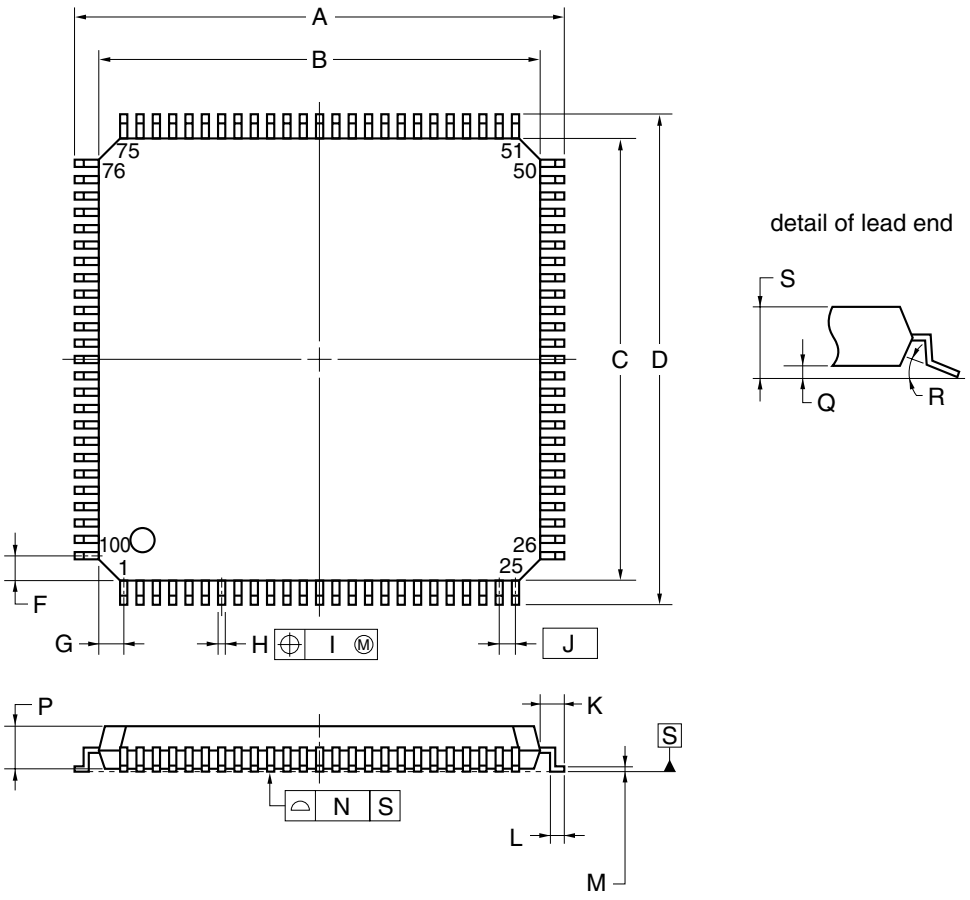
Parameter	Symbol		Conditions	MIN.	TYP.	MAX.	Unit
V_{DD}↑ to V_{PP}↑ set time	<71>	t_{DRPSR}		t_{RG} + 0.01			ms
V_{PP}↑ to $\overline{RESET}$↑ set time	<72>	t_{PSRRF}		1			μs
$\overline{RESET}$↑ to V_{PP} count start time	<73>	t_{RFOF}	V_{PP} = 7.8 V	10T + 1500			ns
Count execution time	<74>	t_{COUNT}				15	ms
V_{PP} counter high-level width	<75>	t_{CH}		1			μs
V_{PP} counter low-level width	<76>	t_{CL}		1			μs
V_{PP} counter rise time	<77>	t_R				1	μs
V_{PP} counter fall time	<78>	t_F				1	μs
V_{PP}↓ to REGIN↓ reset time	<79>	t_{PFDR}		10			μs

Remarks 1. t_{RG}: Regulator output stabilization time

2. T = t_{CYK}



100-PIN PLASTIC LQFP (FINE PITCH) (14x14)



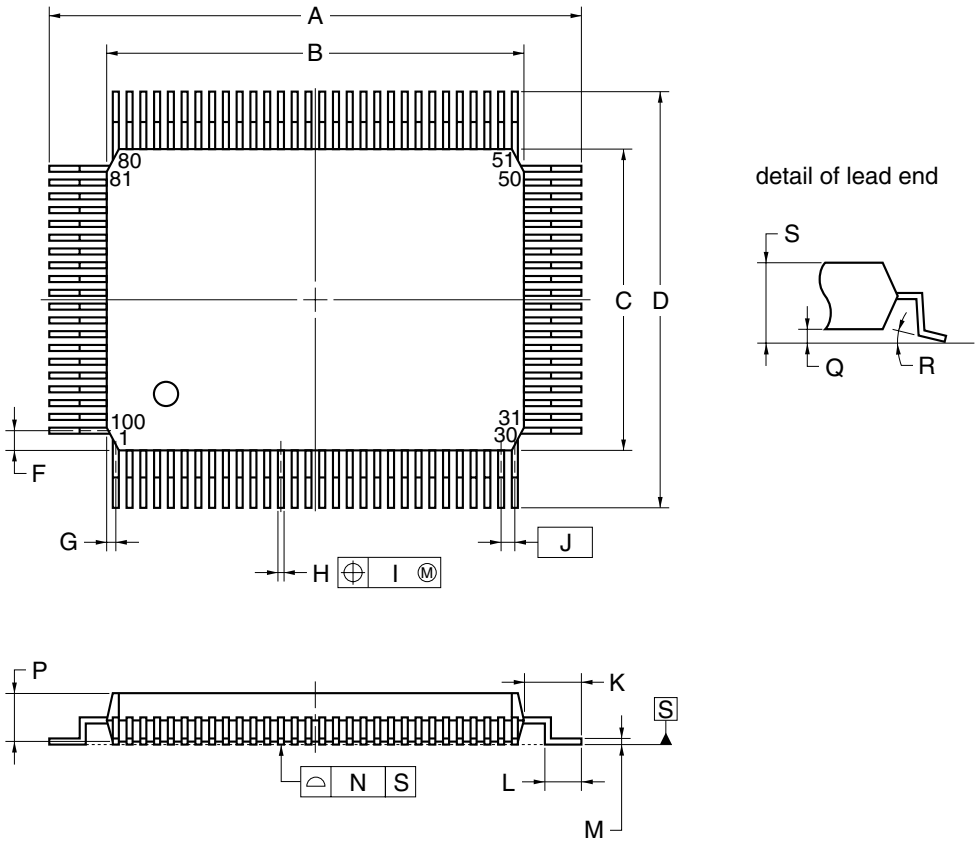
NOTE

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

100-PIN PLASTIC QFP (14x20)



NOTE

Each lead centerline is located within 0.15 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	23.6±0.4
B	20.0±0.2
C	14.0±0.2
D	17.6±0.4
F	0.8
G	0.6
H	0.30±0.10
I	0.15
J	0.65 (T.P.)
K	1.8±0.2
L	0.8±0.2
M	$0.15^{+0.10}_{-0.05}$
N	0.10
P	2.7±0.1
Q	0.1±0.1
R	5°±5°
S	3.0 MAX.

P100GF-65-3BA1-4

CHAPTER 18 RECOMMENDED SOLDERING CONDITIONS

The μPD703114 and 70F3114 should be soldered and mounted under the following recommended conditions. For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.necel.com/pkg/en/mount/index.html)

Table 18-1. Surface Mounting Type Soldering Conditions

(1) μPD703114GC-×××-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703114GC(A)-×××-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD70F3114GC-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD70F3114GC(A)-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Twice or less Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 to 72 hours)	IR35-107-2
VPS	Package peak temperature: 215°C, Time: 25 to 40 seconds (at 200°C or higher), Count: Twice or less Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 to 72 hours)	VP15-107-2
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

(2) μPD703114GF-×××-3BA: 100-pin plastic QFP (14 × 20)
μPD70F3114GF-3BA: 100-pin plastic QFP (14 × 20)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Twice or less Exposure limit: 7 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	IR35-207-2
VPS	Package peak temperature: 215°C, Time: 25 to 40 seconds (at 200°C or higher), Count: Twice or less Exposure limit: 7 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	VP15-207-2
Wave soldering	Solder bath temperature: 260°C max., Time: 10 seconds max., Count: Once Preheating temperature: 120°C max. (package surface temperature) Exposure limit: 7 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	WS60-207-1
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

★ **(3) μPD703114GC-×××-8EU-A: 100-pin plastic LQFP (fine pitch) (14 × 14)**
μPD703114GC(A)-×××-8EU-A: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD70F3114GC-8EU-A: 100-pin plastic LQFP (fine pitch) (14 × 14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: Three times or less Exposure limit: 7 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	IR60-207-3
Wave soldering	For details, consult an NEC Electronics sales representative.	–
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

★ **(4) μPD703114GF-×××-3BA-A: 100-pin plastic QFP (14 × 20)**

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: Three times or less Exposure limit: 3 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	IR60-203-3
Wave soldering	For details, consult an NEC Electronics sales representative.	–
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).

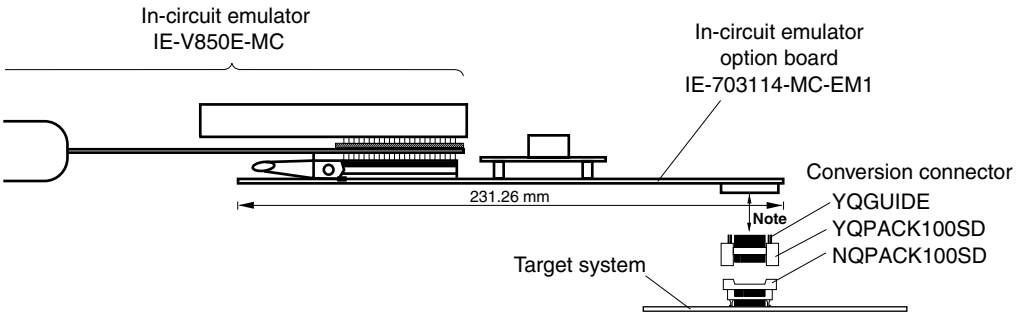
Remarks 1. Products with -A at the end of the part number are lead-free products.
2. For soldering methods and conditions other than those recommended above, consult an NEC Electronics sales representative.
3. For soldering conditions for the μPD70F3114GC(A)-8EU-A and 70F3114GF-3BA-A, consult an NEC Electronics sales representative.

APPENDIX A NOTES ON TARGET SYSTEM DESIGN

The following shows a diagram of the connection conditions between the in-circuit emulator option board and conversion connector. Design your system making allowances for conditions such as the form of parts mounted on the target system as shown below.

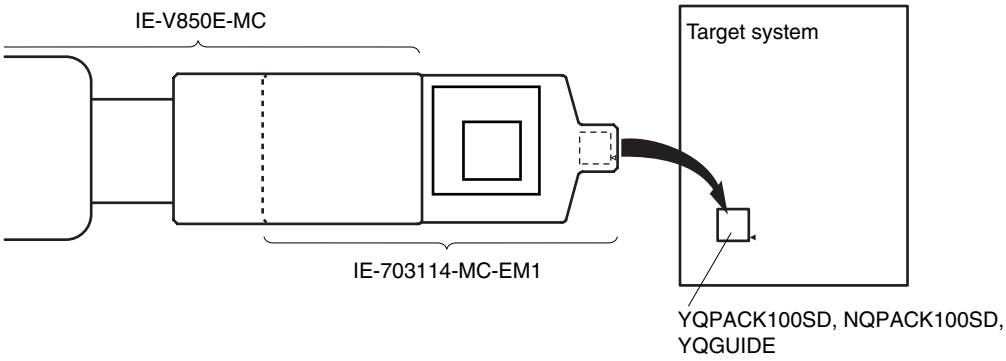
Figure A-1. 100-Pin Plastic LQFP (Fine Pitch) (14 × 14)

Side view



Note YQSOCKET100SDN (sold separately) can be inserted here to adjust the height (height: 3.2 mm).

Top view



Connection condition diagram

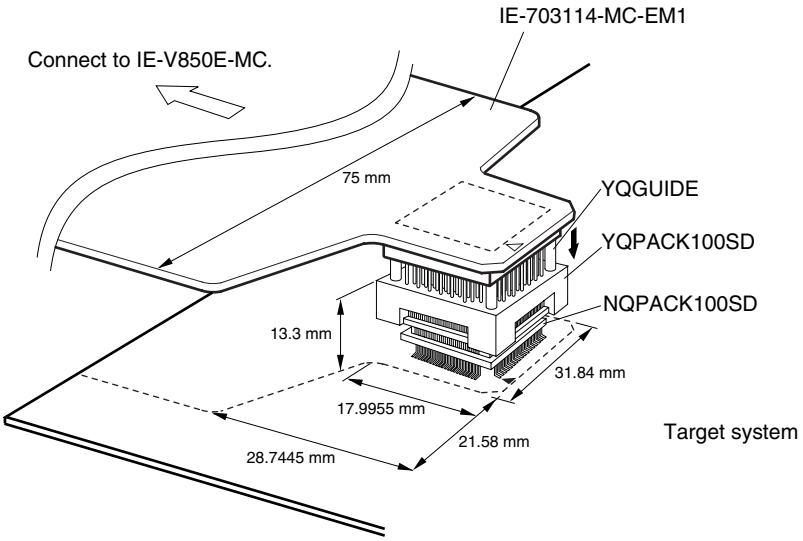
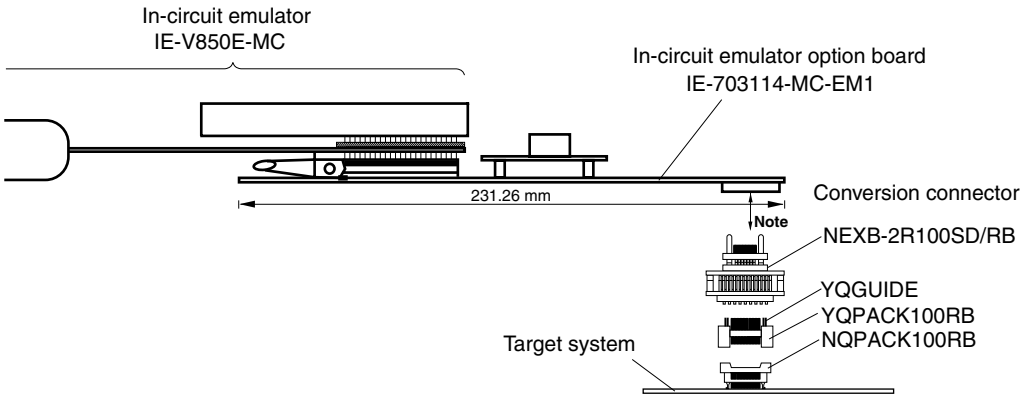


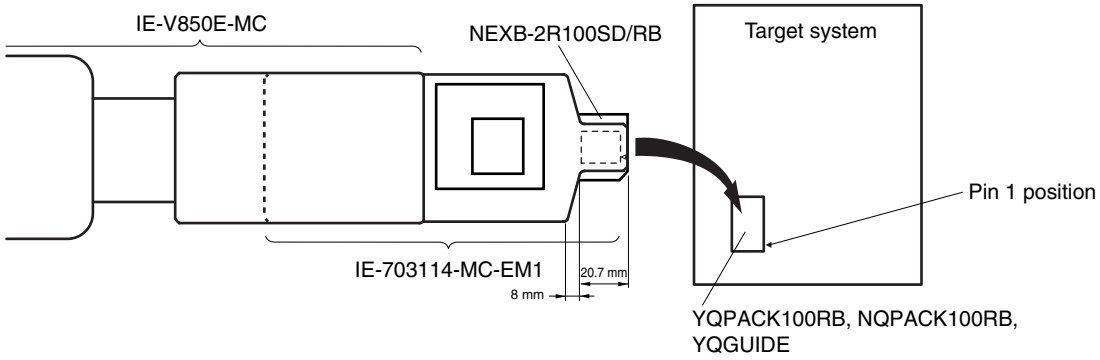
Figure A-2. 100-Pin Plastic QFP (14 × 20)

Side view

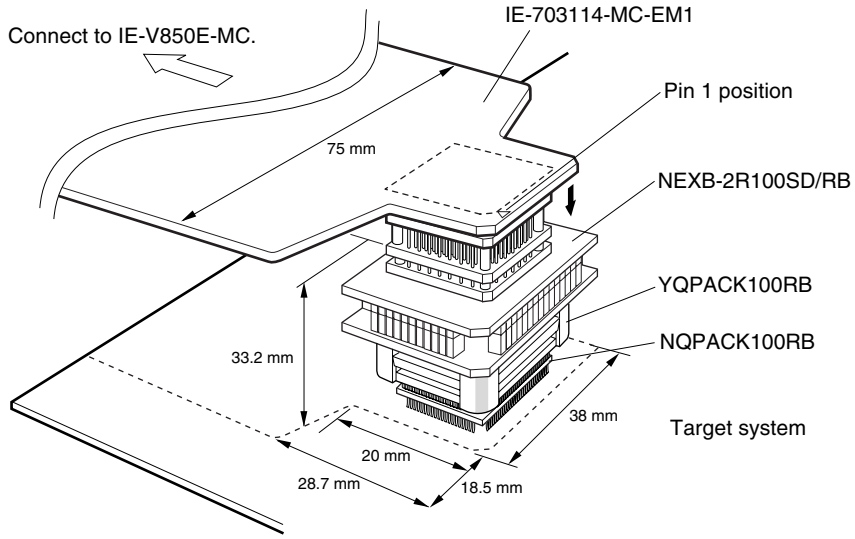


Note YQSOCKET100SDN (sold separately) can be inserted here to adjust the height (height: 3.2 mm).

Top view



Connection condition diagram



APPENDIX B REGISTER INDEX

(1/9)

Symbol	Register Name	Unit	Page
ADCR00	A/D conversion result register 00	ADC	521
ADCR01	A/D conversion result register 01	ADC	521
ADCR02	A/D conversion result register 02	ADC	521
ADCR03	A/D conversion result register 03	ADC	521
ADCR04	A/D conversion result register 04	ADC	521
ADCR05	A/D conversion result register 05	ADC	521
ADCR10	A/D conversion result register 10	ADC	521
ADCR11	A/D conversion result register 11	ADC	521
ADCR12	A/D conversion result register 12	ADC	521
ADCR13	A/D conversion result register 13	ADC	521
ADCR14	A/D conversion result register 14	ADC	521
ADCR15	A/D conversion result register 15	ADC	521
ADCR16	A/D conversion result register 16	ADC	521
ADCR17	A/D conversion result register 17	ADC	521
ADETM0	A/D voltage detection mode register 0	ADC	520
ADETM0H	A/D voltage detection mode register 0H	ADC	520
ADETM0L	A/D voltage detection mode register 0L	ADC	520
ADETM1	A/D voltage detection mode register 1	ADC	520
ADETM1H	A/D voltage detection mode register 1H	ADC	520
ADETM1L	A/D voltage detection mode register 1L	ADC	520
ADIC0	Interrupt control register	INTC	150
ADIC1	Interrupt control register	INTC	150
ADSCM00	A/D scan mode register 00	ADC	517
ADSCM00H	A/D scan mode register 00H	ADC	517
ADSCM00L	A/D scan mode register 00L	ADC	517
ADSCM01	A/D scan mode register 01	ADC	519
ADSCM01H	A/D scan mode register 01H	ADC	519
ADSCM01L	A/D scan mode register 01L	ADC	519
ADSCM10	A/D scan mode register 10	ADC	517
ADSCM10H	A/D scan mode register 10H	ADC	517
ADSCM10L	A/D scan mode register 10L	ADC	517
ADSCM11	A/D scan mode register 11	ADC	519
ADSCM11H	A/D scan mode register 11H	ADC	519
ADSCM11L	A/D scan mode register 11L	ADC	519
ASIF0	Asynchronous serial interface mode transmit status register 0	UART0	415
ASIM0	Asynchronous serial interface mode register 0	UART0	411
ASIM10	Asynchronous serial interface mode register 10	UART1	442
ASIM11	Asynchronous serial interface mode register 11	UART1	444

(2/9)

Symbol	Register Name	Unit	Page
ASIS0	Asynchronous serial interface status register 0	UART0	414
ASIS1	Asynchronous serial interface status register 1	UART1	445
AWC	Address wait control register	BCU	95
BCC	Bus cycle control register	BCU	97
BCT0	Bus cycle type configuration register 0	BCU	85
BCT1	Bus cycle type configuration register 1	BCU	85
BFCM00	Buffer register CM00	TM00	202
BFCM01	Buffer register CM01	TM00	202
BFCM02	Buffer register CM02	TM00	202
BFCM03	Buffer register CM03	TM00	204
BFCM04	Buffer register CM04	TM00	202
BFCM05	Buffer register CM05	TM00	202
BFCM10	Buffer register CM10	TM01	202
BFCM11	Buffer register CM11	TM01	202
BFCM12	Buffer register CM12	TM01	202
BFCM13	Buffer register CM13	TM01	204
BFCM14	Buffer register CM14	TM01	202
BFCM15	Buffer register CM15	TM01	202
BRGC0	Baud rate generator control register 0	UART0	433
BSC	Bus size configuration register	BCU	87
CC100	Capture/compare register 100	TM10	308
CC101	Capture/compare register 101	TM10	309
CC10IC0	Interrupt control register	INTC	150
CC10IC1	Interrupt control register	INTC	150
CC2IC0	Interrupt control register	INTC	150
CC2IC1	Interrupt control register	INTC	150
CC2IC2	Interrupt control register	INTC	150
CC2IC3	Interrupt control register	INTC	150
CC2IC4	Interrupt control register	INTC	150
CC2IC5	Interrupt control register	INTC	150
CC30	Capture/compare register 30	TM3	371
CC31	Capture/compare register 31	TM3	371
CC3IC0	Interrupt control register	INTC	150
CC3IC1	Interrupt control register	INTC	150
CCR0	Capture/compare control register 0	TM10	302
CCSTATE0	Timer 2 capture/compare 1 to 4 status register 0	TM2	344
CCSTATE0H	Timer 2 capture/compare 1 to 4 status register 0H	TM2	344
CCSTATE0L	Timer 2 capture/compare 1 to 4 status register 0L	TM2	344
CKC	Clock control register	CG	177
CKSR0	Clock select register 0	UART0	432
CM000	Compare register 000	TM00	201

(3/9)

Symbol	Register Name	Unit	Page
CM001	Compare register 001	TM00	202
CM002	Compare register 002	TM00	202
CM003	Compare register 003	TM00	202
CM004	Compare register 004	TM00	202
CM005	Compare register 005	TM00	202
CM00IC1	Interrupt control register	INTC	150
CM010	Compare register 010	TM01	201
CM011	Compare register 011	TM01	201
CM012	Compare register 012	TM01	201
CM013	Compare register 013	TM01	202
CM014	Compare register 014	TM01	202
CM015	Compare register 015	TM01	202
CM01IC1	Interrupt control register	INTC	150
CM02IC1	Interrupt control register	INTC	150
CM03IC0	Interrupt control register	INTC	150
CM03IC1	Interrupt control register	INTC	150
CM04IC0	Interrupt control register	INTC	150
CM04IC1	Interrupt control register	INTC	150
CM05IC0	Interrupt control register	INTC	150
CM05IC1	Interrupt control register	INTC	150
CM100	Compare register 100	TM10	307
CM101	Compare register 101	TM10	307
CM10IC0	Interrupt control register	INTC	150
CM10IC1	Interrupt control register	INTC	150
CM4	Compare register 4	TM4	398
CM4IC0	Interrupt control register	INTC	150
CMSE050	Timer 2 subchannel 0, 5 capture/compare control register	TM2	338
CMSE120	Timer 2 subchannel 1, 2 capture/compare control register	TM2	339
CMSE340	Timer 2 subchannel 3, 4 capture/compare control register	TM2	341
CSC0	Chip area selection control register	BCU	82
CSC1	Chip area selection control register	BCU	82
CSCE0	Timer 2 software event capture register	TM2	346
CSE0	Timer 2 count clock/control edge selection register 0	TM2	331
CSE0H	Timer 2 count clock/control edge selection register 0H	TM2	331
CSE0L	Timer 2 count clock/control edge selection register 0L	TM2	331
CSIC0	Clocked serial interface clock selection register 0	CSI0	480
CSIC1	Clocked serial interface clock selection register 1	CSI1	480
CSIIC0	Interrupt control register	INTC	150
CSIIC1	Interrupt control register	INTC	150
CSIM0	Clocked serial interface mode register 0	CSI0	478
CSIM1	Clocked serial interface mode register 1	CSI1	478

(4/9)

Symbol	Register Name	Unit	Page
CSL10	CC101 capture input selection register	TM10	306
CVPE10	Timer 2 subchannel 1 main capture/compare register	TM2	328
CVPE20	Timer 2 subchannel 2 main capture/compare register	TM2	328
CVPE30	Timer 2 subchannel 3 main capture/compare register	TM2	328
CVPE40	Timer 2 subchannel 4 main capture/compare register	TM2	328
CVSE00	Timer 2 subchannel 0 capture/compare register	TM2	328
CVSE10	Timer 2 subchannel 1 sub capture/compare register	TM2	329
CVSE20	Timer 2 subchannel 2 sub capture/compare register	TM2	329
CVSE30	Timer 2 subchannel 3 sub capture/compare register	TM2	329
CVSE40	Timer 2 subchannel 4 sub capture/compare register	TM2	329
CVSE50	Timer 2 subchannel 5 capture/compare register	TM2	329
DADC0	DMA addressing control register 0	DMAC	112
DADC1	DMA addressing control register 1	DMAC	112
DADC2	DMA addressing control register 2	DMAC	112
DADC3	DMA addressing control register 3	DMAC	112
DBC0	DMA transfer count register 0	DMAC	111
DBC1	DMA transfer count register 1	DMAC	111
DBC2	DMA transfer count register 2	DMAC	111
DBC3	DMA transfer count register 3	DMAC	111
DCHC0	DMA channel control register 0	DMAC	114
DCHC1	DMA channel control register 1	DMAC	114
DCHC2	DMA channel control register 2	DMAC	114
DCHC3	DMA channel control register 3	DMAC	114
DDA0H	DMA destination address register 0H	DMAC	109
DDA0L	DMA destination address register 0L	DMAC	110
DDA1H	DMA destination address register 1H	DMAC	109
DDA1L	DMA destination address register 1L	DMAC	110
DDA2H	DMA destination address register 2H	DMAC	109
DDA2L	DMA destination address register 2L	DMAC	110
DDA3H	DMA destination address register 3H	DMAC	109
DDA3L	DMA destination address register 3L	DMAC	110
DDIS	DMA disable status register	DMAC	116
DETIC0	Interrupt control register	INTC	150
DETIC1	Interrupt control register	INTC	150
DMAIC0	Interrupt control register	INTC	150
DMAIC1	Interrupt control register	INTC	150
DMAIC2	Interrupt control register	INTC	150
DMAIC3	Interrupt control register	INTC	150
DRST	DMA restart register	DMAC	116
DSA0H	DMA source address register 0H	DMAC	107
DSA0L	DMA source address register 0L	DMAC	108

(5/9)

Symbol	Register Name	Unit	Page
DSA1H	DMA source address register 1H	DMAC	107
DSA1L	DMA source address register 1L	DMAC	108
DSA2H	DMA source address register 2H	DMAC	107
DSA2L	DMA source address register 2L	DMAC	108
DSA3H	DMA source address register 3H	DMAC	107
DSA3L	DMA source address register 3L	DMAC	108
DTFR0	DMA trigger factor register 0	DMAC	117
DTFR1	DMA trigger factor register 1	DMAC	117
DTFR2	DMA trigger factor register 2	DMAC	117
DTFR3	DMA trigger factor register 3	DMAC	117
DTM00	Dead time timer 00	TM00	201
DTM01	Dead time timer 01	TM00	201
DTM02	Dead time timer 02	TM00	201
DTM10	Dead time timer 10	TM01	201
DTM11	Dead time timer 11	TM01	201
DTM12	Dead time timer 12	TM01	201
DTRR0	Dead time timer reload register 0	TM00	201
DTRR1	Dead time timer reload register 1	TM00	201
DWC0	Data wait control register 0	BCU	94
DWC1	Data wait control register 1	BCU	94
FEM0	Timer 2 input filter mode register 0	TM2	160, 586
FEM1	Timer 2 input filter mode register 1	TM2	160, 586
FEM2	Timer 2 input filter mode register 2	TM2	160, 586
FEM3	Timer 2 input filter mode register 3	TM2	160, 586
FEM4	Timer 2 input filter mode register 4	TM2	160, 586
FEM5	Timer 2 input filter mode register 5	TM2	160, 586
FLPMC	Flash programming mode control register	CPU	624
IMR0	Interrupt mask register 0	INTC	153
IMR0H	Interrupt mask register 0H	INTC	153
IMR0L	Interrupt mask register 0L	INTC	153
IMR1	Interrupt mask register 1	INTC	153
IMR1H	Interrupt mask register 1H	INTC	153
IMR1L	Interrupt mask register 1L	INTC	153
IMR2	Interrupt mask register 2	INTC	153
IMR2H	Interrupt mask register 2H	INTC	153
IMR2L	Interrupt mask register 2L	INTC	153
IMR3	Interrupt mask register 3	INTC	153
IMR3H	Interrupt mask register 3H	INTC	153
IMR3L	Interrupt mask register 3L	INTC	153
INTM0	External interrupt mode register 0	INTC	142
INTM1	External interrupt mode register 1	INTC	156

(6/9)

Symbol	Register Name	Unit	Page
INTM2	External interrupt mode register 2	INTC	156
ISPR	In-service priority register	INTC	154
ITRG0	A/D internal trigger selection register 0	ADC	524
ITRG1	A/D internal trigger selection register 1	ADC	524
LOCKR	Lock register	CPU	180
NRC10	Timer 10 noise elimination time selection register	TM10	583
NRC3	Timer 3 noise elimination time selection register	TM3	584
OCTLE0	Timer 2 output control register	TM2	336
OCTLE0H	Timer 2 output control register 0H	TM2	336
OCTLE0L	Timer 2 output control register 0L	TM2	336
ODELE0	Timer 2 output delay register	TM2	345
ODELE0H	Timer 2 output delay register 0H	TM2	345
ODELE0L	Timer 2 output delay register 0L	TM2	345
P0	Port 0	Port	563
P0IC0	Interrupt control register	INTC	150
P0IC1	Interrupt control register	INTC	150
P0IC2	Interrupt control register	INTC	150
P0IC3	Interrupt control register	INTC	150
P0IC4	Interrupt control register	INTC	150
P1	Port 1	Port	564
P2	Port 2	Port	566
P3	Port 3	Port	568
P4	Port 4	Port	570
PCM	Port CM	Port	579
PCT	Port CT	Port	576
PDH	Port DH	Port	572
PDL	Port DL	Port	574
PDLH	Port DLH	Port	574
PDLL	Port DLL	Port	574
PFC1	Port 1 function control register	Port	565
PFC2	Port 2 function control register	Port	567
PFC3	Port 3 function control register	Port	569
PHCMD	Peripheral command register	CPU	176
PHS	Peripheral status register	CPU	179
PM1	Port 1 mode register	Port	564
PM2	Port 2 mode register	Port	566
PM3	Port 3 mode register	Port	568
PM4	Port 4 mode register	Port	571
PMC1	Port 1 mode control register	Port	565
PMC2	Port 2 mode control register	Port	567
PMC3	Port 3 mode control register	Port	569

(7/9)

Symbol	Register Name	Unit	Page
PMC4	Port 4 mode control register	Port	571
PMCCM	Port CM mode control register	Port	579
PMCCT	Port CT mode control register	Port	577
PMCDH	Port DH mode control register	Port	573
PMCDL	Port DL mode control register	Port	575
PMCDLH	Port DL mode control register H	Port	575
PMCDLL	Port DL mode control register L	Port	575
PMCM	Port CM mode register	Port	579
PMCT	Port CT mode register	Port	577
PMDH	Port DH mode register	Port	573
PMDL	Port DL mode register	Port	575
PMDLH	Port DL mode register H	Port	575
PMDLL	Port DL mode register L	Port	575
POER0	PWM output enable register 0	TM00	218
POER1	PWM output enable register 1	TM01	218
PRCMD	Command register	CPU	184
PRM01	Timer 0 clock selection register	TM0	205
PRM02	Timer 1/timer 2 clock selection register	TM1/TM2	299, 330
PRM03	Timer 3 clock selection register	TM3	373
PRM10	Prescaler mode register 10	TM10	305
PRSCM1	Prescaler compare register 1	UART1	469
PRSCM3	Prescaler compare register 3	CSI0, CSI1	510
PRSM1	Prescaler mode register 1	UART1	467
PRSM3	Prescaler mode register 3	CSI0, CSI1	509
PSC	Power save control register	CPU	185
PSMR	Power save mode register	CPU	184
PSTO0	PWM software timing output register 0	TM00	219
PSTO1	PWM software timing output register 1	TM01	219
REGC	Regulator control register	Regulator	602
RXB0	Receive buffer register	UART0	416
RXB1	2-frame continuous reception buffer registers 1	UART1	447
RXBL1	Receive buffer register L1	UART1	447
SEIC0	Interrupt control register	INTC	150
SESA10	Signal edge selection register 10	INTC, TM10	157, 303
SESC	Valid edge selection register	INTC, TM3	159, 378
SESE0	Timer 2 subchannel input event edge selection register	TM2	332
SESE0H	Timer 2 subchannel input event edge selection register 0H	TM2	332
SESE0L	Timer 2 subchannel input event edge selection register 0L	TM2	332
SIO0	Serial I/O shift register 0	CSI0	490
SIO1	Serial I/O shift register 1	CSI1	490
SIOL0	Serial I/O shift register L0	CSI0	491

(8/9)

Symbol	Register Name	Unit	Page
SIOL1	Serial I/O shift register L1	CSI1	491
SIRB0	Clocked serial interface receive buffer register 0	CSI0	482
SIRB1	Clocked serial interface receive buffer register 1	CSI1	482
SIRBE0	Clocked serial interface read-only receive buffer register 0	CSI0	484
SIRBE1	Clocked serial interface read-only receive buffer register 1	CSI1	484
SIRBEL0	Clocked serial interface read-only receive buffer register L0	CSI0	485
SIRBEL1	Clocked serial interface read-only receive buffer register L1	CSI1	485
SIRBL0	Clocked serial interface receive buffer register L0	CSI1	483
SIRBL1	Clocked serial interface receive buffer register L1	CSI0	483
SOTB0	Clocked serial interface transmit buffer register 0	CSI1	486
SOTB1	Clocked serial interface transmit buffer register 1	CSI0	486
SOTBF0	Clocked serial interface initial transmit buffer register 0	CSI1	488
SOTBF1	Clocked serial interface initial transmit buffer register 1	CSI0	488
SOTBFL0	Clocked serial interface initial transmit buffer register L0	CSI1	489
SOTBFL1	Clocked serial interface initial transmit buffer register L1	CSI0	489
SOTBL0	Clocked serial interface transmit buffer register L0	CSI1	487
SOTBL1	Clocked serial interface transmit buffer register L1	CSI0	487
SPEC0	TOMR write enable register 0	TM00	228
SPEC1	TOMR write enable register 1	TM01	228
SRIC0	Interrupt control register	INTC	150
SRIC1	Interrupt control register	INTC	150
STATUS0	Status register 0	TM10	306
STIC0	Interrupt control register	INTC	150
STIC1	Interrupt control register	INTC	150
STOPTE0	Timer 2 clock stop register 0	TM2	330
STOPTE0H	Timer 2 clock stop register 0H	TM2	330
STOPTE0L	Timer 2 clock stop register 0L	TM2	330
TBSTATE0	Timer 2 timer base status register 0	TM2	343
TBSTATE0H	Timer 2 timer base status register 0H	TM2	343
TBSTATE0L	Timer 2 timer base status register 0L	TM2	343
TCRE0	Timer 2 time base control register 0	TM2	333
TCRE0H	Timer 2 time base control register 0H	TM2	333
TCRE0L	Timer 2 time base control register 0L	TM2	333
TM00	Timer 00	TM00	200
TM01	Timer 01	TM01	200
TM0IC0	Interrupt control register	INTC	150
TM0IC1	Interrupt control register	INTC	150
TM10	Timer 10	TM10	297
TM20	Timer 20	TM2	328
TM21	Timer 21	TM2	328
TM2IC0	Interrupt control register	INTC	150

(9/9)

Symbol	Register Name	Unit	Page
TM2IC1	Interrupt control register	INTC	150
TM3	Timer 3	TM3	370
TM3IC0	Interrupt control register	INTC	150
TM4	Timer 4	TM4	397
TMC00	Timer control register 00	TM00	206
TMC00H	Timer control register 00H	TM00	206
TMC00L	Timer control register 00L	TM00	206
TMC01	Timer control register 01	TM01	206
TMC01H	Timer control register 01H	TM01	206
TMC01L	Timer control register 01L	TM01	206
TMC10	Timer control register 10	TM10	301
TMC30	Timer control register 30	TM3	374
TMC31	Timer control register 31	TM3	376
TMC4	Timer control register 4	TM4	400
TMIC0	Timer connection selection register 0	TM1/TM2	405
TO3C	Timer 3 output control register	TM3	379
TOMR0	Timer output mode register 0	TM00	213
TOMR1	Timer output mode register 1	TM01	213
TUC00	Timer unit control register 00	TM00	212
TUC01	Timer unit control register 01	TM01	212
TUM0	Timer unit mode register 0	TM10	300
TXB0	Transmit buffer register 0	UART0	417
TXS1	2-frame continuous transmission shift register 1	UART1	450
TXSL1	Transmit shift register L1	UART1	450
VSWC	System wait control register	BCU	78

APPENDIX C INSTRUCTION SET LIST

C.1 Conventions

(1) Symbols used in operand descriptions

Symbol	Explanation
reg1	General-purpose register (Used as source register)
reg2	General-purpose register (Usually used as destination register. Used as source register in some instructions.)
reg3	General-purpose register (Usually stores remainder of division result or higher 32 bits of multiplication result.)
bit#3	3-bit data for bit number specification
immX	X-bit immediate data
dispX	X-bit displacement data
regID	System register number
vector	5-bit data that specifies a trap vector (00H to 1FH)
cccc	4-bit data that shows a condition code
sp	Stack pointer (r3)
ep	Element pointer (r30)
list×	X-item register list

(2) Symbols used in operands

Symbol	Explanation
R	1 bit of data of code that specifies reg1 or regID
r	1 bit of data of code that specifies reg2
w	1 bit of data of code that specifies reg3
d	1 bit of data of a displacement
I	1 bit of immediate data (Shows higher bit of immediate data)
i	1 bit of immediate data
cccc	4-bit data that shows a condition code
CCCC	4-bit data that shows condition code of Bcond instruction
bbb	3-bit data for bit number specification
L	1 bit of data that specifies a program register in a register list
S	1 bit of data that specifies a system register in a register list

(3) Symbols used in operations

Symbol	Explanation
←	Assignment
GR [ ]	General-purpose register
SR [ ]	System register
zero-extend (n)	Zero-extend n to word length.
sign-extend (n)	Sign-extend n to word length.
load-memory (a, b)	Read data of size “b” from address “a”.
store-memory (a, b, c)	Write data “b” of size “c” to address “a”.
load-memory-bit (a, b)	Read bit “b” of address “a”.
store-memory-bit (a, b, c)	Write “c” in bit “b” of address “a”.
saturated (n)	Perform saturation processing of n (n is 2’s complement). If n is a computation result and n > 7FFFFFFFH, make n = 7FFFFFFFH. If n is a computation result and n < 80000000H, make n = 80000000H.
result	Reflect result in flag.
Byte	Byte (8 bits)
Half-word	Halfword (16 bits)
Word	Word (32 bits)
+	Addition
–	Subtraction
||	Bit concatenation
×	Multiplication
÷	Division
%	Remainder of division result
AND	Logical product
OR	Logical sum
XOR	Exclusive logical sum
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	When executing another instruction immediately after instruction execution (issue).
r	When repeating same instruction immediately after instruction execution (repeat)
l	When using instruction execution result in instruction immediately after instruction execution (latency)

(5) Symbols used in flag operations

Symbol	Explanation
(Blank)	No change
0	Clear to 0.
×	Set or cleared according to result.
R	Previously saved value is restored.

(6) Condition codes

Condition Name (cond)	Condition Code (CCCC)	Condition Expression	Explanation
V	0000	OV = 1	Overflow
NV	1000	OV = 0	No overflow
C/L	0001	CY = 1	Carry Lower (Less than)
NC/NL	1001	CY = 0	No carry No lower (Greater than or equal)
Z/E	0010	Z = 1	Zero Equal
NZ/NE	1010	Z = 0	Not zero Not equal
NH	0011	(CY or Z) = 1	Not higher (Less than equal)
H	1011	(CY or Z) = 0	Higher (Greater than)
N	0100	S = 1	Negative
P	1100	S = 0	Positive
T	0101	–	Always (Unconditional)
SA	1101	SAT = 1	Saturated
LT	0110	(S xor OV) = 1	Less then signed
GE	1110	(S xor OV) = 0	Greater than or equal signed
LE	0111	((S xor OV) or Z) = 1	Less than or equal signed
GT	1111	((S xor OV) or Z) = 0	Greater than signed

C.2 Instruction Set (Alphabetical Order)

(1/5)

Mnemonic	Operands	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
ADD	reg1, reg2	r r r r r 0 0 1 1 1 0 R R R R R	GR[reg2] ← GR[reg2] + GR[reg1]		1	1	1	×	×	×	×	
	imm5, reg2	r r r r r 0 1 0 0 1 0 i i i i i	GR[reg2] ← GR[reg2] + sign-extend (imm5)		1	1	1	×	×	×	×	
ADDI	imm16, reg1, reg2	r r r r r 1 1 0 0 0 0 R R R R R i i i i i i i i i i i i i i i i	GR[reg2] ← GR[reg1] + sign-extend (imm16)		1	1	1	×	×	×	×	
AND	reg1, reg2	r r r r r 0 0 1 0 1 0 R R R R R	GR[reg2] ← GR[reg2] AND GR[reg1]		1	1	1		0	×	×	
ANDI	imm16, reg1, reg2	r r r r r 1 1 0 1 1 0 R R R R R i i i i i i i i i i i i i i i i	GR[reg2] ← GR[reg1] AND zero-extend (imm 16)		1	1	1		0	0	×	
Bcond	disp9	d d d d d 1 0 1 1 d d d c c c c **Note 1**	if conditions are satisfied then PC ← PC + sign extend (disp9)	Conditions satisfied	**3** **Note 2**	**3** **Note 2**	**3** **Note 2**					
				Conditions not satisfied	1	1	1					
BSH	reg2, reg3	r r r r r 1 1 1 1 1 1 0 0 0 0 0 w w w w w 0 1 1 0 1 0 0 0 0 1 0	GR[reg3] ← GR[reg2] (23:16) ll GR[reg2] (31:24)llGR [reg2] (7:0)llGR[reg2] (15:8)		1	1	1	×	0	×	×	
BSW	reg2, reg3	r r r r r 1 1 1 1 1 1 0 0 0 0 0 w w w w w 0 1 1 0 1 0 0 0 0 0 0	GR[reg3] ← GR[reg2] (7:0) ll GR[reg2] (15:8)llGR [reg2] (23:16)llGR[reg2] (31:24)		1	1	1	×	0	×	×	
CALLT	imm6	0 0 0 0 0 0 1 0 0 0 i i i i i i	CTPC ← PC + 2 (return PC) CTPSW ← PSW adr ← CTBP + zero-extend (imm6 logically shift left by 1) PC ← CTBP + zero-extend (Load-memory (adr, Halfword)		5	5	5					
CLR1	bit#3, disp16[reg1]	1 0 b b b 1 1 1 1 1 0 R R R R R d d d d d d d d d d d d d d d d	adr ← GR[reg1] + sign-extend (disp 16) 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]	1 0 b b b 1 1 1 1 1 0 R R R R R d d d d d d d d d d d d d d d d	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	r r r r r 1 1 1 1 1 1 i i i i i w w w w w 0 1 1 0 0 0 c c c c 0	if conditions are satisfied then GR[reg3] ← sign-extend (imm5) else GR[reg3] ← GR[reg2]		1	1	1					
	cccc, reg1, reg2, reg3	r r r r r 1 1 1 1 1 1 R R R R R w w w w w 0 1 1 0 0 1 c c c c 0	if conditions are satisfied then GR[reg3] ← GR[reg1] else GR[reg3] ← GR[reg2]		1	1	1					
CMP	reg1, reg2	r r r r r 0 0 1 1 1 1 R R R R R	result ← GR[reg2] – GR[reg1]		1	1	1	×	×	×	×	
	imm5, reg2	r r r r r 0 1 0 0 1 1 i i i i i	result ← GR[reg2] – sign-extend (imm5)		1	1	1	×	×	×	×	
CTRET		0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0	PC ← CTPC PSW ← CTPSW		4	4	4	R	R	R	R	R
DBRET		0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 1 0	PC ← DBPC PSW ← DBPSW		4	4	4	R	R	R	R	R
DBTRAP		1 1 1 1 1 0 0 0 0 1 0 0 0 0 0 0	DBPC ← PC + 2 (return PC) DBPSW ← PSW PSW.NP ← 1 PSW.EP ← 1 PSW.ID ← 1 PC ← 00000060H		4	4	4					
DI		0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0	PSW.ID ← 1		1	1	1					

(2/5)

Mnemonic	Operands	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
DISPOSE	imm5, list12	0 0 0 0 0 1 1 0 0 1 i i i i i L L L L L L L L L L L L 0 0 0 0 0	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 regs in list 12 is loaded		n+1 Note 4	n+1 Note 4	n+1 Note 4					
	imm5, list12[reg1]	0 0 0 0 0 1 1 0 0 1 i i i i i L L L L L L L L L L L L R R R R R **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 regs in list12 is loaded PC ← GR[reg1]		n+3 Note 4	n+3 Note 4	n+3 Note 4					
DIV	reg1, reg2, reg3	r r r r r 1 1 1 1 1 1 R R R R R w w w w w 0 1 0 1 1 0 0 0 0 0 0	GR[reg2] ← GR[reg2] ÷ GR[reg1] GR[reg3] ← GR[reg2]%GR[reg1]		35	35	35		×	×	×	
DIVH	reg1, reg2	r r r r r 0 0 0 0 1 0 R R R R R	GR[reg2] ← GR[reg2] ÷ GR[reg1]**Note 6**		35	35	35		×	×	×	
	reg1, reg2, reg3	r r r r r 1 1 1 1 1 1 R R R R R w w w w w 0 1 0 1 0 0 0 0 0 0 0	GR[reg2] ← GR[reg2] ÷ GR[reg1]**Note 6** GR[reg3] ← GR[reg2]%GR[reg1]		35	35	35		×	×	×	
DIVHU	reg1, reg2, reg3	r r r r r 1 1 1 1 1 1 R R R R R w w w w w 0 1 0 1 0 0 0 0 0 1 0	GR[reg2] ← GR[reg2] ÷ GR[reg1]**Note 6** GR[reg3] ← GR[reg2]%GR[reg1]		34	34	34		×	×	×	
DIVU	reg1, reg2, reg3	r r r r r 1 1 1 1 1 1 R R R R R w w w w w 0 1 0 1 0 0 0 0 0 1 0	GR[reg2] ← GR[reg2] ÷ GR[reg1] GR[reg3] ← GR[reg2]%GR[reg1]		34	34	34		×	×	×	
EI		1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0	PSW.ID ← 0		1	1	1					
HALT		0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0	Stop		1	1	1					
HSW	reg2, reg3	r r r r r 1 1 1 1 1 1 0 0 0 0 0 w w w w w 0 1 1 0 1 0 0 0 1 0 0	GR[reg3] ← GR[reg2] (15:0)||GR[reg2] (31:16)		1	1	1	×	0	×	×	
JARL	disp22, reg2	r r r r r 1 1 1 1 0 d d d d d d d d d d d d d d d d d d d d d 0	GR[reg2] ← PC + 4 PC ← PC + sign-extend (disp22)		3	3	3					
JMP	[reg1]	0 0 0 0 0 0 0 0 0 0 1 1 R R R R R	PC ← GR[reg1]		4	4	4					
JR	disp22	0 0 0 0 0 1 1 1 1 0 d d d d d d d d d d d d d d d d d d d d d 0 **Note 7**	PC ← PC + sign-extend (disp22)		3	3	3					
LD.B	disp16[reg1], reg2	r r r r r 1 1 1 0 0 0 R R R R R d d d d d d d d d d d d d d d d	adr ← GR[reg1] + sign-extend (disp16) GR[reg2] ← sign-extend (Load-memory (adr, Byte))		1	1	Note 11					
LD.BU	disp16[reg1], reg2	r r r r r 1 1 1 1 0 b R R R R R d d d d d d d d d d d d d d d 1 **Notes 8, 10**	adr ← GR[reg1] + sign-extend (disp16) GR[reg2] ← zero (Load-memory (adr, Byte))		1	1	Note 11					
LD.H	disp16[reg1], reg2	r r r r r 1 1 1 0 0 1 R R R R R d d d d d d d d d d d d d d d 0 **Note 8**	adr ← GR[reg1] + sign-extend (disp16) GR[reg2] ← sign-extend (Load-memory (adr, Halfword))		1	1	Note 11					
LDSR	reg2, regID	r r r r r 1 1 1 1 1 1 R R R R R	SR[regID] ← GR[reg2]	Other than regID = PSW	1	1	1					
		0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 **Note 12**		regID = PSW	1	1	1	×	×	×	×	×
LD.HU	disp16[reg1], reg2	r r r r r 1 1 1 0 0 1 R R R R R d d d d d d d d d d d d d d d 1 **Note 8**	adr ← GR[reg1] + sign-extend (disp16) GR[reg2] ← zero-extend (Load-memory (adr, Halfword))		1	1	Note 11					

(3/5)

Mnemonic	Operands	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
LD.W	disp16[reg1], reg2	r r r r r 1 1 1 0 0 1 R R R R R d d d d d d d d d d d d d d d d **Note 8**	adr ← GR[reg1] + sign-extend (disp16) GR[reg2] ← Load-memory (adr, Word)	1	1	Note 11					
MOV	reg1, reg2	r r r r r 0 0 0 0 0 0 R R R R R	GR[reg2] ← GR[reg1]	1	1	1					
	imm5, reg2	r r r r r 0 1 0 0 0 0 i i i i i	GR[reg2] ← sign-extend (imm5)	1	1	1					
	imm32, reg1	0 0 0 0 0 1 1 0 0 0 1 R R R R R i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i	GR[reg1] ← imm32	2	2	2					
MOVEA	imm16, reg1, reg2	r r r r r 1 1 0 0 0 1 R R R R R i i i i i i i i i i i i i i i i	GR[reg2] ← GR[reg1] + sign-extend (imm16)	1	1	1					
MOVHI	imm16, reg1, reg2	r r r r r 1 1 0 0 1 0 R R R R R i i i i i i i i i i i i i i i i	GR[reg2] ← GR[reg1] + (imm16 ‖ 0^{16})	1	1	1					
MUL Note 22	reg1, reg2, reg3	r r r r r 1 1 1 1 1 1 R R R R R w w w w w 0 1 0 0 0 1 0 0 0 0 0	GR[reg3] ‖ GR[reg2] ← GR[reg2] × GR[reg1] reg1 ≠ reg2 ≠ reg3, reg3 ≠ r0	1	2 Note 14	2					
	imm9, reg2, reg3	r r r r r 1 1 1 1 1 1 i i i i i w w w w w 0 1 0 0 1 I I I I 0 0 **Note 13**	GR[reg3] ‖ GR[reg2] ← GR[reg2] × sign-extend (imm9)	1	2 Note 14	2					
MULH	reg1, reg2	r r r r r 0 0 0 1 1 1 R R R R R	GR[reg2] ← GR[reg2] **Note 6** × GR[reg1] **Note 6**	1	1	2					
	imm5, reg2	r r r r r 0 1 0 1 1 1 i i i i i	GR[reg2] ← GR[reg2] **Note 6** × sign-extend (imm5)	1	1	2					
MULHI	imm16, reg1, reg2	r r r r r 1 1 0 1 1 1 R R R R R i i i i i i i i i i i i i i i i	GR[reg2] ← GR[reg1] **Note 6** × imm16	1	1	2					
MULU Note 22	reg1, reg2, reg3	r r r r r 1 1 1 1 1 1 R R R R R w w w w w 0 1 0 0 0 1 0 0 1 0	GR[reg3] ‖ GR[reg2] ← GR[reg2] × GR [reg1] reg1 ≠ reg2 ≠ reg3, reg3 ≠ r0	1	2 Note 14	2					
	imm9, reg2, reg3	r r r r r 1 1 1 1 1 1 i i i i i w w w w w 0 1 0 0 1 I I I I 1 0 **Note 13**	GR[reg3]‖GR[reg2] ← GR[reg2] × zero-extend (imm9)	1	2 Note 14	2					
NOP		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Passes at least 1 cycle doing nothing.	1	1	1					
NOT	reg1, reg2	r r r r r 0 0 0 0 0 1 R R R R R	GR[reg2] ← NOT (GR[reg1])	1	1	1		0	×	×	
NOT1	bit#3, disp16[reg1]	0 1 b b b 1 1 1 1 1 0 R R R R R d d d d d d d d d d d d d d d d	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]	r r r r r 1 1 1 1 1 1 R R R R R 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0	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				×	
OR	reg1, reg2	r r r r r 0 0 1 0 0 0 R R R R R	GR[reg2] ← GR[reg2] OR GR[reg1]	1	1	1		0	×	×	
ORI	imm16, reg1, reg2	r r r r r 1 1 0 1 0 0 R R R R R i i i i i i i i i i i i i i i i	GR[reg2] ← GR[reg1] OR zero-extend (imm16)	1	1	1		0	×	×	
PREPARE	list12, imm5	0 0 0 0 0 1 1 1 1 0 i i i i i L L L L L L L L L L L L 0 0 0 0 1	Store-memory (sp-4, GR[reg in list12], Word) sp ← sp–4 repeat 1 steps above until 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 Note15	0 0 0 0 0 1 1 1 1 0 i i i i i L L L L L L L L L L L L f f 0 1 1 imm16/imm32 **Note 16**	Store-memory (sp-4, GR[reg in list12], Word) GR[reg in list12] ← Load-memory (sp, Word) sp ← sp + 4 repeat 2 steps above until regs in list12 is loaded PC ← GR[reg1]	n+2 Note 4 Note 17	n+2 Note 4 Note 17	n+2 Note 4 Note 17					

(4/5)

Mnemonic	Operands	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
RETI		0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0	if PSW.EP = 1 then PC ← EIPC PSW ← EIPSW else if PSW.NP = 1 then PC ← FEPC PSW ← FEPSW else PC ← EIPC PSW ← EIPSW	4	4	4	R	R	R	R	R
SAR	reg1, reg2	r r r r r 1 1 1 1 1 1 R R R R R 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0	GR[reg2] ← GR[reg2] arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5, reg2	r r r r r 0 1 0 1 0 1 i i i i i	GR[reg2] ← GR[reg2] arithmetically shift right by zero-extend (imm5)	1	1	1	×	0	×	×	
SASF	cccc, reg2	r r r r r 1 1 1 1 1 1 0 c c c c 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0	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	r r r r r 0 0 0 1 1 0 R R R R R	GR[reg2] ← saturated (GR[reg2] + GR[reg1])	1	1	1	×	×	×	×	×
	imm5, reg2	r r r r r 0 1 0 0 0 1 i i i i i	GR[reg2] ← saturated (GR[reg2] sign-extend (imm5))	1	1	1	×	×	×	×	×
SATSUB	reg1, reg2	r r r r r 0 0 0 1 0 1 R R R R R	GR[reg2] ← saturated (GR[reg2] – GR[reg1])	1	1	1	×	×	×	×	×
SATSUBI	imm16, reg1, reg2	r r r r r 1 1 0 0 1 1 R R R R R i i i i i i i i i i i i i i i i	GR[reg2] ← saturated (GR[reg1] – sign-extend (imm16)	1	1	1	×	×	×	×	×
SATSUBR	reg1, reg2	r r r r r 0 0 0 1 0 0 R R R R R	GR[reg2] ← saturated (GR[reg1] – GR[reg2])	1	1	1	×	×	×	×	×
SETF	cccc, reg2	r r r r r 1 1 1 1 1 1 0 c c c c 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	if conditions are satisfied then GR[reg2] ← 00000001H else GR[reg2] ← 00000000H	1	1	1					
SET1	bit#3, disp16 [reg1]	0 0 b b b 1 1 1 1 1 0 R R R R R d d d d d d d d d d d d d d d d	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]	r r r r r 1 1 1 1 1 1 R R R R R 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0	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	r r r r r 1 1 1 1 1 1 R R R R R 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0	GR[reg2] ← GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×	
	imm5, reg2	r r r r r 0 1 0 1 1 0 i i i i i	GR[reg2] ← GR[reg2] logically shift left by zero-extend (imm5)	1	1	1	×	0	×	×	
SHR	reg1, reg2	r r r r r 1 1 1 1 1 1 R R R R R 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0	GR[reg2] ← GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5, reg2	r r r r r 0 1 0 1 0 0 i i i i i	GR[reg2] ← GR[reg2] logically shift right by zero-extend (imm5)	1	1	1	×	0	×	×	
SLD.B	disp7[ep], reg2	r r r r r 0 1 1 0 d d d d d d d	adr ← ep + zero-extend (disp7) GR[reg2] ← sign-extend (Load-memory (adr, Byte))	1	1	Note 9					
SLD.BU	disp4[ep], reg2	r r r r r 0 0 0 0 1 1 0 d d d d **Note 18**	adr ← ep + zero-extend (disp4) GR[reg2] ← zero-extend (Load-memory (adr, Byte))	1	1	Note 9					
SLD.H	disp8[ep], reg2	r r r r r 1 0 0 0 d d d d d d d **Note 19**	adr ← ep + zero-extend (disp8) GR[reg2] ← sign-extend (Load-memory (adr, Halfword))	1	1	Note 9					

(5/5)

Mnemonic	Operands	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SLD.HU	disp5[ep], reg2	rrrrr000001111dddd **Notes 18, 20**	adr ← ep + zero-extend (disp5) GR[reg2] ← zero-extend (Load-memory (adr, Halfword)	1	1	**Note 9**					
SLD.W	disp8[ep], reg2	rrrrr1010ddddddd0 **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], Halfword)	1	1	1					
SST.W	reg2, disp8[ep]	rrrrr1010ddddddd1 **Note 21**	adr ← ep + zero-extend (disp8) Store-memory (adr, GR[reg2], Word)	1	1	1					
ST.B	reg2, disp16 [reg1]	rrrrr111010RRRRR ddddddddddddddddd	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], Halfword)	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 0000000000100000	GR[reg2] ← SR[regID]	1	1	1					
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	0000000000010RRRRR	adr ← (PC + 2) + GR[reg1] logically shift left by 1) PC ← (PC + 2) + (sign-extend (Load-memory (adr, Halfword)) logically shift left by 1	5	5	5					
SXB	reg1	0000000001010RRRRR	GR[reg1] ← sign-extend (GR[reg1] (7:0))	1	1	1					
SXH	reg1	0000000001110RRRRR	GR[reg1] ← sign-extend (GR[reg1] (15:0))	1	1	1					
TRAP	vector	0000011111iiiii 0000000010000000	EIPC ← PC + 4 (return PC) EIPSW ← PSW ECR.EICC ← exception code (40H to 4FH, 50H to 5FH) PSW.EP ← 1 PSW.ID ← 1 PC ← 00000040H (when vector is 00H to 0FH (exception code: 40H to 4FH)) 00000050H (when vector is 10H to 1FH (exception code: 50H to 5FH))	4	4	4					
TST	reg1, reg2	rrrrr001011RRRRR	result ← GR[reg2] AND GR[reg1]	1	1	1		0	×	×	
TST1	bit#3, disp16 [reg1]	11bbb111110RRRRR ddddddddddddddd	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 0000000001110110	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	0000000001100RRRRR	GR[reg1] ← zero-extend (GR[reg1] (7:0))	1	1	1					
ZXH	reg1	0000000001110RRRRR	GR[reg1] ← zero-extend (GR[reg1] (15:0))	1	1	1					

Notes 1. dddddddd is the higher 8 bits of disp9.

2. 4 if there is an instruction to overwrite the contents of the PSW immediately before.

3. If there is no wait state (3 + number of read access wait states)

4. n is the total number of load registers in list12 (According to the number of wait states. If there are no wait states, n is the total number of registers in list12. When n = 0, the operation is the same as n = 1.)

5. RRRRR: Other than 00000

6. Only the lower halfword of data is valid.

7. ddddddddddddddddddddd is the higher 21 bits of disp22.

8. ddddddddddddddd is 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)

12. In this instruction, although the source register is regarded as reg2 for convenience of the mnemonic description, the reg1 field is used in the opcode. Therefore, the meanings of register specifications assigned in the mnemonic description and in the opcode differ from those in other instructions.
rrrrr = regID specification
RRRRR = reg2 specification

13. iiiii: Lower 5 bits of imm9
IIII: Higher 4 bits of imm9

14. Shortened by 1 clock if reg2 = reg3 (lower 32 bits of result are not written to register) or reg3 = r0 (higher 32 bits of result are not written to register).

15. sp/imm: Specify in bits 19 and 20 of sub-opcode.

16. ff = 00: Load sp in ep.
01: Load sign-extended 16-bit immediate data (bits 47 to 32) in ep.
10: Load 16-bit immediate data (bits 47 to 32) logically shifted 16 bits to the right in ep.
11: Load 32-bit immediate data (bits 63 to 32) in ep.

17. n + 3 clocks when imm = imm32

18. rrrrr: Other then 00000

19. ddddddd is the higher 7 bits of disp8.

20. dddd is the higher 4 bits of disp5.

21. dddddd is the higher 6 bits of disp8.

22. In the MUL reg1, reg2, reg3 and MULU reg1, reg2, reg3 instructions, prevent a combination of registers that satisfies all of the following conditions. The operation when the instructions are executed with the following conditions satisfied is not guaranteed.

- reg1 = reg3
- reg1 ≠ reg2
- reg1 ≠ r0
- reg3 ≠ r0

APPENDIX D REVISION HISTORY

D.1 Major Revisions in This Edition

(1/2)

Page	Description
Throughout	• Addition of the following lead-free products μPD703114GC-xxx-8EU-A, 703114GC(A)-xxx-8EU-A, 703114GF-xxx-3BA-A, 70F3114GC-8EU-A, 70F3114GC(A)-8EU-A, 70F3114GF-3BA-A • Addition of FLPMC register
p. 18	Addition of **Note** to **Table 1-1 Differences Between V850E/IA1 and V850E/IA2**
p. 19	Change of number of instructions in **1.2 Features**
p. 49	Addition of **Note** to **Table 3-2 System Register Numbers**
pp. 50, 51, 53, 54	Addition of **3.2.2 (1) Interrupt status saving registers (EIPC, EIPSW)**, **(2) NMI status saving registers (FEPC, FEPSW)**, **(5) CALLT execution status saving registers (CTPC, CTPSW)**, **(6) Exception/debug trap status saving registers (DBPC, DBPSW)**, and **(7) CALLT base pointer (CTBP)**
p. 79	Addition of **3.4.11 (2) Restriction on conflict between sld instruction and interrupt request**
pp. 114, 115	Modification of description in **6.3.5 DMA channel control registers 0 to 3 (DCHC0 to DCHC3)**
p. 116	Modification of description in **6.3.7 DMA restart register (DRST)**
pp. 117, 119	Modification of description and addition of **Caution** to **6.3.8 DMA trigger factor registers 0 to 3 (DTFR0 to DTFR3)**
p. 123	Addition of **Figure 6-7 Block Transfer Example**
p. 123	Modification of description of **Caution** in **6.5.1 Two-cycle transfer**
p. 124	Addition of **Note** to **Table 6-1 Relationship Between Transfer Type and Transfer Target**
p. 125	Deletion of a part of description in **6.7 DMA Channel Priorities**
pp. 125, 126	Modification of description in **6.8 Next Address Setting Function**
p. 129	Addition of **Figure 6-9 Example of Forcible Termination of DMA Transfer**
p. 132	Modification of descriptions in **6.14 (2) Transfer of misaligned data** and **(4) DMA start factors**
p. 132	Addition of **6.14 (5) Program execution and DMA transfer with internal RAM**
p. 134	Addition of **Caution** to **7.1 Features**
pp. 135, 137	Addition of **Note** and **Remark** to **Table 7-1 Interrupt/Exception Source List**
p. 160	Addition of **Caution** to **7.3.8 (4) Timer 2 input filter mode registers 0 to 5 (FEM0 to FEM5)**
p. 169	Addition of **Caution** to **7.5.2 (2) Restore**
p. 173	Modification of description in **7.8 Periods in Which CPU Does Not Acknowledge Interrupts**
p. 185	Modification of description in **8.5.2 (3) Power save control register (PSC)**
p. 189	Addition of description to **Table 8-4 Operation Status in IDLE Mode**
p. 190	Addition of **Caution** to **8.5.4 (2) (a) Release by a non-maskable interrupt request or an unmasked maskable interrupt request**
p. 191	Addition of description to **Table 8-6 Operation Status in Software STOP Mode**
p. 192	Addition of **Caution** to **8.5.5 (2) (a) Release by a non-maskable interrupt request or an unmasked maskable interrupt request**
p. 279	Addition of **9.1.6 (4) [Output waveform width with respect to set value] (d) When BFCMnx = 0000H is set while DTMnx = 000H or TM0CEDn bit = 1**
p. 281	Addition of **9.1.6 (4) [Output waveform width with respect to set value] (e) When BFCMnx = CM0n3 = a is set**
p. 297	Addition of **Caution** to **9.2.3 (1) Timer 10 (TM10)**
p. 305	Modification of description in table in **9.2.4 (6) (b) UDC mode (CMD bit of TUM0 register = 1)**
p. 313	Modification of description in **Table 9-8 List of Count Operations in UDC Mode**

(2/2)

Page	Description
p. 337	Addition of **9.3.4 (6) (a) Caution for PWM output change timing**
p. 410	Addition of **Remark** to **Figure 10-2 Asynchronous Serial Interface 0 Block Diagram**
p. 414	Deletion of a part of description and addition of **Caution** to **10.2.3 (2) Asynchronous serial interface status register 0 (ASIS0)**
p. 438	Addition of description to **10.2.6 (5) Transfer rate during continuous transmission**
p. 438	Addition of description to **10.2.7 Cautions (2)**
p. 459	Modification of **Figure 10-20 Asynchronous Serial Interface Reception Completion Interrupt Timing**
p. 583	Addition of **Caution** to **12.5.2 (1) Timer 10 noise elimination time selection register (NRC10)**
p. 584	Addition of **Caution** to **12.5.2 (2) Timer 3 noise elimination time selection register (NRC3)**
p. 586	Addition of **Caution** to **12.5.3 (1) Timer 2 input filter mode registers 0 to 5 (FEM0 to FEM5)**
p. 611	Addition of **15.6 Programming Method**
p. 614	Addition of **15.7 Flash Memory Programming by Self-Programming**
p. 635	Addition of **15.8 How to Distinguish Flash Memory and Mask ROM Versions**
p. 663	Addition of **(3)** and **(4)** to **Table 18-1 Surface Mounting Type Soldering Conditions**
p. 626 in previous edition	Deletion of **APPENDIX A NOTES**

D.2 Revision History up to Previous Edition

The following table shows the revision history up to the previous edition. The “Applied to:” column indicates the chapters of each edition in which the revision was applied.

(1/7)

Edition	Major Revision up to Previous Edition	Applied to:
2nd	Change of description on memory space in **1.2 Features**	**CHAPTER 1 INTRODUCTION**
	Change of description on regulator in **1.2 Features**	
	Deletion of **Note** in **1.4 Ordering Information**	
	Change of ASTB (PCT6) pin status in **2.2 Pin Status**	**CHAPTER 2 PIN FUNCTIONS**
	Change of I/O circuit type from 5-K to 5-AC in **2.4 Types of Pin I/O Circuits and Connection of Unused Pins**	
	Change of I/O circuit type from 5-K to 5-AC in **2.5 Pin I/O Circuits**	
	Modification of **Figure 3-3 Memory Map**	**CHAPTER 3 CPU FUNCTION**
	Addition and deletion of description in **3.4.5 (2) Internal RAM area**	
	Modification of description in **3.4.5 (4) External memory area**	
	Deletion of description in **3.4.7 (1) Program space**	
	Deletion of part of description in example of wrap-around application in **3.4.7 (2) Data space**	
	Modification of **Figure 3-5 Recommended Memory Map**	
	Addition and modification of description in **3.4.8 Peripheral I/O registers**	
	Addition and modification of description in **3.4.10 System wait control register (VSWC)**	
	Addition and modification of description in **4.2.1 Pin status during internal ROM, internal RAM, and peripheral I/O access**	**CHAPTER 4 BUS CONTROL FUNCTION**
	Addition and modification of description in **4.3 Memory Block Function**	
	Addition of **4.3.1 Chip select control function**	
	Addition of description in **4.4.1 (1) Bus cycle type configuration registers 0, 1 (BCT0, BCT1)**	
	Addition of indication of **Note** in **4.5.1 Number of access clocks**	
	Addition of **4.5.2 Bus sizing function**	
	Addition of description in **4.6.1 (1) Data wait control registers 0, 1 (DWC0, DWC1)**	
	Addition of description in **4.6.1 (2) Address wait control register (AWC)**	
	Change of timing in **Figure 4-2 Example of Wait Insertion**	
	Addition of description in **4.7 (1) Bus cycle control register (BCC)**	
	Addition of description in **6.3.3 DMA byte count registers 0 to 3 (DBC0 to DBC3)**	**CHAPTER 6 DMA FUNCTIONS (DMA CONTROLLER)**
	Change of description when DS1, DS0 bits = 1, 0 in **6.3.4 DMA addressing control registers 0 to 3 (DADC0 to DADC3)**	
	Addition of **Cautions** in **6.3.5 DMA channel control registers 0 to 3 (DCHC0 to DCHC3)**	
	Change of description on bit that can be manipulated in **6.3.6 DMA disable status register (DDIS)**	
	Change of description on bit that can be manipulated in **6.3.7 DMA restart register (DRST)**	
	Addition of description in **6.5.1 Single transfer mode**	
	Addition of description in **6.5.2 Single-step transfer mode**	
	Change of transfer status when transfer target is in internal RAM in **Table 6-1 Relationship Between Transfer Type and Transfer Target**	
	Addition of **Caution** in **6.8 DMA Channel Priorities**	
	Addition of **6.14 (5) DMA start factors**	

(2/7)

Edition	Major Revision up to Previous Edition	Applied to:
2nd	Addition of generating source of CC10IC1 register in **Table 7-1 Interrupt/Exception Source List**	**CHAPTER 7 INTERRUPT/ EXCEPTION PROCESSING FUNCTION**
	Change of description in **Figure 7-2 Acknowledging Non-Maskable Interrupt Request**	
	Addition of **Caution** and change of description in **7.3.8 (2) Signal edge selection register 10 (SESA10)**	
	Addition of **Caution** in **7.3.8 (3) Valid edge selection register (SESC)**	
	Addition and change of description in **7.3.8 (4) Timer 2 input filter mode registers 0 to 5 (FEM0 to FEM5)**	
	Modification of description in **7.8 Periods in Which Interrupts Are Not Acknowledged**	
	Change of description on bits that can be manipulated and data setting sequences to CKC in **8.3.4 Clock control register (CKC)**	**CHAPTER 8 CLOCK GENERATION FUNCTION**
	Modification of **Note** in **Figure 8-1 Power Save Mode State Transition Diagram**	
	Modification of operation status of ASTB in **Table 8-4 Operation Status in IDLE Mode**	
	Addition and modification of description in **8.5.4 (2) Release of IDLE mode**	
	Change of operation status of ASTB in **Table 8-6 Operation Status in Software STOP Mode**	
	Addition and modification of description in **8.5.5 (2) Release of software STOP mode**	
	Addition and modification of description and change of timing chart in **8.6.1 (1) Securing the time using an on-chip time base counter**	
	Modification of timing chart in **8.6.1 (2) Securing the time according to the signal level width (RESET pin input)**	
	Addition of a table in **9.1.2 Function overview (timer 0)**	**CHAPTER 9 TIMER/COUNTER FUNCTION (REAL-TIME PULSE UNIT)**
	Addition of **Caution** in **Table 9-2 Operation Modes of Timer 0**	
	Addition and modification of description in **9.1.5 (3) Timer unit control registers 00, 01 (TUC00, TUC01)**	
	Modification of description in **9.1.5 (4) Timer output mode registers 0, 1 (TOMR0, TOMR1)**	
	Addition and modification of description in **9.1.5 (6) PWM software timing output registers 0, 1 (PSTO0, PSTO1) and addition of Figures 9-9 to 9-14**	
	Addition of **Remark** in **9.1.6 Operation**	
	Addition of **Remark** in **9.1.6 (2) PWM mode 0: Triangular wave modulation (right-left symmetric waveform control) [Output waveform width in respect to set value]**	
	Addition of **Remark** in **9.1.6 (3) PWM mode 1: Triangular wave modulation (right-left asymmetric waveform control) [Output waveform width in respect to set value]**	
	Addition of **Remark** in **9.1.6 (4) PWM mode 2: Sawtooth wave modulation [Output waveform width in respect to set value]**	
	Addition of **Remark** in **Figure 9-30 TM0CEn Bit Write and TM0n Timer Operation Timing**	
	Change of description in **9.2.2 Function overview (timer 1)**	
	Change of description in **Table 9-5 Timer 1 Configuration List**	
	Modification of **Figure 9-45 Block Diagram of Timer 1**	
	Modification of description in **9.2.4 (1) Timer 1/timer 2 clock selection register (PRM02)**	
	Addition of description in **9.2.4 (3) Timer control register 10 (TMC10)**	
	Modification of description in **9.2.4 (5) Signal edge selection register 10 (SESA10)**	
	Change of description in **Figure 9-46 TM10 Block Diagram (During PWM Output Operation)**	
	Change of description in **9.3.2 Function overview (timer 2)**	
	Change of description in **Table 9-9 Timer 2 Configuration List**	
	Addition of **Table 9-10 Capture/Compare Operation Sources**	
	Addition of **Table 9-11 Output Level Sources During Timer Output**	
	Change of description in **Figure 9-62 Block Diagram of Timer 2**	

(3/7)

Edition	Major Revision up to Previous Edition	Applied to:
2nd	Modification of description in **9.3.4 (1) Timer 1/timer 2 clock selection register (PRM02)**	**CHAPTER 9 TIMER/COUNTER FUNCTION (REAL-TIME PULSE UNIT)**
	Modification of description in **9.3.4 (2) Timer 2 clock stop register 0 (STOPTE0)**	
	Addition of **Caution** and modification in **9.3.4 (5) Timer 2 time base control register 0 (TCRE0)**	
	Addition of **Note** and deletion of **Caution** in **Figure 9-95 Cycle Measurement Operation Timing Example**	
	Modification of description in **Figure 9-97 Example of Timing During TM4 Operation**	
	Modification of **Caution** in **10.2.3 (1) Asynchronous serial interface mode register 0 (ASIM0)**	**CHAPTER 10 SERIAL INTERFACE FUNCTION**
	Change of description on bits that can be manipulated in **10.2.3 (2) Asynchronous serial interface status register 0 (ASIS0)**	
	Addition of Caution and modification of description in **10.2.3 (3) Asynchronous serial interface transmission status register 0 (ASIF0)**	
	Change of description on bits that can be manipulated in **10.2.3 (4) Receive buffer register (RXB0)**	
	Change of description on bits that can be manipulated in **10.2.3 (5) Transmit buffer register 0 (TXB0)**	
	Addition and modification of description in **10.2.5 (3) Continuous transmission operation**	
	Addition of **Figure 10-5 Continuous Transmission Processing Flow**	
	Addition of **Note** and change of description in table in **Figure 10-6 Continuous Transmission Starting Procedure**	
	Change of description of table in **Figure 10-7 Continuous Transmission End Procedure**	
	Addition of **Cautions** in **Figure 10-8 Asynchronous Serial Interface Reception Completion Interrupt Timing**	
	Change of description on bits that can be manipulated and addition of **Caution** in **10.2.6 (2) (a) Clock select register 0 (CKSR0)**	
	Change of description on bits that can be manipulated in **10.2.6 (2) (b) Baud rate generator control register 0 (BRGC0)**	
	Addition of (2) in **10.2.7 Cautions**	
	Change of description on bits that can be manipulated in **10.3.3 (4) 2-frame continuous reception buffer register 1 (RXB1)/receive buffer register L1 (RXBL1)**	
	Addition of Caution in **10.3.4 (1) Reception completion interrupt (INTSR1)**	
	Addition of **10.3.5 (3) Continuous transmission of 3 or more frames**	
	Change of description on bits that can be manipulated in **10.3.7 (2) (c) Prescaler compare register 1 (PRSCM1)**	
	Addition of **10.3.7 (3) Allowable baud rate range during reception**	
	Addition of **10.3.7 (4) Transfer rate in 2-frame continuous reception**	
	Change of description on bits that can be manipulated in **10.4.3 (4) Clocked serial interface receive buffer registers L0, L1 (SIRBL0, SIRBL1)**	
	Change of description on bits that can be manipulated in **10.4.3 (6) Clocked serial interface read-only receive buffer registers L0, L1 (SIRBEL0, SIRBEL1)**	
	Change of description on bits that can be manipulated in **10.4.3 (8) Clocked serial interface transmit buffer registers L0, L1 (SOTBL0, SOTBL1)**	
	Change of description on bits that can be manipulated in **10.4.3 (10) Clocked serial interface initial transmit buffer registers L0, L1 (SOTBFL0, SOTBFL1)**	
	Change of description on bits that can be manipulated in **10.4.3 (12) Serial I/O shift registers L0, L1 (SIOL0, SIOL1)**	
	Modification of caution description in **10.4.6 (2) (b) Prescaler mode register 3 (PRSM3)**	
	Change of description on bits that can be manipulated and **Caution** in **10.4.6 (2) (c) Prescaler compare register 3 (PRSCM3)**	

(4/7)

Edition	Major Revision up to Previous Edition	Applied to:
2nd	Addition of **Caution** in **11.4 (1) A/D scan mode registers 00 and 10 (ADSCM00, ASDSCM10)**	**CHAPTER 11 A/D CONVERTER**
	Change of description on bits that can be manipulated and change of explanation of FR2 to FR0 bits in **11.4 (2) A/D scan mode registers 01 and 11 (ADSCM01, ADSCM11)**	
	Addition of **11.11.6 Timing that makes the A/D conversion result undefined**	
	Addition of **11.12 How to Read A/D Converter Characteristics Table**	
	Modification of description in **12.2 (1) Functions of each port**	**CHAPTER 12 PORT FUNCTIONS**
	Modification of Figure **12-4 Type D Block Diagram**	
	Modification of Figure **12-7 Type G Block Diagram**	
	Modification of Figure **12-8 Type H Block Diagram**	
	Modification of Figure **12-13 Type M Block Diagram**	
	Addition of Figure **12-14 Type N Block Diagram**	
	Change of description in **12.3.6 (1) Operation in control mode**	
	Modification of Figure **12-15 Example of Noise Elimination Timing**	
	Addition of **Caution** and change of description in **12.4.3 (1) Timer 2 input filter mode registers 0 to 5 (FEM0 to FEM5)**	
	Addition of **13.2 (2) <1> Reset circuit** and **<2> Reset timing**	**CHAPTER 13 RESET FUNCTION**
	Addition of item and change of description in **Table 13-2 Initial Values of CPU, Internal RAM, and On-Chip Peripheral I/O After Reset**	
	Modification of description in **14.1 Features**	**CHAPTER 14 REGULATOR**
	Addition and modification of description in **14.2 Functional Outline**	
	Modification of **Figure 14-1 Example of Connection When Using N-ch Transistor**	
	Addition of **Figure 14-2 Mount Pad Dimensions When Mounted on 2SD1950 (VL Standard Product) (Glass Epoxy Board) (Unit: mm)**	
	Addition of **Figure 14-3 Connection When Using External Regulator**	
	Addition and modification of description in **Caution** in **14.4 (1) Regulator control register (REGC)**	
	Addition of **Caution** in **15.2 Writing Using Flash Programmer**	**CHAPTER 15 FLASH MEMORY (μPD70F3114)**
	Addition of description in **15.2 (2) Off-board programming**	
	Modification of description in 15.3 Programming Environment	
	Change of description in **15.4 (1) UART0**	
	Change of description in **15.4 (2) CSI0**	
	Change of description in **15.4 (3) Handshake-supported CSI communication**	
	Modification of description in **15.5.8 Power supply**	
	Change of description in B.2 Instruction Set (Alphabetical Order)	**APPENDIX B INSTRUCTION SET LIST**
3rd	Addition of 100-pin plastic QFP (14 × 20) package	Throughout
	Addition of **Table 1-2 Differences Between V850E/IA1 and V850E/IA2 Register Setting Values**	**CHAPTER 1 INTRODUCTION**
	Modification of description in **4.2.1 Pin status during internal ROM, internal RAM, and on-chip peripheral I/O access**	**CHAPTER 4 BUS CONTROL FUNCTION**
	Addition of **Caution** to **4.3.1 (1) Chip area select control registers 0, 1 (CSC0, CSC1)**	
	Modification and deletion of description in **4.9.1 Program space**	
	Addition of description to **6.3.1 (1) DMA source address registers 0H to 3H (DSA0H to DSA3H)**	**CHAPTER 6 DMA FUNCTIONS (DMA CONTROLLER)**
	Addition of description to **6.3.2 (1) DMA destination address registers 0H to 3H (DDA0H to DDA3H)**	

(5/7)

Edition	Major Revision up to Previous Edition	Applied to:
3rd	Addition of description and **Caution** to **6.3.4 DMA addressing control registers 0 to 3 (DADC0 to DADC3)**	**CHAPTER 6 DMA FUNCTIONS (DMA CONTROLLER)**
	Addition of description and **Caution** to and modification of bit description in **6.3.5 DMA channel control registers 0 to 3 (DCHC0 to DCHC3)**	
	Addition of description to **6.3.6 DMA disable status register (DDIS)**	
	Addition of description to **6.3.7 DMA restart register (DRST)**	
	Addition of **Caution** to **6.6.1 Two-cycle transfer**	
	Addition of description to **Remark** in **6.13 Forcible Termination**	
	Modification of description in **6.14 (3) Times related to DMA transfer**	
	Addition of **Caution** to **7.3.4 Interrupt control register (xxICn)**	**CHAPTER 7 INTERRUPT/ EXCEPTION PROCESSING FUNCTION**
	Addition of **Caution** to **7.3.6 In-service priority register (ISPR)**	
	Modification of description in **Figure 7-14 Pipeline Operation at Interrupt Request Acknowledgment (Outline)**	
	Modification of description in **Table 9-2 Operation Modes of Timer 0**	**CHAPTER 9 TIMER/COUNTER FUNCTION (REAL-TIME PULSE UNIT)**
	Modification of description in **Table 9-4 Operation Modes of Timer 0 (TM0n)**	
	Modification of description in **Remark** in **9.1.6 (2) PWM mode 0: Triangular wave modulation (right-left symmetric waveform control)**	
	Modification of **Figures 9-15**, **9-17** to **9-20**, **9-22** to **9-30**, and **9-32** to **9-35**	
	Modification of maximum transfer rate in **10.2.1 Features**	**CHAPTER 10 SERIAL INTERFACE FUNCTION**
	Addition of description to **Table 10-3 Baud Rate Generator Setting Data**	
	Addition of **Caution** to **12.2 (1) Functions of each port**	**CHAPTER 12 PORT FUNCTIONS**
	Addition of description to **15.2 (2) Off-board programming**	**CHAPTER 15 FLASH MEMORY (μPD70F3114)**
	Addition of **CHAPTER 16 ELECTRICAL SPECIFICATIONS**	**CHAPTER 16 ELECTRICAL SPECIFICATIONS**
	Addition of **CHAPTER 17 PACKAGE DRAWINGS**	**CHAPTER 17 PACKAGE DRAWINGS**
	Addition of **CHAPTER 18 RECOMMENDED SOLDERING CONDITIONS**	**CHAPTER 18 RECOMMENDED SOLDERING CONDITIONS**
	Addition of **APPENDIX A NOTES ON TARGET SYSTEM DESIGN**	**APPENDIX A NOTES ON TARGET SYSTEM DESIGN**
	Modification of description in **C.2 Instruction Set (Alphabetical Order)**	**APPENDIX C INSTRUCTION SET LIST**
	Addition of **APPENDIX D INDEX**	**APPENDIX D INDEX**
	Addition of **APPENDIX E REVISION HISTORY**	**APPENDIX E REVISION HISTORY**

(6/7)

Edition	Major Revision up to Previous Edition	Applied to:
4th	• Addition of the following products μPD703114GC(A)-×××-8EU, 70F3114GC(A)-8EU	Throughout
	Addition of **Note 2** to **1.5 Pin Configuration (Top View)**	**CHAPTER 1 INTRODUCTION**
	Addition of description to **6.3.1 DMA source address registers 0 to 3 (DSA0 to DSA3)**	**CHAPTER 6 DMA FUNCTIONS (DMA CONTROLLER)**
	Addition of **Caution 2** to **6.3.1 DMA source address registers 0H to 3H (DSA0H to DSA3H)**	
	Addition of description to **6.3.2 DMA destination address registers 0 to 3 (DDA0 to DDA3)**	
	Addition of **Caution 2** to **6.3.2 (1) DMA destination address registers 0H to 3H (DDA0H to DDA3H)**	
	Addition of description and **Cautions 1** and **2** to **6.3.3 DMA transfer count registers 0 to 3 (DBC0 to DBC3)**	
	Addition of **Caution 2** to **6.3.4 DMA addressing control registers 0 to 3 (DADC0 to DADC3)**	
	Modification/addition of description of **Caution** in **6.3.5 DMA channel control registers 0 to 3 (DCHC0 to DCHC3)**	
	Modification of description in **6.3.7 DMA restart register (DRST)**	
	Addition of description to **6.3.8 DMA trigger factor registers 0 to 3 (DTFR0 to DTFR3)**	
	Addition of description to **Remark** in **6.7.1 Transfer type and transfer target**	
	Deletion of **Note** from **Table 6-2 External Bus Cycles During DMA Transfer (Two-Cycle Transfer)**	
	Modification of description in **6.9 Next Address Setting Function**	
	Addition of **Cautions 1** and **2** to **6.10 DMA Transfer Start Factors**	
	Modification of description in **6.11 Forcible Suspension**	
	Addition of **6.13.1 Restrictions on forcible termination of DMA transfer**	
	Modification of description in **6.14 Time Required for DMA Transfer**	
	Addition of **6.15 (5) Restrictions related to automatic clearing of TCn bit of DCHCn register** and **(6) Read values of DSAn and DDAn registers**	
	Modification of description in **CHAPTER 7 INTERRUPT/EXCEPTION PROCESSING FUNCTION**	**CHAPTER 7 INTERRUPT/ EXCEPTION PROCESSING FUNCTION**
	Addition of **Caution 2** to **9.1.6 (2) PWM mode 0: Triangular wave modulation (right-left symmetric waveform control)**	**CHAPTER 9 TIMER/COUNTER FUNCTION (REAL-TIME PULSE UNIT)**
	Addition of **Note** to **9.3.4 (3) Timer 2 count clock/control edge selection register 0 (CSE0)**	
	Addition of **9.3.6 PWM output operation in timer 2 compare mode**	
	Modification of description in **Figure 9-91 TM3 Compare Operation Example (Set/Reset Output Mode)**	
	Addition of **Caution 2** to **10.2.3 (1) Asynchronous serial interface mode register 0 (ASIM0)**	**CHAPTER 10 SERIAL INTERFACE FUNCTION**
	Addition of **Caution** to **10.2.5 (3) Continuous transmission operation**	
	Addition of description of transfer rate to **10.3.1 Features**	
	Addition of **Cautions 1** and **2** to **10.3.3 (1) Asynchronous serial interface mode register 10 (ASIM10)**	
	Addition of **Caution 3** to **10.3.7 (2) (c) Prescaler compare register 1 (PRSCM1)**	
	Modification of description in **Table 10-8 Baud Rate Generator Setting Data (BRG = f_{XX}/2)**	

(7/7)

Edition	Major Revision up to Previous Edition	Applied to:
4th	Addition of **Caution** to **12.3.2 (1) Operation in control mode**	**CHAPTER 12 PORT FUNCTIONS**
	Addition of **Caution** to **12.3.3 (1) Operation in control mode**	
	Addition of **Caution** to **12.3.4 (1) Operation in control mode**	
	Modification of description of bits 7 to 5 in **12.3.4 (2) (a) Port 3 mode register (PM3)**	
	Addition of **Caution** to **12.3.5 (1) Operation in control mode**	
	Addition of **Note** to **12.3.9 (1) Operation in control mode**	
	Addition of **12.4 Operation of Port Function**	
	Addition of **12.6 Cautions**	
	Addition of description to **Caution 2** in **13.2 (2) <3> Description**	**CHAPTER 13 RESET FUNCTION**
	Addition of **Caution** to **Data Retention Characteristics** in **16.1 Normal Operation Mode**	**CHAPTER 16 ELECTRICAL SPECIFICATIONS**
	Addition of **(b)** to **AC test input test points** in **16.1 Normal Operation Mode**	
	Change of description of Stabilization capacitance in the Conditions column in **16.1 (3) Regulator output stabilization time**	
	Modification of description of t_{HSTWT1} in **16.1 (5) (a) CLKOUT asynchronous**	
	Addition of **Caution** to **16.1 (5) (c) Read cycle (CLKOUT synchronous/asynchronous, 1 wait)**	
	Addition of **Caution** to **16.1 (5) (d) Write cycle (CLKOUT synchronous/asynchronous, 1 wait)**	
	Addition of **Remark** to **16.1 (8) Timer operating frequency**	
	Addition of description of TXD1 output delay time to **16.1 (11) (a) Clocked master mode**	
	Modification of descriptions in V_{PP} supply voltage (V_{PPL}) row of **Basic Characteristics** in **16.2 Flash Memory Programming Mode**	
	Addition of **APPENDIX A NOTES**	**APPENDIX A NOTES**
	Addition of **Note 22** to MUL and MULU in **D.2 Instruction Set (Alphabetical Order)** in **APPENDIX D**	**APPENDIX C INSTRUCTION SET LIST**
	Modification of description in **APPENDIX E REVISION HISTORY**	**APPENDIX E REVISION HISTORY**